ONTOLOGIES AS BAYESIAN NETWORKS FOR SPACE DEBRIS

by

Stephania Vasilieva

A Thesis Submitted to the Faculty of the

DEPARTMENT OF SYSTEMS AND INDUSTRIAL ENGINEERING

In Partial Fulfillment of the Requirements

For the Degree of

MASTER OF SCIENCE
WITH A MAJOR IN SYSTEMS ENGINEERING

In the Graduate College

THE UNIVERSITY OF ARIZONA

2016

STATEMENT BY AUTHOR

The thesis titled *Ontologies as bayesian networks for space debris* prepared by *Stephania Vasilieva* has been submitted in partial fulfillment of requirements for a master's degree at the University of Arizona and is deposited in the University Library to be made available to borrowers under rules of the Library.

SIGNED: *Stephania Vasilieva*

APPROVAL BY THESIS DIRECTOR

This thesis has been approved on the date shown below:

Roberto Furfaro
Professor of Systems Engineering 05/04/2016

Acknowledgements

I would like to thank my advisor Dr. Roberto Furfaro for giving me the opportunity to work on this thesis and guiding me through it. He has inspired me, always answered my questions, and taught me countless things in order to complete my research. His support and patience have made me grateful to have him as my advisor.

I would like to thank Dr. Richard Linares at the University of Minnesota for his expertise and complementary work, and Dr. Ramona Walls for taking the time to look at my research and providing feedback. She also set up the Space Object Behavior Workshop, where I had the opportunity to meet and learn from industry and academia experts concerning the field of my thesis. I also thank these workshop members.

I would also like to acknowledge Dr. Larry Head and Professor Robert Lepore as the second readers of this thesis.

Finally, I must thank my family and friends for continuously providing me with support and advice on school, research, and writing this thesis. This accomplishment would not have been possible without them.

Thank you,

Stephania Vasilieva

UNIVERSITY OF ARIZONA

Abstract

Roberto Furfaro

Systems and Industrial Engineering

Master of Science

Ontologies as Bayesian Network for Space Debris

by Stephania VASILIEVA

Space debris is a rising problem in today's world. Because there is so much in space that is unknown, it is critical to eventually catalog every piece. Since there are many attributes and properties attached to space objects, it is preferable to use an ontological classification method. The information presented in the ontology can then be used to answer questions about space debris. A Bayesian network would accomplish that because of its quantitative nature. The similarities between ontologies and Bayesian networks, such as their architectures and their flexibility, make it possible to integrate an ontology into a Bayesian network. Image determination and object collision assessment were used as applications to check the viability of integrating ontologies and Bayesian networks. It was determined that ontologies and Bayesian networks are tools that when combined can result in new useful quantitative information.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Space debris will continue to accumulate over the next century. As the problem is apparent, and organizations are starting to think about decreasing the amount of objects sent to space and how to deal with the current debris in space (European Space Agency 2016), there will be a rise in questions concerning how to get rid of the debris. Efforts are currently made to record and catalog space objects; as the number of instances in the catalogs will increase, so will the data about these objects.

With new data come new possibilities of using it. Depending on the information gathered about the newly recorded objects, original findings will be made. However, these findings need to be quantitative, and catalogs about space objects only provide qualitative information; thus the data would have to be converted into a quantitative form in order to output conclusions. Thankfully, there exists a classification method that can easily quantify the qualitative information. Ontologies can be paralleled into Bayesian networks. Space object catalogs (such as the one created by the North American Aerospace Defense Command) could be made into one ontology and converted into a Bayesian network to process the data, which would create useful results that would answer rising questions about the space debris problem.

The general research question is:

Is it possible to combine ontologies and Bayesian networks in an orbital debris application to output useful information?

1.2 Contribution

The research conducted and presented in this thesis was built on top of the existing work concerning ontologies and Bayesian networks. It extended their use into the space domain; and validated the need for a full ontology that includes information related to recorded objects.

By applying this to the space domain context, different applications of combining ontologies and Bayesian networks to increase space domain awareness and answer questions about the orbital debris issue were displayed. This was achieved by using properties from ontologies and Bayesian networks. These properties were related to existing methods, which were adapted to be applied to different types of questions related to the space domain.

1.3 Organization

This thesis is divided into 6 chapters. Chapter 2 provides background information on the research conducted; Chapter 3 presents the methodology followed to create Bayesian networks from ontologies; Chapter 4 applies the methodology and tools to specific case studies; Chapter

5 discusses the importance and results from the case studies; Chapter 6 provides a conclusion to the research completed and suggests future work ideas.

1.4 Previous Work

1.4.1 Converting Ontologies to Bayesian Networks

"OWL: A Framework for Probabilistic Ontologies" by Costa and Laskey

Costa and Laskey tackled the problem of uncertainty being non-existent in ontologies. Uncertainties had previously been ignored, but now the demand to capture it is growing. Ontologies were deemed a good method to record uncertainties because of their structure, as long as they were adapted for this context:

- All relevant uncertainties could be included in an ontology in a logical manner
- The information could easily be re-used throughout the ontology
- Reasoning could be conducted on the information

Costa and Laskey showed that even though numerical information can be added to ontologies, it is not enough to consider them probabilistic ontologies. The basic ontology language, OWL, has shortcomings because it is based on web languages that do not include uncertainty.

Costa and Laskey developed an extension of OWL, and named it PR-OWL. PR-OWL combines ontologies and Bayesian logic, to create a probabilistic ontology. The definition of a probabilistic ontology is as following (Costa):

A probabilistic ontology is an explicit, formal knowledge representation that expresses knowledge about a domain of application. This includes (Costa):

- Types of entities in the domain
- Properties of those entities
- Relationships between entities
- Processes and events involving those entities
- Statistical regularities that explain the domain
- Inconclusive, ambiguous, incomplete, unreliable, and dissonant information related to entities of the domain
- Uncertainty about all the above forms of knowledge

Costa and Laskey made it clear that a probabilistic ontology is not a probabilistic model, such as an ontology is not a database; however, probabilistic models and databases could be built on top of them.

Probabilistic ontologies are different than simple ontologies because they contain a Bayesian logic aspect, specifically MEBN (Multi-Entity Bayesian Network), in order to include probabilistic information about the classes. MEBN was chosen as a logic because it fit the flexibility vs. expressiveness trade-off that probabilistic ontologies need, since ontologies need to model a complete system but also follow the basic OWL guidelines. MEBN is a first-order Bayesian logic that produces a logical method to include uncertainties to ontologies. Similar to ontologies, it also is comprised of entities, attributes, and relationships.

The steps Costa and Laskey determined to create a probabilistic ontology are as following:

- Import into any OWL editor an OWL file containing the PR-OWL classes, subclasses, and properties
- Construct domain-specific concepts, using the PR-OWL definitions to represent uncertainty about their attributes and relationships

They added a note, that probabilistic ontologies may be “partial” if a user is only interested in one concept of an ontology. No matter how much probability is included, the ontology will still be considered extended and hold additional value because of the PR-OWL use.

"An ontology-based approach for constructing Bayesian networks" by Fenz

Fenz presented a method to convert ontologies to Bayesian networks. His method integrates both concepts to yield data and bypasses challenges implicated with Bayesian networks (Fenz 2011):

- Identification of variables that are relevant to the considered concepts (nodes)
- Identification of relationships between the identified variables (links)
- Creation of the conditional probability table for each variable

The biggest issue with using Bayesian networks is that they use a limited sample to make inferences about a population. To allay it, Fenz references Druzdzel et al., as they emphasized the need to reduce the amount of probabilities and tools to create a Bayesian network. Ontologies were considered a useful companion to Bayesian networks to help create them, because of their similar structures concerning classes and relationships.

Ontologies have similar structures to Bayesian networks, and have defined terms and properties concerning the topic of interest. Ontologies use classes and relationships to define their domains; Bayesian networks use nodes and links. Classes can be converted to nodes, while relationships can be converted to links. That implies that the domain from an ontology can then be defined in a similar manner in a Bayesian network. The two tools are not equivalent, since ontologies describe concepts, and Bayesian networks are probabilistic models. However, information can be transferred from ontologies to create a Bayesian network and its Conditional Probability Tables (CPTs) (by using probabilistic data included in the ontology).

Fenz used the four following steps to create CPTs from an ontology, as there were no straightforward existing method:

1. Select relevant classes, individuals, and properties

2. Create the Bayesian network structure

3. Construct the CPTs

4. Incorporate existing knowledge facts (observations)

Using these steps, Fenz created an ontology about security threats using Protégé and OWL; the classes ranged from specific threats to concrete steps taken in order to mitigate these threats. A Protégé plug-in converted that ontology to a Bayesian network in the Norsys Netica Bayesian Network Software. The first step to converting an ontology to a Bayesian network required human input, but the last three steps were done automatically through the plug-in and the Bayesian network software.

After generating a Bayesian network based on the ontology, and probability values were assigned to each instance, the model was evaluated by domain experts. The domain experts answered the following questions:

1. Is the final Bayesian network correct, in terms of both structure and CPTs, to answer the posed question?
2. Does the final Bayesian network output useful information to the user?

To check the validity of the final Bayesian network, the experts checked the nodes and links to their respective classes and relationships in the ontology. They concluded that the structure was correct, the method presented by Fenz was useful to its user because of its ease of use, and that it provided useful information about security threats. But even though the method was validated, there still exists a need to improve the human bias while creating CPTs.

1.4.2 Space Ontologies

"Ontology of Astronomical Object Types" by IVOA

The International Virtual Observatory Alliance (IVOA) took it upon itself to define a complete astronomical objects ontology that can evolve. It was deemed important to create such an ontology, because it would facilitate the consistency of information shared between human groups and machines. The ontology is able to include basic concepts and advanced inferences, which makes for an automatic consistency check. In this case, the ontology created was based on the SIMBAD (Set of Identifications, Measurements, and Bibliography for Astronomical Data) Astronomical Database (upkept by the "Centre de données astronomiques de Strasbourg" in France), allowing it to receive consistency checks from the acquired data in the database. The ontology was created using OWL-DL (Description Logic) in the Protégé editor because of its trade-off between complexity and expressiveness. The reasoner chosen to infer new information was Racer 1.7.24, but then Pellet was concluded to be better.

To create a working ontology, IVOA had to decide on a scope, define all of the appropriate concepts, build the relationships between those concepts, evaluate the completed ontology through reasoners and expert opinion, and maintain it through real use. All of these steps are part of analysis, building the ontology, evaluation, and maintenance.

The ontology was built from information that was considered useful. An issue that arose, was the need to introduce classes that were not astronomical objects themselves, but had strong relations with them. The solution to that was to create a separate defined class for those entities. The reasoner then completed the evaluation. The reasoner compiled the ontology to check for consistent logic, and completeness of the conditions assigned its classes.

The final ontology consisted of the following top-level classes – both primitive and defined:

- AstrObject

- AstroPortion
- AtomicElements
- ClassificationCategory
- EMSpectrumRange
- Measurement
- Morphology
- Process

The expectations from this ontology are that it continues growing, and applications may be developed to use the ontology more efficiently.

"An ontological architecture for orbital debris data" by Rovetto

Rovetto proposed ontologies as a method for space agencies to share information. Ontologies can encourage data-sharing, inter-operability and knowledge discovery; which all could minimize the rising orbital debris global hazard. To create the ontology, Rovetto stated the importance for all agencies recording and tracking objects to be involved. However, currently there is limited data-sharing between organizations for a copious amount of data. There is also a lot of inconsistencies with the terminology used between the agencies. Rovetto's ontology tackled these issues with these specific goals:

- Have an Orbital debris and SSA domain representation
- Specify and formalize terminologies/vocabularies that are faithful to the respective scientific domain

- Annotate debris and SSA data Orbital debris and SSA data-sharing
- Contribute to: orbital debris remediation, and the improvement of SSA and space safety.

Rovetto asserts that other ontologies may be integrated if it they are deemed related. Some of these additional ontologies may be "Reference" or "Data" ontologies. The final ontology had these top-level classes:

- Types of orbital debris
- Properties/features of debris
- Sensors observing the orbital environment
- Sensing and measurement processes
- Relations relating debris to other relevant entities
- Debris processes, including causes of debris formation
- State and Positional information

The architecture presented a method to improve overall space awareness, by providing a way to present information about space. This ontology could grow by including more classes, as well as computable specifications.

1.4.3 Using Ontologies and Bayesian Networks

"Data enrichment of spatial databases using ontologies and Bayesian networks" by Alexander Melchior

Melchior presented a combination of ontologies, classifiers, and Bayesian networks in order to enrich data for map generalization. Data enrichment is known as a tedious effort, thus there is a large interest to automate the data enrichment process. Ontologies were used because they can contain decision making information, and can transfer that information into a Bayesian network, which is the actual probabilistic model.

Defined classes were created into an ontology, and then translated into a Bayesian network. These classes either represented one specific element from a picture, or a combination of elements. A concern Melchior expressed was that the Bayesian network would not be able to represent dynamic relationships (one-to-many relationships) - e.g. a node would not make the distinction between one or ten houses.

The method to go from an ontology to image determination is divided into four parts:

1. Natural language
2. Formal models of spatial concepts
3. Algorithms
4. Database enrichment

Steps 1 and 2 were the basic steps to create the ontology; while steps 3 and 4 used it.

Melchior used a residential area with a lot of trees as a use case to validate his proposed steps. There was a focus on houses surrounded by trees for ease of analysis. An ontology was created based on elements included for a house surrounded by trees. The classifier then made assumptions from the data. The generated Bayesian network then calculated the percentage of each element to actually be in the image.

The results were compared with the real-life locations of these images. The method showed a 56% image determination accuracy, even though by manipulating the classes looked at the number grew to 90%. This accuracy means that the method presented in this paper can determine the elements constituting an image.

Chapter 2

Background

2.1 Space Domain Awareness and Orbital Debris

There is a big space debris problem (European Space Agency 2013); whether it be from previously active satellites becoming inactive, parts detaching from active objects, or collisions and explosions between objects in space (NASA Orbital Debris Program Office).

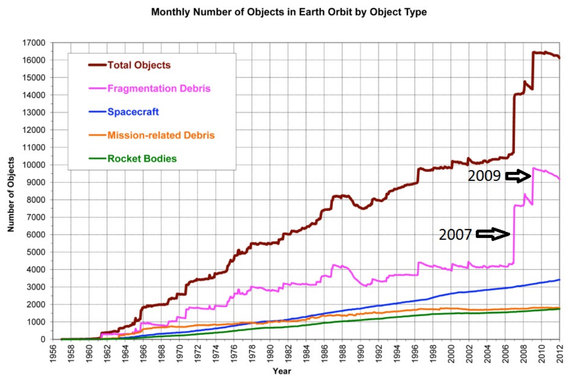

FIGURE 2.1: Monthly Number of Cataloged Objects in Earth Orbit by Object Type (NASA Orbital Debris Program Office)

Figure 2.1 shows the current state of objects in space, and how it is only expected that these numbers will grow as time passes as more objects, ranging from cube-satellites to rockets, will be sent out.

Not only will these objects be sent out by companies and organizations, but also from high schools that send out satellites through weather balloons (Woods 2010) and now have the opportunity to launch CubeSats to put in orbit (Walker 2016).

Even though the objects are small in physical size, they still can have a big impact on big objects due to their velocities (NASA Orbital Debris Program Office 2012). There are currently over 21,000 orbital debris that are larger than 10 cm; there are 500,000 estimated particles between 1 and 10 cm, and over 100 million estimated particles smaller than 1 cm (Figure 2.2). The two main sources attributed to the start of the space debris trash in Low Earth Orbit (LEO) are (Tallis 2014):

- The 2007 anti-satellite missile (ASM) that destroyed the Fengyun weather satellite, but consequently created a lot of new smaller orbital debris.
- The 2009 collision between the American satellite Iridium 33, and the Russian inactive satellite Kosmos-2251.

These two spikes in orbital debris are witnessed in Figure 2.1

These debris are considered a huge threat because they are unrecorded. No information or trajectory is known about them, and so even by being very small, they can have a critical impact on bigger objects due to their velocity. Active satellites have shields to protect against orbital debris, and bigger ones can maneuver around them, but it is not enough, as the shields can eventually break or the object won’t have enough time to move around the debris (NASA Orbital Debris Program Office 2012). Radars can only identify objects that are least 5 cm in diameter in size in low-orbit, and 1 meter in

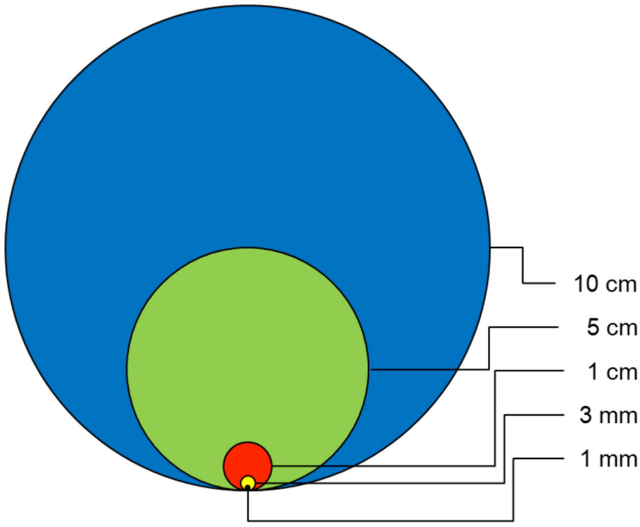

FIGURE 2.2: Debris Size Class Comparison (The Aerospace Corporation)

diameter in geosynchronous orbit (NASA 2013). According to studies conducted by The Aerospace Corporation, the number of smaller untraceable debris will continue growing for the next century. Figure 2.3 shows the state of orbital debris in 2012 from a GEO point of view after almost 60 years of space missions.

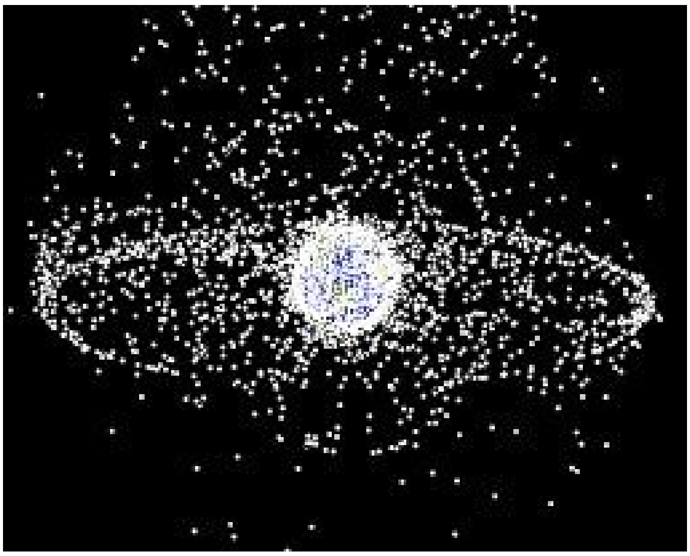

FIGURE 2.3: GEO view of debris location surrounding the Earth (NASA Orbital Debris Program Office 2012)

This high number of debris also instills fear since there are humans in space aboard the ISS. In June of 2011, the six astronauts in the ISS had to take shelter in an escape craft due to debris passing dangerously close to them (Tallis 2014). Having so much debris, regardless of the size, is dangerous due to the speed they are going because any kind of collision will produce considerable energy.

There are steps being made to try to clean up space. The European Space Agency (ESA) has decided to dedicate specific missions to space debris removal (ESA 2012). Another step in helping the orbital debris problem has been contracted out to Lockheed Martin by NASA. Lockheed Martin is creating Space Fence, an advanced, ground-based radar that will detect debris with greater accuracy. Space Fence is planned to be tested and completed in late 2018 (Lockheed Martin).

These are ongoing efforts in the space debris issue, and to increase their usability it is also important to increase space domain awareness. Recording and knowing what is in space would:

1. Let the European Space Agency (ESA) determine which objects are more important to remove
2. Extend the use of Space Fence
3. Hopefully increase international collaboration in order to solve the space debris problem

To help increase space awareness, a proposition is to have a classification system for all debris, such as an ontology (Cambrésy 2009).

2.2 Ontology

“An ontology defines a set of representational primitives with which to model a domain of knowledge or discourse”

-Gruber

An ontology is a classifying system. It is a method which incorporates more information about a system than for example a taxonomy - a hierarchical classification system. Both methods have a hierarchical order to them, but ontologies take it a step further. Ontologies show existing relationships between different classes, and assigns object and/or data properties to classes and/or instances. Figure 2.4 uses the ontology language "OWL" to give a visual representation of what an ontology consists of, and how its classes and properties relate. The green nodes stand for classes, while the blue and orange nodes represent object and data type properties respectively. The colors are used to differentiate between the different concepts of OWL, and are not part of the language. The links connecting the nodes display parent-to-child relationships, and specify which nodes relay properties (object or data) and the direction that information has.

Object properties connect two classes and specify how one class is related to another. For example, following Figure 2.4, the class "Person" causes the class "Event". Thus,the object property connecting "Event" to "Person" could be “hasPerson”. The second type of property is data properties. Those tie-in classes to specific data values of any unit (integer, string, etc.).

Properties can also have the following characteristics (Protege-OWLTutorial):

- Functional

  There can only be one range for that property.

- Inverse

  The property has an opposite direction than another property with the same domains and ranges.

- Transitive

  The relationship connects the domains with all of the ranges.

- Symmetric

  The relationship can relate a domain to a range and a range to a domain without losing its meaning.

- Asymmetric

  The relationship can only have one direction from the domain to the range.

- Reflexive

  The relationship can have the same domain and range.

- Irreflexive

  The relationship cannot have the same domain and range.

These relationships can also be linked to instances, which are specific examples of a class.

Another aspect of ontologies that makes them stand out, is the conditions required to fully model a system. Classes may have "Necessary Conditions" as well as "Sufficient Conditions". Necessary conditions specify that a class always satisfies the conditions stated - these are referred to as defined classes. Sufficient classes allow the ontology to do some inferences by using a reasoner. A reasoner will look at the necessary conditions of that class and decide whether it should be contained somewhere else; these are referred to as primitive classes.

Ontologies are coded through OWL. It is a language that eases machine understanding of the web, and adds vocabulary and formal semantics. There are three basic sub-languages:

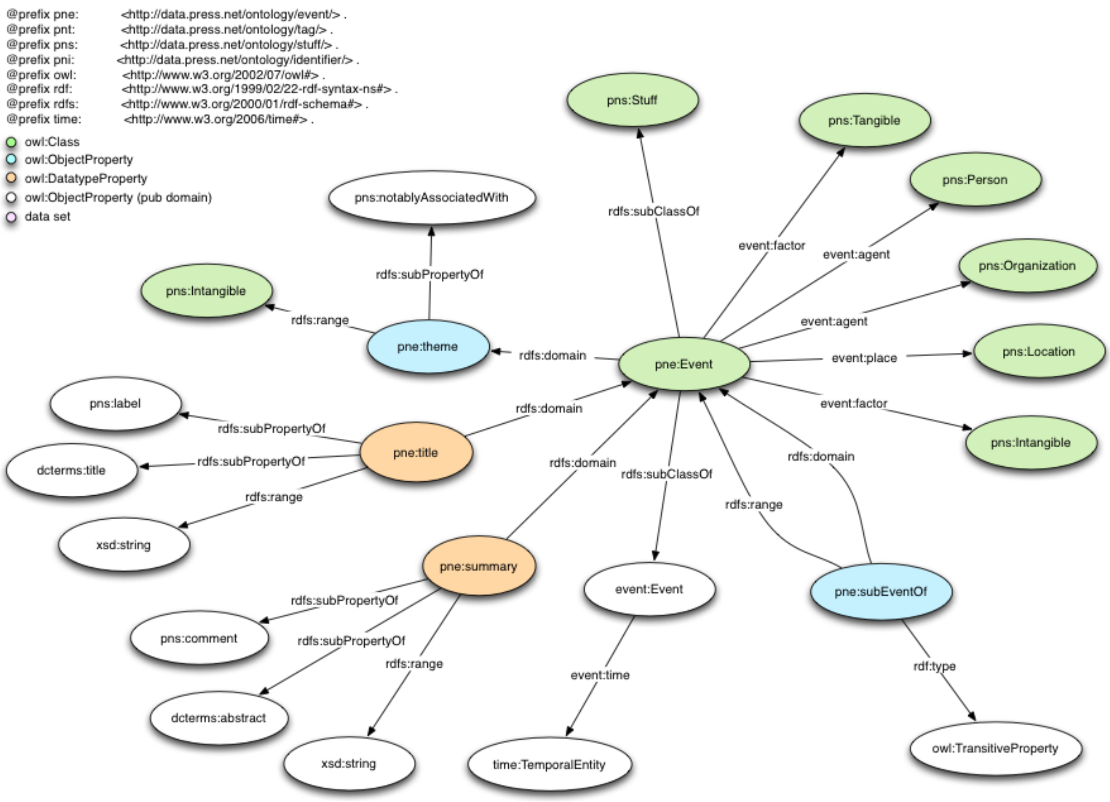

FIGURE 2.4: "Event Ontology" (SNaP Event Ontology)

- OWL Lite

  This version is the most common used one, has restricted in vocabulary, and has a frame-like notation.

- OWL DL

  This version requires for OWL to be typed out like a coding language.

- OWL Full

  This version is the most complex level of OWL, and provides the most freedom out of all three version.

The different sub-languages represent different levels of complexity of the language depending on the user's needs.

2.2.1 Ontology Software

There are many programs that easily allow a user to create and compile ontologies, the most popular one being Protégé. Protégé has a user-friendly interface that allows the user to create the needed classes and links for a certain system, and then add conditions and specific properties to create a complete classification model. The software incorporates OWL through its interface in order to make it easy for the user to create a model; instead of typing each function to create a new class or relationship, the user usually just has to click on a "+" button. Visuals of the Protégé interface are included in Chapter 4.

2.3 Bayesian Networks

Bayesian networks are used as an extension of ontologies in this context. They are probabilistic or deterministic acyclic graphical models (Directed, Acyclic Graph - DAG) that can compute the chance of a node (or event) happening. Their applications include: diagnosis, prediction, financial risk management/portfolio allocation/insurance, ecosystems modeling, sensor fusion, and monitoring and alerting. They are useful because of their flexibility, and easy management and maintainability. They consist of nodes and links. The nodes can be classified as parent or child nodes and store information. These nodes can represent random variables ranging from observable quantities to unknown hypotheses, and have to be mutually exclusive and exhaustive. On the other hand, links represent relationships between the nodes (or conditional dependencies), and show the direction of the model.

From a mathematical perspective, Bayesian networks are represented in the following manner (Vomlel):

- An acyclic graph G = (V, E)
- Each node i $\in$ V corresponds to a random variable X_i with a finite set X_i of mutually exclusive states
- p_a(i) denotes the set of parents of node i in graph G
- each node i $\in$ V corresponds to a Conditional Probability Table (CPT) (Formula 2.1)

$$P(X_i|(X_j)j \in p_a(i)) \tag{2.1}$$

Formula 2.1 specifies that the probability of event X_i given event X_j is included to compute the total probability of event i. X_i represents one possible outcome of event i, while X_j represents events stimulate the instance X_i to happen.

Bayesian networks have a very understandable graphical interface, which makes it easy and intuitive to understand the conditional properties reflected in the network. The interface shows nodes' dependence and independence, or d-separation, between each other through their link relationships (Bellandi 2008). If there is a pair of unconnected parent nodes that means that they have a conditional independence. Figure 2.5 reflects that relationship.

Table 2.1 shows the deterministic conditional probability combinations that need to be inputted for Figure 2.5. The total number of combinations is based on the number of parents, *m*, a node has; there is a total of 2^m combinations, since each node holds a Boolean value (it can either be True or False). The child node can either have quantitative probabilities (probabilistic) as inputs or True/False (deterministic) in the CPT as shown.

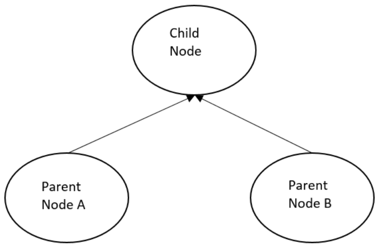

FIGURE 2.5: Simple Bayesian Network

Parent A	Parent B	Child
T	T	T
T	F	F
F	T	F
F	F	T

TABLE 2.1: Simple CPT

Bayesian networks can perform both deductive and inductive reasoning. It can follow the direction of the links, or go backwards. As an example, following a simple Bayesian network representation in Figure 2.6, if the "Child" node's CPT is computed and given an observation, the "Parent" nodes' probabilities are affected (Figure 2.7). Conversely, if observations are added to the "Parent" nodes, the "Child" node is affected (Figure 2.8). Observations refer to new probabilistic information available about a node; it may be positive or negative. Positive observations are specific and means that that the finding is definite; negative observations are ones that state what the node definitely is not.

Due to Bayesian logic and how the networks are built, Bayesian networks can do the following inference tasks:

- Belief Updating

  Propagates probabilities for all nodes to compute new ones

- Finding most probable explanation

  Changes the values in nodes to achieve the most probable configuration

- Finding maximum a-posterior hypothesis

  Uses observations for samples to make predictions about a population

- Finding maximum-expected-utility decision

  Weighs different outcomes and computes which one has the highest value

As a result of those tasks, Bayesian networks are able to combine knowledge and data, efficiently represent the inferences, make incremental learning, handle missing data, and learn causal relationships.

The architecture of a Bayesian Network is similar to one of an ontology, in that there are parent and child nodes (Figure 2.6) that are linked. In the Bayesian network context, the values of the parent nodes are transferred through the links - instead of object/data properties.

There are multiple kinds of connections seen with Bayesian networks:

- **Serial** This is a simple step-by-step connection (Figure 2.9). A -> B -> C

  Also referred to as "causal chains". This relationship means that the probability of C is unchanged whether the given condition is B or AB; which implies that it does not matter what A

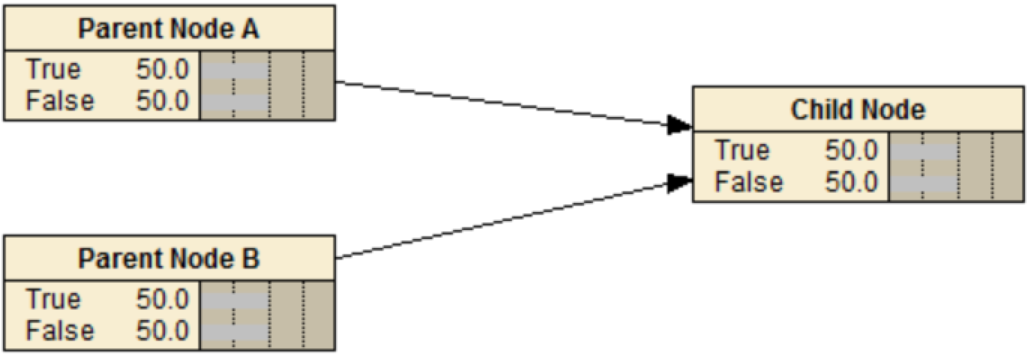

FIGURE 2.6: Parent-Child example relationship in Bayesian Network, using the Netica software interface

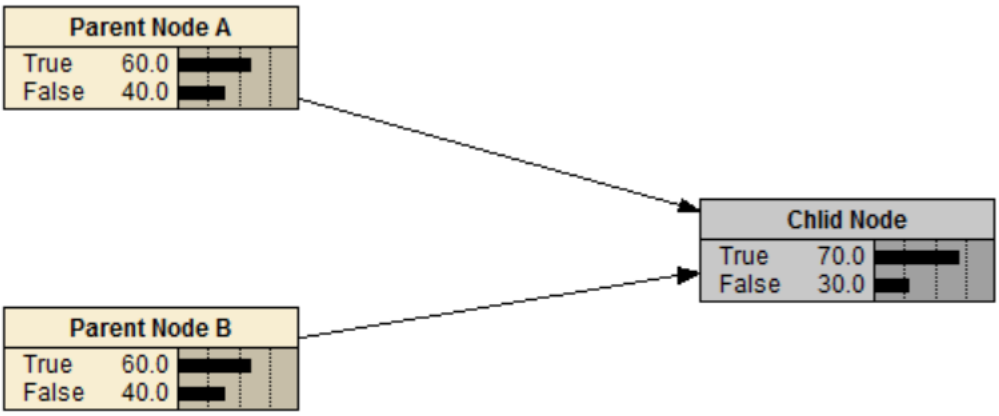

FIGURE 2.7: Simple BN with "Child" CPT and observation

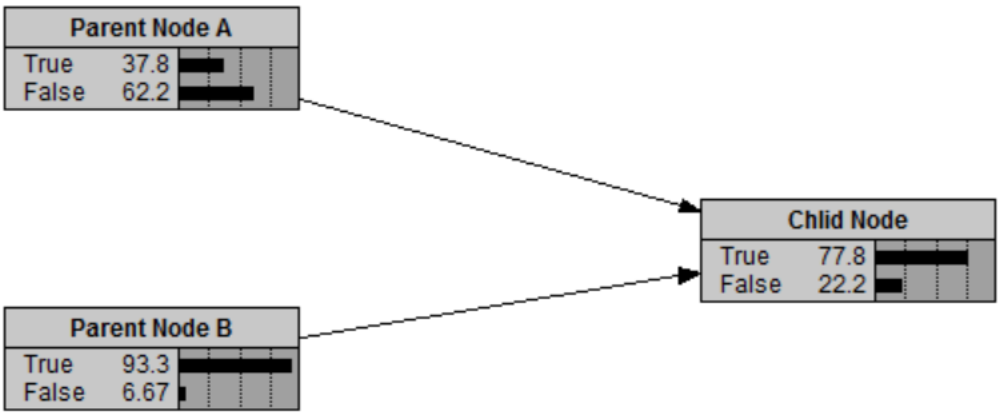

FIGURE 2.8: Simple BN with Parent observation, and Child CPT and observation

is if we already know that B occurred (by the Markov property).

It is a conditional independence.

- **Diverging** This connection represents one parent node with multiple children nodes (Figure 2.10).

  A <- B -> C Also referred to as “common causes”. This relationship means that if B is unknown, but A or C is given an observation, then B will rise, which in turn will increase the probability of the other child node. However, if B is known, then adding an observation to one of the child node will affect the other child node.

- **Converging** This connection represents one child node with multiple parent nodes (Figure 2.11).

  A -> B <- C

  Also referred to as “common effects”. This is the opposite of a diverging connection, and represents conditionally dependent nodes. In this case, if A is given an observation, the other node’s, C, probability will decrease.

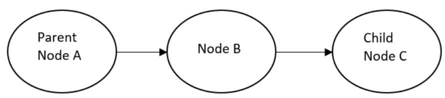

FIGURE 2.9: Serial representation of Bayesian network structure

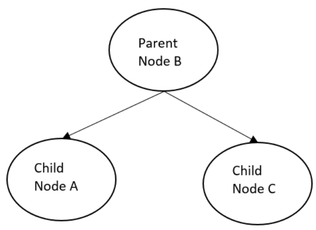

FIGURE 2.10: Diverging representation of Bayesian network structure

These relationships are used by the d-separation concept to explain the dependencies in a network; conversely, if nodes are not

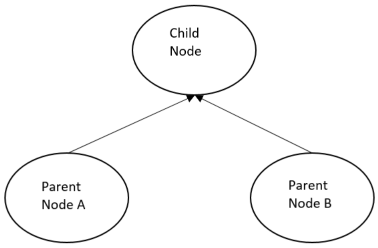

FIGURE 2.11: Converging representation of Bayesian network structure

d-separated, they are said to be d-connected. D-separation occurs to a path by a set of nodes Z if and only if:

- The path is causal and the middle node is in Z
- The path is diverging and the middle node is in Z, or
- The path is converging and the middle node (and any of its descendants) is not in Z

Nodes and d-separations create formal definition for Bayesian networks:

- DAG is said to be an independence (I-map) map of a probability distribution if all of its conditional independencies represented are valid and correctly defined by d-separation (Formula 2.2). A DAG is said to be a minimal I-map if no links can be removed without also removing the I-map property.

$$\langle A|B|Z\rangle_G \Rightarrow I\langle A, B, Z\rangle_M \tag{2.2}$$

The left side of Formula 2.1 represents the d-separation between A and B given Z, while the right side defines d-separation as conditional independence. G and M are the respective domains for each concept.

These connections specify the probabilistic model. The model is solved through conditional probability; which means:

Given an event *b*, what is the probability of event *a* happening?

Event *a* may be the grass being wet, while there could be multiple events *b*s, such as rain or the sprinklers being on.

In a Bayesian network context, the value of *b* may either be a number or have the value "True" or "False" depending on whether *b* is set as deterministic or probabilistic. The probabilities are calculated in the background and based on Bayes rule (Formula 2.3).

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \tag{2.3}$$

Bayes rule computes the conditional probability of an event, by taking into account P(B | A) as the reverse conditional probability, and P(A) and P(B) as the independent probabilities for each event.

Besides the probabilities, the conditional probability tables need to be computed. CPTs are critical elements of Bayesian networks, as they assist in calculating the final probability of an event to happen. They consist of specific probabilities for each scenario a node and its parents are in. They are accessed through the children node, and created by parameters and/or constraints.

The following formulas compute CPTs (Fenz 2011):

$$C(S_{x_1}, ..., S_{x_n}) = \begin{cases} 1, & \prod_{i=1}^{n} S_{x_i} \neq 0 \lor T = 0. \\ 0, & \prod_{i=1}^{n} S_{x_i} = 0 \land T = 1. \end{cases} \tag{2.4}$$

$$P(N|X_1, ..., X_n) = \left(\frac{S_{x_1}}{h_{x_1}} + w_{x_i}\right) + ... + \left(\frac{S_{x_n}}{h_{x_n}} + w_{x_i}\right) * C(S_{x_1}, ..., S_{x_n}) \tag{2.5}$$

Formula 2.4 states whether a certain node will have an impact (value of 1) or not (value of 0) on its child. It is decided on the current value of the state and whether the parent nodes do not always a direct effect on its children nodes (represented by T). If the parent node has a non-zero state and fully affects the child node, then it is given a C value of 1. Otherwise, if the parent node has a zero state value or does not have a full effect on its children node, then the parent node is given a C value of 0. Bayesian networks do not account for partial relationships between nodes, and thus this step takes into account that parent nodes may not always be relevant to a child node.

Formula 2.5 computes the probability of the child node (N) happening, given its parent nodes X_n. It says that given the parent nodes, the ones that are have a state value (S_x) of 1 (nodes considered True) and are always related to their children nodes (C value), need to have their values added. Adding those values computes the final probability for each specific instances in a CPT. These values need to add up to 1.

CPTs are part of joint probability distributions (Formula 2.6), with the use of conditional properties on the nodes. Formula 2.5 states that, by showing conditional probabilities included in computing the final probability of event i happening, for node X.

$$P((X_i)i) = \prod{}_{v=1}^{n} P(X_i|(X_j)j \in p_a(i)) \tag{2.6}$$

Through a complete Bayesian network, both the model and the user can make inferences and learn about the system in question.

Chest Clinic Bayesian Network Example by Netica

This Bayesian network is based on the Asia network of Lauritzen and Spiegelhalter (1988). A patient comes in with chest pain. What is the probability that he/she/they has: tuberculosis, lung cancer, or bronchitis? Figure 2.12 shows the Bayesian network created to answer the question. The percentages for the diagnostics are affected by whether the patient has traveled recently or whether he/she/they is a smoker. Following the network, someone that has traveled recently has a higher percentage of having tuberculosis, but it says nothing about lung cancer or bronchitis. Observations can also be added to the x-ray result nodes, which by properties of Bayesian logic, will affect the "Tuberculosis or Cancer" node. The "Tuberculosis or Cancer" node is not necessary to answer the question presented, but since tuberculosis and cancer have similar relationships, the node simplifies the final network. Through propagation, changing one node will affect all others.

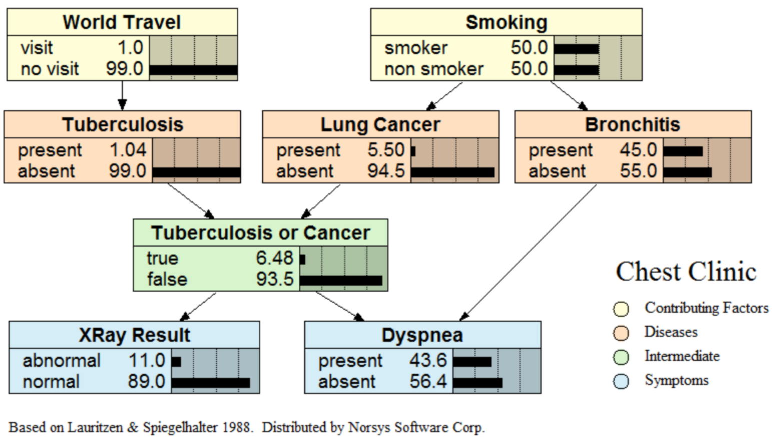

FIGURE 2.12: Chest Clinic Bayesian Network

2.3.1 Bayesian Network Software

There are many programs that easily allow a user to create and compile Bayesian networks, the most popular free limited version one being Netica. Netica has a user-friendly graphical interface that allows the user to create the needed nodes and links by dragging-and-dropping, as well as providing an easy method for inputting observations and CPTs for each node. Netica also offers other tools to aid in decision making.

Netica increases the power of Bayesian networks, by making it easy to conduct both deductive and abductive reasoning. It combines the graphical interface with its quantitative aspects by allowing the user to directly give a node a state. Doing so directly computes the probabilities of both that node's children and parents due to deductive and abductive reasoning.

2.4 Combining Ontologies and Bayesian Networks

Ontologies and Bayesian networks have similar architectures. They can be built with the same elements and relationships, and represent a model of a system. Because of their architecture similarities, it is possible to construct a Bayesian network based on an ontology. Classes and relationships from the ontology can be paralleled into a Bayesian network (Fenz 2011). The ontology contextually simplifies the understanding of how to build the Bayesian network by already having the classes, relationships, values, and weights predetermined. By inputting the classes and properties needed, it is then possible to create a probabilistic network that can compute probabilities of a node being true (e.g. the probability of an event happening).

An ontology consists of classes and relationships, which are the equivalent of nodes and links in a Bayesian network. The major disparity between both concepts, is how an ontology contains qualitative information, while Bayesian networks are quantitative. OWL extensions that include probabilistic properties are introduced, and adjust the information from ontologies to be quantitative:

- BayesOWL

  BayesOWL directs how to create a Bayesian network from the already existing OWL vocabulary. The structure of the ontology is first translated into the Bayesian network, then the CPTs are computed. The probabilistic values that determined the CPTs appear in the ontology as object and/or data properties in the ontology. The ontology is used as a storing space, where the information is then used and manually converted to provide quantitative information.

  BayesOwl is advantageous because there is no need to modify a current OWL ontology, but it fails to use all of the ontology since it only considers classes and ignores instances, specific data types, and properties.

- OntoBayes

  OntoBayes is a Bayesian network that is constructed based on the OWL format (Yang 2005). It lets a Bayesian network comment an ontology with its probabilistic properties. There are three kinds of probabilities that are transferred: PriorProb, CondProb, and FullProbDist. These deal with unconditional, conditional, and join probability distribution tables.

OntoBayes is advantageous because it has the ability to process multi-values random variables, since the flow goes from Bayesian network to ontologies.

- PR-OWL

  PR-OWL is an extension of the OWL language and is based on Bayesian logic. It adds new definition and expands the OWL vocabulary to allow the inclusion of quantitative properties to ontologies. It allows to create a probabilistic model from an ontology, instead of just a deterministic one (Costa 2005).

  PR-OWL is advantageous because it incorporates uncertainty directly in the language, and can reproduce repeated structures in a Bayesian network (it can specify whether someone has exactly 1 or 10 cats). However, PR-OWL is still a work in progress, and is not currently directly compatible with OWL.

Chapter 3

Methodology

3.1 Producing Bayesian networks through ontologies

Changing an ontology into a Bayesian network is straight-forward due to the their similarities in structure through BayesOwl. The principal steps are (Fenz 2011):

1. Selection of relevant classes, individuals, and properties
2. Creation of the Bayesian network structure
3. Construction of the Conditional Probability Tables (CPTs)
4. Incorporation of existing knowledge facts

Classes will be chosen from an ontology based on a given question. Once chosen, they are created as nodes in a Bayesian network. The links that connects these nodes in the Bayesian network come from relationships in the ontology. Anything related to the question being asked needs to be included in the ontology. The links can range from representing an "is_A" relationship to an object property relationship such as "participatesIn". The only difference between the ontology and Bayesian network structures resides in how the link directions in an ontology go from top-to-bottom, while a Bayesian network is acyclic, and so any kind of direction can be set-up.

Once the Bayesian network structure is complete, its CPTs need to be computed. There are two states of CPTs, chance (probabilistic) or deterministic. A probabilistic CPT adds up the weights for all "True" parent nodes; a deterministic state only needs to be given a "True" or "False value based on the combinations of parents nodes. Afterwards, the observations are inputted to any relevant nodes. These observations affect nodes, as they are considered extra information that have an effect on the CPTs.

Since the observations and weights are not values required in making an ontology, it is critical to include them in an ontology for the sake of conversion, and are specified as "Weights" and "Values". Ultimately, all of the information used by the Bayesian network is taken from the ontology, so it is unnecessary to find information from an outside source.

Chapter 4

Case Studies

4.1 Overview

Since space debris is accumulating faster than the problem is being solved, it is important and pertinent to find out if an object is a debris and how dangerous it is in space. This method of converting an ontology into a Bayesian Network can be helpful in giving an idea of how dangerous recorded objects are.

4.2 Orbit Classification

There is a need for a complete space ontology, including space debris, that shows any and all information pertinent to space. One manner to create one all-encompassing ontology is by combining multiple smaller ones. These smaller ontologies can also be used on their own and manipulated into Bayesian networks. One of these small ontologies happens to be straight-forward and is a "Classification of Orbit Classes" (Figure 4.1). It only has "is_a" relationships, which reflects a hierarchical structure. The translation of this ontology to a Bayesian network demonstrates how closely related both architectures are.

Orbit has a total of six different kind of orbits:

- Positive Energy Orbit (unbounded)

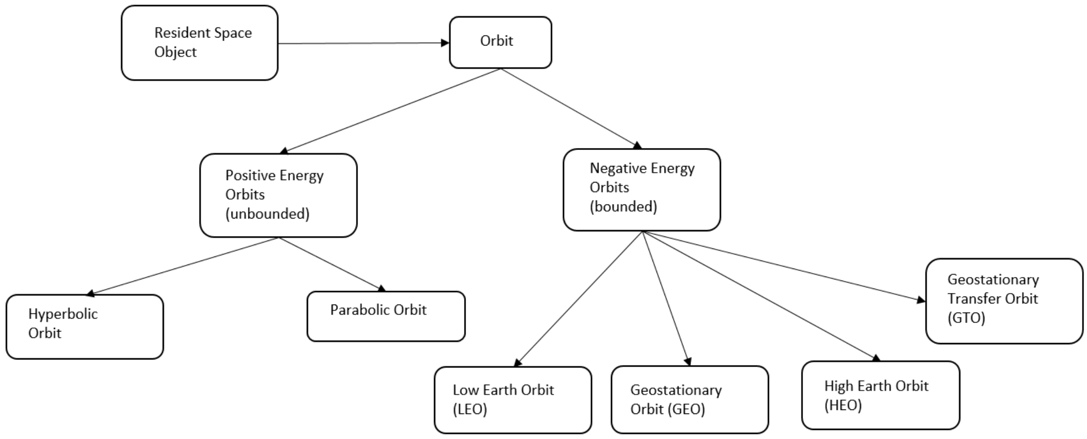

FIGURE 4.1: Classification of Orbits

  - Hyperbolic Orbit
  - Parabolic Orbit

- Negative Energy Orbit (bounded)
  - Low Earth Orbit (LEO)
  - Geostationary Orbit (GEO)
  - Geostationary Transfer Orbit (GTO)
  - High Earth Orbit (HEO)

To convert from the ontology to its corresponding Bayesian network, the classes are first paralleled with their links representing the hierarchy (Figure 4.2). The links in the Bayesian network only represent a parent-child relationship, and so the CPTs for “Negative Energy Orbit”, "Positive Energy Orbit" and “Orbit” are deterministic instead of probabilistic. The “Negative Energy Orbit” and "Positive Energy Orbit" node will always be True as long as one of their parent nodes is also True (Figure 4.3); the same goes for the node “Orbit”. The Bayesian network outputted the percentages representing that property, as seen in the shaded classes. These percentages were then propagated to compute the "Orbit" probability, combined with the

CPT for "Orbit". There is no need for probabilistic data to be imported in this context, unless they were observations about the most-parent nodes.

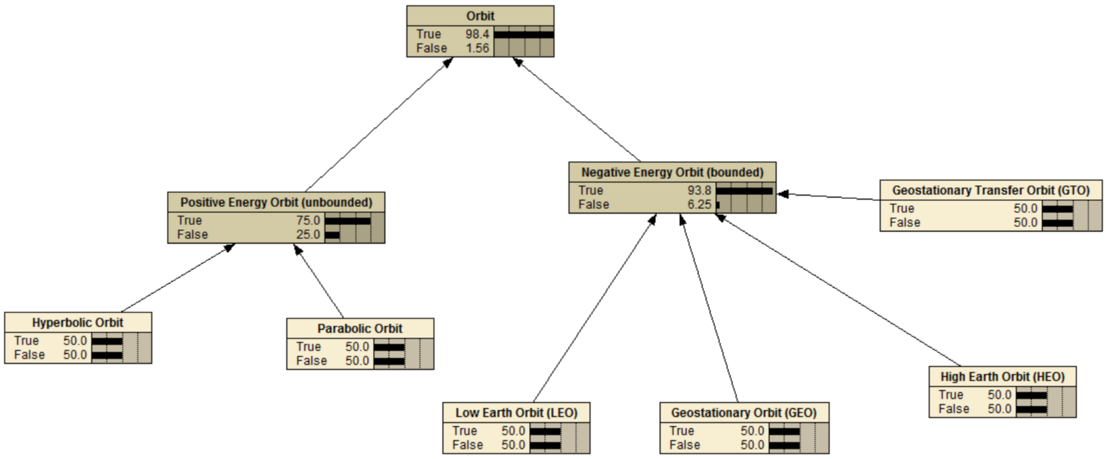

FIGURE 4.2: Classification of Orbit Classes ontology translated into a Bayesian Network

Hyperbolic Orbit	Parabolic Orbit	Positive Energy Orbit
True	True	True
True	False	True
False	True	True
False	False	False

FIGURE 4.3: Positive Energy Orbit Deterministic CPT

In Figure 4.2,

This application does two things:

1. Computes the likelihood of an object to have one of the six specific orbits

2. Shows how similar the architecture between ontologies and Bayesian networks is

3. Shows that quantitative results can be outputted using information from a qualitative ontology

4.3 object Identification

4.3.1 Object Identification Purpose

An ontology can give a framework and a map for a Bayesian Network. Depending on the given values, weights, and properties of the chosen classes, the Bayesian network can determine the likelihood of a space object being a debris. The purpose of an ontology and a Bayesian network in this context is to help determine what an unknown, but identified object may be.

4.3.2 Ontology prepared for Object Identification

The ontology created was based on a taxonomy (Figure 4.4).

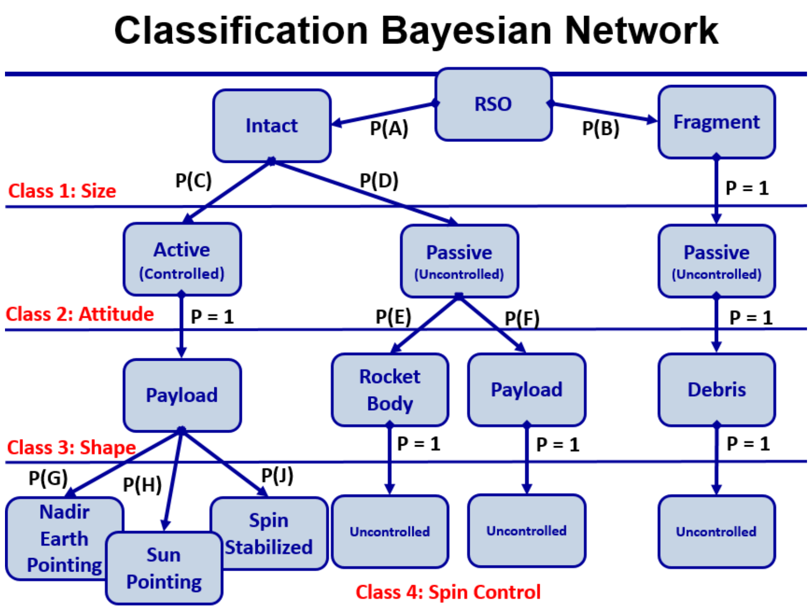

FIGURE 4.4: Taxonomy Representation of Space Object Classification (Linares)

The classes in the taxonomy list different objects and whether they are controlled or not. Based on their attitude property, their spin is specified. If the object is "Active", then it has three spin control options:

- Nadir Earth Pointing

  The object is facing downward to view the Earth.

- Sun Pointing

  The object is facing the Sun.

- Spin Stabilized

  The object spins around a fixed axis.

It was converted from the taxonomy to an ontology in the following manner:

1. The object's physical properties were put into one class (RSO), and were given an object type

   - RSO
     - Fragment
       * Debris
     - Intact
       * Payload
       * Rocket

2. Any other properties (object or data) were put in sibling classes

   - Attitude
     - ActiveAttitude
     - PassiveAttitude
   - Spin Control
     - NadirEarthPointing
     - SpinStabilized
     - SunPointing

– Uncontrolled

Figure 4.5 provides a visual representation of what it looks like in Protégé. The "PassiveUncontrolledRSO" class is an amalgamation of RSOs that have both passive and uncontrolled properties.

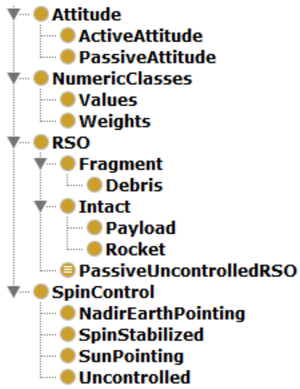

FIGURE 4.5: Protégé view of ontology

Separating the type of properties created a more complete classification system and used ontology aspects fully. Creating two separate broad classes integrated hierarchical taxonomy elements. It easily paralleled the connections between the classes without repeating elements. Instead, it attributed multiple properties to the elements - such as specifying that all RSO sub-classes have an attitude and/or spin control. An extension of the ontology is that it allows the user to add data to those properties. Data properties were added to non-physical properties, as simulations were run to predict the attitude, position, and velocity of an object.

The "NumericClasses" class in the ontology includes weights and values chosen for each RSO. These values are only used in order to convert the ontology to a Bayesian network, since the ontology has more of a qualitative form and the Bayesian network is quantitative. The values are taken from simulations.

Figure 4.6 represents all of the relationships in the ontology. The lines without annotations represent "is_a" relationships. All other lines represent object properties between the classes; RSO also has a data property represented by a circle. The different colors represent the relationships between the classes ("is_a", "hasAttitude", and "hasSpinControl"). The relationships between the sub-classes are not represented because those are included in the ancestor relationships that are shown; it is implied they are passed down.

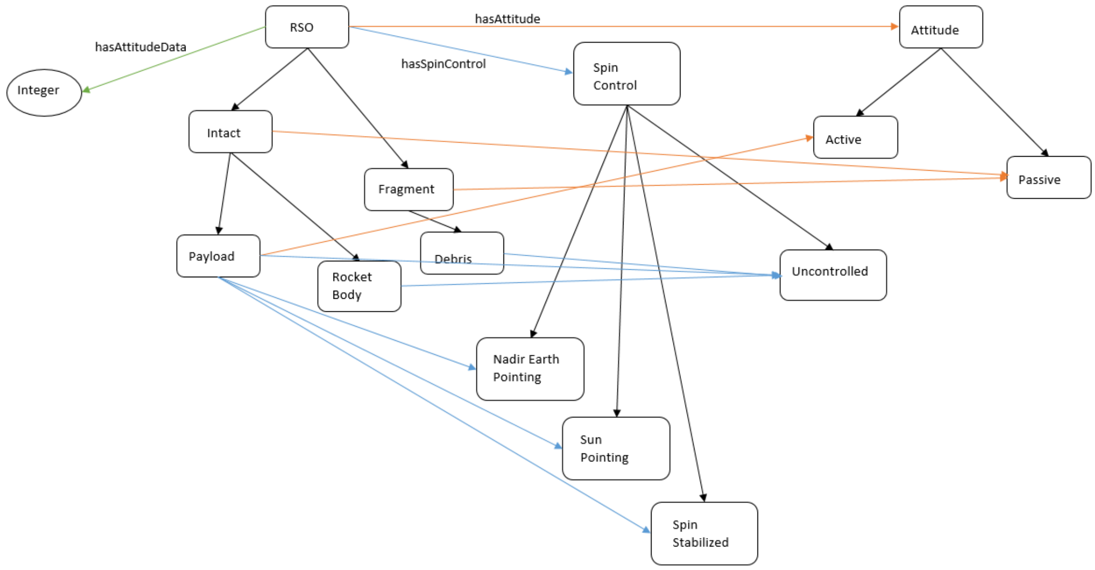

FIGURE 4.6: High-level diagram of Ontology

4.3.3 Bayesian Network prepared for Object Identification

This application of transforming an ontology to a Bayesian Network creates links that represent parent-child relationships between properties. Based on information about a detected object and simulations on it, a classifier determines how much of each property that object has (on a probabilistic case), and then, based off of the links, the

Bayesian Network computes the probability of this object being an active satellite or a debris.

To successfully convert the ontology to a Bayesian Network, it is required to have weights for each subclass as well as values to fill the Conditional Probability Tables (CPTs). These weights and values are derived from models, and then added into the ontology for completeness and easy translation into the Bayesian Network.

The Bayesian Network was created by having the parents be the "Intact" and "Fragment" classes. Those classes can be decomposed into other classes, and can also be coupled into a “PassiveUncontrolledRSO” class.

Based on models and simulations, an object will either have a controlled spin, or be listed as having an uncontrolled spin. Therefore, the “Uncontrolled Spin” node in the Bayesian Network is deterministic instead of by chance like the other nodes. The node’s parents can only have binary values concerning its uncontrolled spin state.

Figure 4.7 shows the architecture of the transformed ontology-to-Bayesian network. The CPTs for each child class have been determined based on the ontology relationships. They are not optimal, as the information does not come from models; however, they show that the Bayesian network can compute probabilities for each object class due to its acyclic property. If a node is shaded gray, it means that an observation was added for that node. These observations, just like the CPTs, have been determined based on the possible relationships between these classes. They can be better determined from simulations. The Bayesian network outputted the probability of each type of object being true. Based on the CPTs and observations, it is most likely that an unknown object is a rocket or a debris.

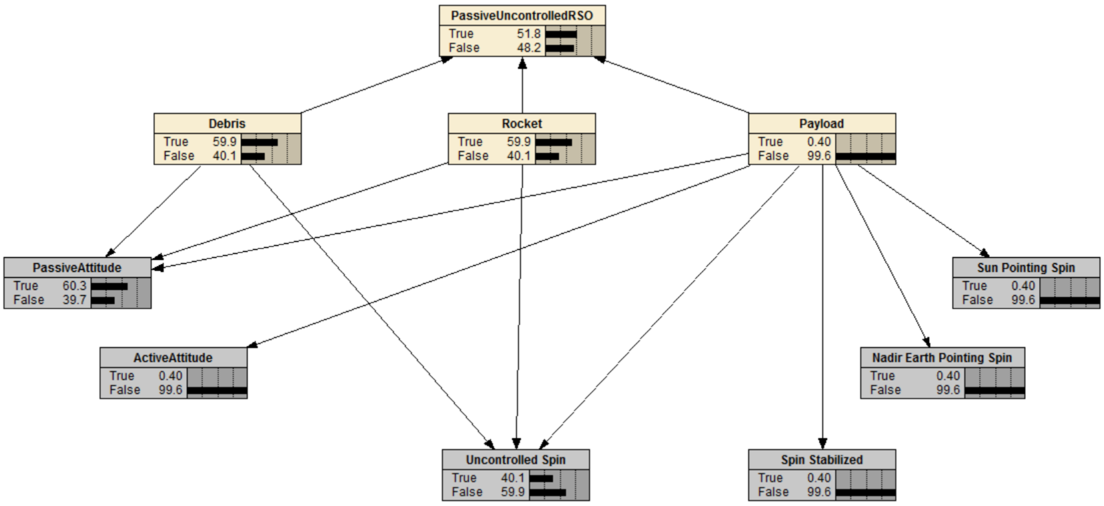

FIGURE 4.7: Bayesian Network skeleton for ontology based Object Identification Bayesian Network

4.4 Collision Threat

4.4.1 Collision Threat Set-Up

Using an ontology proposed in "An ontological architecture for orbital debris data" (Rovetto 2015) as inspiration, a space debris ontology was created. Following "Ontology-to-Bayesian Network" (Fenz 2011) methods, that ontology was then transformed into a Bayesian Network to calculate the probability of a collision between debris and active satellites, as well as their threat.

4.4.2 Ontology prepared for Collision Threat

A simplified ontology was created by using "An ontological architecture for orbital space debris" as inspiration. It was simplified; some classes were omitted, because they did not pertain to the final construction of the Bayesian network for this application. The architecture chosen let the ontology be potentially used with a Protégé plugin that can automatically create a Bayesian Network based on the ontology itself. Figure 4.8 lists the classes created for this application.

Important data to define debris threat in a Bayesian network are (Scientific and Technical Subcommittee of the United Nations Committee on the Peaceful uses of Outer Space 1999):

- apogee
- perigee
- radar cross-section
- period

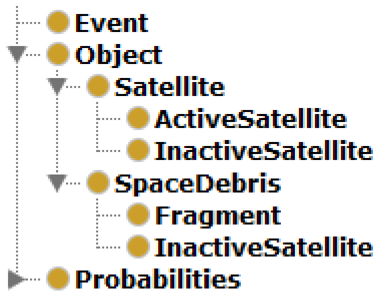

FIGURE 4.8: List of classes pertaining to a space debris ontology to be converted to a Bayesian Network

Other properties are included in the ontology, but do not have their own personal class; instead, they are listed in the Two-Line Element (TLE) of each instance, which is then added as a string data property related to each instance. Some properties not listed in the TLE, such as the instances' launch date, are recorded as other data property. Figure 4.9 shows the properties for a specific instance "1993-036LBQ".

An additional class was added to the ontology, called "Probabilities." This class is used by a Protégé plugin that automatically generates a Bayesian network. There are two sub-classes that provide

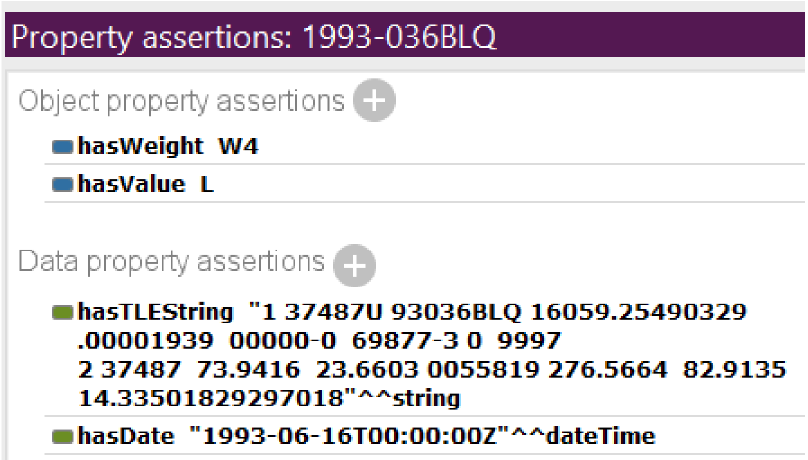

FIGURE 4.9: Property Assertions of instance 1993-036LBQ, where the relationships between the instance and its properties are created.

values and weights to the instances. The "Values" subclass consists of specific chances of an instance itself being a threat based on a criteria. There are 5 different values that stand for:

VL (Very Low) = 10%

L (Low) = 30%

M (Medium) = 50%

H (High) = 70%

VH (Very High) = 90%

For example, the specific space debris instance "1993-036BLQ" has an "L" value of 0.3. The value was chosen because the fragment is small (based on its radar-cross section), has a regular period, and has an apogee and perigee that are not too high (>8000km).

The weights' purpose is to calculate the threat probabilities of the instances' super-classes (or the parent classes). The instances' weights need to add up to 100%. This specifies which instance for that particular class is more or less of a threat. There are 3 possible weights for the active satellites class, and 3 possible weights for the debris class:

W_1 = 18% (active satellites)

W_2 = 13% (active satellites)

W_3 = 15% (active satellites)

W_4 = 15.5% (debris)

W_5 = 18% (debris)

W_6 = 20% (debris)

The 3 choices for both the "Satellite" and "SpaceDebris" classes represent a high, medium, and low value. The weights were chosen based on the amount of instances the classes have, and based on the risk assessment completed from the classes' properties (which were decided on the same way as the "values" class). Since it is required that all of them add up to 100%, they have a balanced difference between them. The weights were assigned to the instances based on the instances' already given values.

Figure 4.10 shows the final ontology classes and relationships, with "1993-036A" as an instance example, with weight "W_4" which is equal to "15.5".

4.4.3 Bayesian Network prepared for Collision Threat

Since all of the required information to create a Bayesian Network was included in the created ontology, it became possible to auto-generate a Bayesian network by using a plugin, or to manually create the Bayesian Network from the ontology. In both cases, the ontology serves as a place to store all of the needed classes, properties, and relationships.

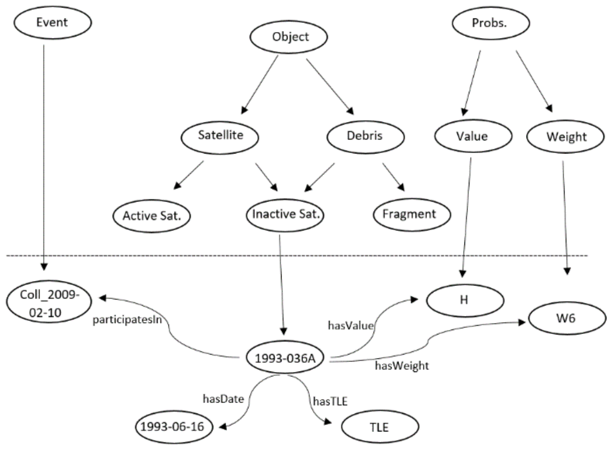

FIGURE 4.10: Structure of the Space Object ontology for instance 1993-036A. The structure shows that the instance is connected to other instances from other classes through object properties relationships. It also has connections to the data properties: date and TLE.

To manually create the Bayesian network, the nodes were first created for the Bayesian Network based on the classes from the ontology. In this application, the classes range from specific instances, to the threat level of the instances' superclasses ("Active Satellites" and "Debris"), and to the final child "Collision." That set-up calculates the collision probability of the objects in space, and allows the user to see if the satellites or debris classes have an impact risk.

Besides the classes, properties were also imported from the ontology. If the properties were not involved in calculating the collision risks, they were excluded from the quantitative aspect of the Bayesian network and included in the description of its respective class. As an example, the nodes representing instances have their respective TLEs in their description. The properties "Value" and "Weight" were however used as inputs to the Bayesian Network. "Values" was used as an observation input for each instance, while "Weight" was used for the children nodes' CPTs. Figure 4.11 shows instance "1993-036A" and its value inputted as an observation, while

Figure 4.12 shows parts of the probability table for “Debris Threat”, where the "Weight" were used. The percentages for “True” and “False” in Figure 4.13 were computed by adding the weights for each instance that was listed as “True” following Formula 2.4.

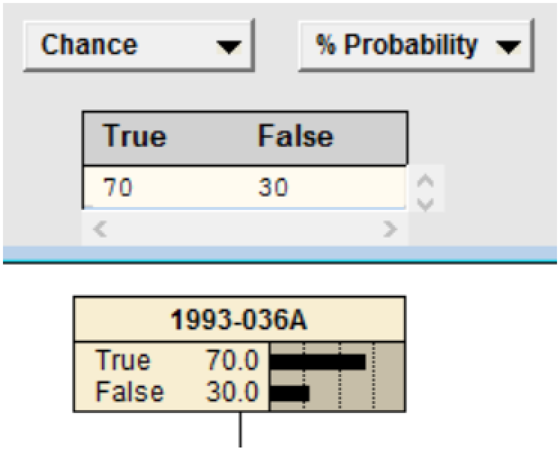

FIGURE 4.11: Instance “1993-036A” in Bayesian Network with its respected value

1993-036A	1993-036BLQ	1993-036BLR	1993-036BLV	2002-005J	2014-089C	True	False
True	True	True	True	True	True	100	0
True	True	True	True	True	False	82	18
True	True	True	True	False	True	84.5	15.5
True	True	True	True	False	False	66.5	33.5
True	True	True	False	True	True	84.5	15.5
True	True	True	False	True	False	66.5	33.5
True	True	True	False	False	True	69	31
True	True	True	False	False	False	51	49
True	True	False	True	True	True	84.5	15.5
True	True	False	True	True	False	66.5	33.5
True	True	False	True	False	True	69	31
True	True	False	True	False	False	51	49
True	True	False	False	True	True	69	31
True	True	False	False	True	False	51	49
True	True	False	False	False	True	53.5	46.5
True	True	False	False	False	False	25.5	74.5
True	False	True	True	True	True	84.5	15.5
True	False	True	True	True	False	66.5	33.5
True	False	True	True	False	True	69	31
True	False	True	True	False	False	51	49
True	False	True	False	True	True	69	31
True	False	True	False	True	False	51	49
True	False	True	False	False	True	53.5	46.5
True	False	True	False	False	False	35.5	64.5
True	False	False	True	True	True	69	31

FIGURE 4.12: Probability table for node “Debris Threat”

In Figure 4.13, the table for “Collision” is deterministic instead of probabilistic, and so “Collision” can only be true if both “Satellite Threat” and “Debris Threat” are true; otherwise, a collision may not happen (both classes need to have a collision risk).

Active Satellite Threat	Debris Threat	Collision
True	True	True
True	False	False
False	True	False
False	False	False

FIGURE 4.13: Deterministic table for risk of “Collision”

Once all of the CPTs were calculated, the finished result was a Bayesian network that calculated the risk for a collision. Based on the instances, values, and weights given the Bayesian network, it computed a collision risk of 28.8% (Figure 4.14), while showing the "Debris Threat" as having a risk of 44.4% due to their high speed and small size. Their impact on active satellites is reduced though, because the active satellites are provided with shields and maneuverability, which are reflected in their values and weights.

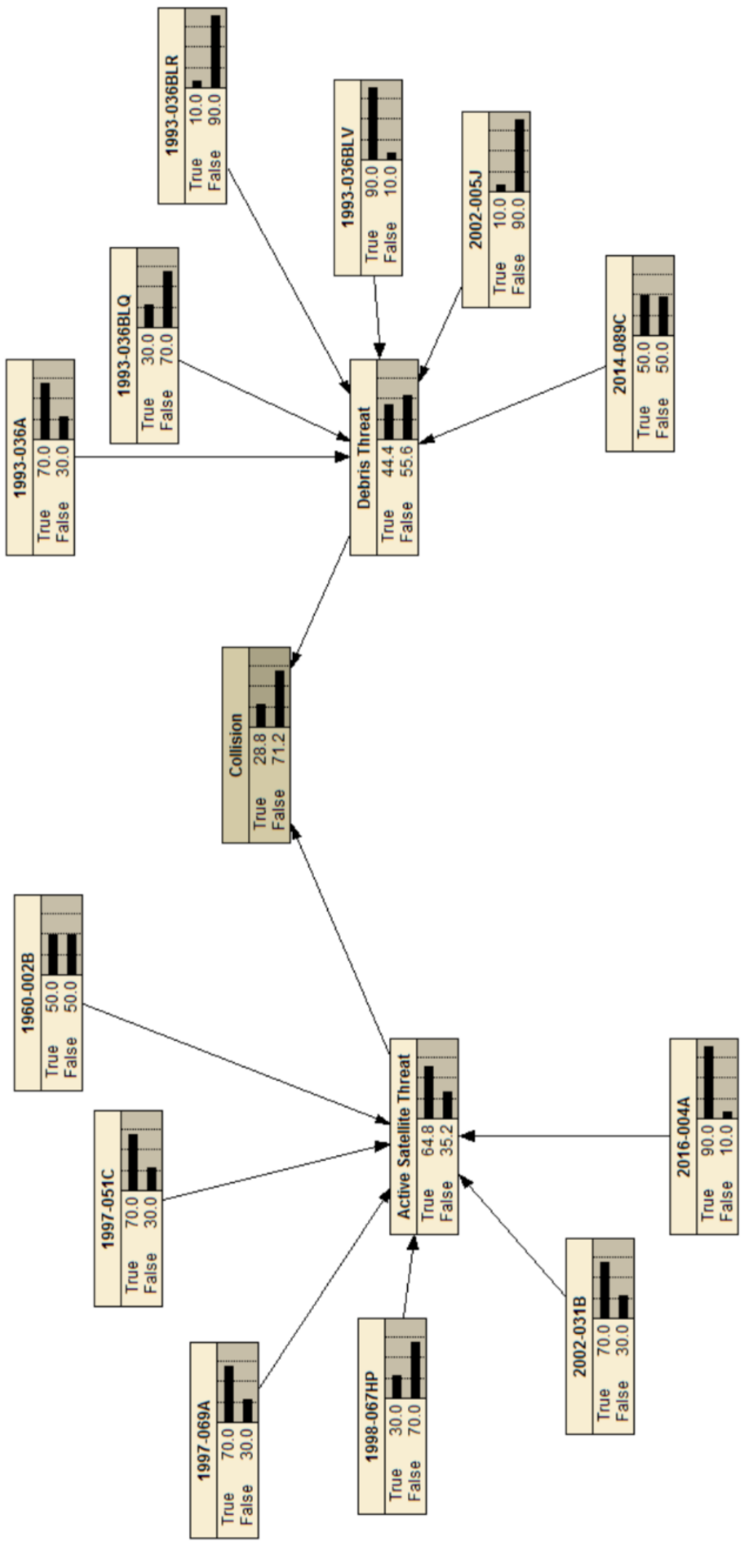

FIGURE 4.14: Final Bayesian Network with probabilities of each node being a risk

Chapter 5

Discussion

Ontologies are shown to be effectively used in Bayesian networks to solve questions related to orbital debris. The architecture of ontologies provides a clean transition to the creation of Bayesian networks. BayesOwl was used to go from one form to the other. The appropriate classes and their respective probabilistic values and weights were set-up in both Protégé and Netica. The transfer of information was used in three ways:

- To show the transition from ontology-to-Bayesian network
- Space debris detection based on an image
- To compute the probability of collisions

The first application specifically showed how similar the architectures between the two tools are. Converting the ontology to the Bayesian network for orbit classes set up a base to extend ontologies to include probabilistic information. The transformation fits in with the current methods which specify the criticality of paralleling the classes and relationships from ontologies to the nodes and links of Bayesian networks.

The ontology-to-Bayesian network for image identification provided an understanding of how a taxonomy can be manipulated to create an ontology in order to solve a problem in a Bayesian network. The classes ranged from objects to properties that had their own respective nodes. Doing so, we adapted previous methods to a space awareness application. The kind of questions answered were not directly related to finding the probabilities of the states of the classes, but instead solved for a combination of them. By manipulating known methods, new questions can be asked and answered.

Next, an ontology was manipulated transferring information through the links in a Bayesian network. The Bayesian network to solve for a collision probability showed through four instances that debris hold a high threat. That threat is reduced due to active satellites having shields and maneuverability. The collision threat accounts for both known and unknown debris. Depending on the instances of the ontology, the collision probabilities will change. The end goal is to create a complete database from an ontology, and auto-generate a Bayesian network. This application integrated all of BayesOWL. It was a straight-forward use of the methodology, and resulted in:

1. Extracting quantitative information from the qualitative data
2. Answering space awareness questions using the ontology's structure

The combination of ontologies and Bayesian networks to solve space debris questions showed to be effective and straight-forward in multiple aspects. It allowed for there to be a flow of information from a qualitative aspect to compute new quantitative information.

Chapter 6

Conclusion and Future Work

Ontologies show themselves to have a beneficial structure to record information about space debris. They can improve space awareness, and can be used by other tools. Integrating ontologies with Bayesian networks can answer questions by making use of the information available. Orbital debris are increasing, and so are questions and fears pertaining to them. The merging of ontologies and Bayesian networks opens new doors to develop solutions to the space debris problem and raise space awareness.

As the number of orbital debris increases, and the threat they provide rises, it will become more important to have a database with all of the available information about these debris, and have an easy way to transform that information into quantitative data. Since ontologies provide a complete form to classify and create a database, it can grow as detected objects get recorded. As the information database will grow with both property information about the objects and data from simulations, it will become possible to continuously feed the data from the ontology to Bayesian networks to compute up-to-date results with new data. This extension would rely heavily on having a copious amount of data, and completely paralleling the Bayesian networks to their respective ontology architecture.

References

Agency, European Space. *ESA PLANNING MISSIONS TO TAKE OUT SPACE TRASH*.

— (2013). *Space Debris*.

— (2016). *Clean Space*.

Bellandi, Andrea (2008). "Extending Ontology Queries with Bayesian Network Reasoning". PhD thesis. IMT Institute for Advanced Studies.

Corporation, The Aerospace. *SPACE DEBRIS BASICS*.

Costa, Paulo C. G. da (2005). "Bayesian Semantics for the Semantic Web". PhD thesis. School of Information Technology and Engineering, George Mason Univer-sity.

D. Mehrholz L. Leushacke, W. Flury R. Jehn H. Klinkrad M. Landgraf (2002). *Detecting, Tracking and Imaging Space Debris*.

F. Johansson, G. Falkman (2008). "A Bayesian network approach to threat evaluation with application to an air defense scenario". In: *Information Fusion, 2008 11th International Conference on*.

Fenz, Stefan (2011). "An ontology-based approach for constructing Bayesian networks". In: *Elsevier*.

Gruber, Tom (2009). *Ontology*.

Jarred McGinnis Paul Wilton, Paul Harman (2012). *SNaP Event Ontology*.

John L. Crassidis Barry Smith, Christopher K. Nebelecky Ronald Rudnicki Moriba Jah William A. Tagliaferri (2015). *A Space Object Ontology for Space Domain Awareness Applications*.

Kathryn Laskey Ghazi Alghamdi, Xun Wang. *Detecting Threatening Behavior Using Bayesian Networks*.

Kehe Wu, Shichao Ye (2014). "An Information Security Threat Assessment Model based on Bayesian Network and OWA Operator". In: *Applied Mathematics and Information Sciences An International Journal*.

Kumar Ravi, Sheopujan Singh. "Bayesian Network for Uncertainty Representation in Semantic Web: A Survey". In: *International Journal of Computer Applications Technology and Research*.

L. Cambrésy S. Derriere, P. Padovani A. Preite Martinez A. Richard (2009). *Ontology of Astronomical Object Types*. Tech. rep. International Virtual Observatory Alliance.

Martin, Lockheed. *Lockheed Martin's Space Fence Program on Track to Start Tracking Space Junk*.

Melchior, Alexander (2013). "Data enrichment of spatial databases using ontologies and Bayesian networks". MA thesis. University Utrecht.

NASA (2011). *Definition of Two-line Element Set Coordinate System*.

— (2013). *Space Debris and Human Spacecraft*.

Office, NASA Orbital Space Debris Program. *Orbital Debris Photo Gallery*.

— (2012). *Orbital Debris Frequently Asked Questions*.

Paulo C. G. COSTA, Kathryn B. LASKEY (2016). *PR-OWL: A Framework for Probabilistic Ontologies*.

Quratulain Rajput, Sajjad Haider. *BNOSA: A Bayesian Network and Ontology based Semantic Annotation Framework*.

Recommendation, W3C (2009). *OWL Web Ontology Language Overview*.

Rovetto, Robert J. (2015). "An ontological architecture for orbital debris data". In: *Springer-Verlag Berlin Heidelberg*.

Scientific and Technical Subcommittee of the United Nations Committee on the Peaceful uses of Outer Space (1999). *Technical Report on Space Debris*. Tech. rep. UNITED NATIONS PUBLICATION.

Stefan Fenz, A Min Tjoa. *Ontology- and Bayesian-based Threat Probability Determination*.

Tallis, Josh. *Lockheed wins contract to track space trash*.

Unlimited, Tethers. *The Orbital Debris Problem*.

Vomlel, Jiri. *Some applications of Bayesian networks*.

Walker, Theresa. "Sky's the limit for Irvine high schoolers as they seek to send a satellite into orbit". In: *The Orange County Register*.

Woods, Tori. *Students Send Balloons to the Stratosphere*.

Yang, Yi. "OntoBayes: An Ontology-Driven Uncertainty Model". In: *International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce*.

Zhongli DING Yun PENG, Rong PAN (2006). *A Bayesian Approach to Uncertainty Modelling in OWL Ontology*.