PR-OWL: A Framework for Probabilistic Ontologies

Paulo C. G. COSTA, Kathryn B. LASKEY
The Volgenau School of Information Technology and Engineering
George Mason University
4444 University Drive
Fairfax, VA 22032-4444 USA
[pcosta, klaskey]@gmu.edu

Abstract. Across a wide range of domains, there is an urgent need for a well-founded approach to incorporating uncertain and incomplete knowledge into formal domain ontologies. Although this subject is receiving increasing attention from ontology researchers, there is as yet no broad consensus on the definition of a probabilistic ontology and on the most suitable approach to extending current ontology languages to support uncertainty. This paper presents two contributions to developing a coherent framework for probabilistic ontologies: (1) a formal definition of a probabilistic ontology, and (2) an extension of the OWL Web Ontology Language that is consistent with our formal definition. This extension, PR-OWL, is based on Multi-Entity Bayesian Networks (MEBN), a first-order Bayesian logic that unifies Bayesian probability with First-Order Logic. As such, PR-OWL combines the full representation power of OWL with the flexibility and inferential power of Bayesian logic.

Keywords. Probabilistic Ontologies, PR-OWL, MEBN, Bayesian networks, uncertainty, Semantic Web, knowledge sharing

Introduction

Since its adoption in the field of Information Systems, the term ontology has been given many different definitions. A common underlying assumption is that classical logic would provide the formal foundation for knowledge representation and reasoning. Until recently, theory and methods for representing and reasoning with uncertain and incomplete knowledge have been neglected almost entirely. However, as research on knowledge engineering and applications of ontologies matures, the ubiquity and importance of uncertainty across a wide array of application areas has generated consumer demand for ontology formalisms that can capture uncertainty.

Although interest in probabilistic ontologies has been growing, there is as yet no commonly accepted formal definition of the term. We demonstrate that augmenting an ontology to carry numerical and/or structural information about probabilistic relationships is not enough to deem it a probabilistic ontology. This paper proposes a formal definition based on the core notion that a probabilistic ontology formalism should provide the means to express all relevant uncertainties about the entities and relationships that exist in a domain in a logically coherent manner. This would not only provide a consistent representation of uncertain knowledge that can be reused by different probabilistic systems, but would also allow applications to perform plausible reasoning with that knowledge. We also introduce PR-OWL, an extension of the Web Ontology Language OWL that provides a consistent framework for building probabilistic ontologies. PR-OWL combines the expressive power of OWL with the flexibility and inferential power of Bayesian logic.

1. Basics of Representing and Reasoning under Uncertainty

1.1. Why should we care about uncertainty?

OWL has its roots in its own web language predecessors (i.e. XML, RDF), and in traditional knowledge representation formalisms that have historically not considered uncertainty. Examples of these formalisms include Frame systems [1] and Description Logics, which evolved from the so-called "Structured Inheritance Networks" [2]. This historical background somewhat explains the lack of support for uncertainty in OWL, a serious limitation for a language expected to support applications in uncertainty-laden domains such as biogenetics or medicine.

Although OWL itself is focused on the Semantic Web, by extending it to become a probabilistic-aware language we are tackling a problem that predates the current WWW: the quest for more efficient data exchange. Clearly, solving that problem requires more precise semantics and flexible ways to convey information. While the WWW provided a new presentation medium and technologies such as XML presented new data exchange formats, both failed to address the semantics of data being exchanged. The SW is meant to fill this gap. Realization of its goals will require major improvements in technologies for data exchange. However, since virtually all current ontology formalisms are based on classical logic, SW languages such as OWL provide no consistent support for uncertainty representation or plausible reasoning.

This lack of support for uncertainty can be justified in closed systems designed to perform well-defined tasks, for which clear and unambiguous vocabularies can be constructed. But the semantic web vision requires heterogeneous systems to interoperate in an open world. Inevitably, vocabularies that are adequate for a single stand-alone application break down when required to interoperate with systems employing different vocabularies originally tailored to different tasks. Inevitably, there is incomplete and partial overlap of terminology and concepts. Even when concepts are clearly defined, in an open-world system available inputs may be insufficient to determine which meaning is most appropriate. For example, a standard ontology might enumerate different senses for the word "Washington," such as the United States as an agent, the first President of the United States, a state in the Pacific Northwest, or a baseball team. Semantic Web applications employing the ontology must identify which of these senses is most appropriate in a given context, e.g., when the word is embedded in the sentence, "Washington voiced strong objections to the proposed policy." As another example, the developers of an ontology for military planning [3] identified over a dozen different doctrinal uses of the word "clear" within the United States Department of Defense [4]. In complex open-world problems, legislating unambiguous usage is often infeasible. Several items of evidence in combination may be required to disambiguate among different meanings of a given term. Evidential reasoners require information about the strength of association between items of evidence and the conclusions to which they point, as well as contextual factors that affect the strength of evidence.

We argue that the ontology layer is the appropriate place in the Semantic Web architecture for representing declarative knowledge about likelihood. That is, in environments in which noisy and incomplete information is the rule, likelihood information is a key aspect of domain knowledge, and should be included in formal domain ontologies. A counter-argument has been made that probability (with the possible exception of microscopic quantum phenomena) is epistemic, but formal ontology should represent phenomena and relationships as they exist in the world. Carried to its extreme, however, this philosophical stance would preclude the use of virtually every ontology that has yet been developed. To explore this idea further, we note that if computational ontologies had existed in the 17th century, Becher and his followers might well have developed an ontology of phlogiston. We may chuckle now at their naïveté, but who among our 17th century predecessors had the foresight to judge which of the many scientific theories then in circulation would stand the test of time? Researchers in medicine, biology, defense, astronomy, and other communities have developed a plethora of domain ontologies. It is virtually certain that at least some aspects of some of these ontologies will, as human knowledge progresses, turn out in retrospect to be as well founded as the theory of phlogiston. Shall we outlaw use of all these ontologies until the day we can prove they contain only that which is ontological, and nothing that is mere epistemology? We take a pragmatic stance that although our ultimate objective is to seek the truth about Reality as it is, that ultimate objective is unattainable in the lifetime of any human. Nevertheless, it is necessary and desirable to do the best we can with the knowledge we have. To pretend certainty when we are uncertain is not doing the best we can. Formal ontology provides a useful means of communicating domain knowledge in a precise and interoperable manner, and of extending and revising our descriptions as human knowledge accrues. To do this in a sound and principled manner requires a sound and principled way to represent, communicate, and reason with uncertainty. Probabilistic ontologies provide a means of doing so.

Not surprisingly, as ontology engineering research has achieved a greater level of maturity, the need for representing uncertainty in ontologies in a principled way has become more and more clear. There is increasing interest in extending traditional ontology formalisms to include sound mechanisms for representing and reasoning with uncertainty.

1.2. Probabilistic ontologies

In general, people faced with the complex challenge of representing uncertainty in languages like OWL tend to begin by writing probabilities as annotations (e.g. marked-up text describing some details related to a specific object or property). This is a palliative solution that addresses only part of the information that needs to be represented. Over the past several decades, semantically rich and computationally efficient formalisms have emerged for representing and reasoning with probabilistic knowledge (e.g., [5]-[6]). Annotating a standard ontology with numerical probabilities is just not enough, as too much information is lost to the lack of a good representational scheme that captures structural constraints and dependencies among probabilities. A true probabilistic ontology must be capable of properly representing those nuances. More formally:

Definition 1 (from [7]): A probabilistic ontology is an explicit, formal knowledge representation that expresses knowledge about a domain of application. This includes:

- Types of entities that exist in the domain;
- Properties of those entities;
- Relationships among entities;

- Processes and events that happen with those entities;
- Statistical regularities that characterize the domain;
- Inconclusive, ambiguous, incomplete, unreliable, and dissonant knowledge related to entities of the domain; and
- Uncertainty about all the above forms of knowledge;

where the term entity refers to any concept (real or fictitious, concrete or abstract) that can be described and reasoned about within the domain of application. ■

Probabilistic Ontologies are used for the purpose of comprehensively describing knowledge about a domain and the uncertainty associated with that knowledge in a principled, structured and sharable way, ideally in a format that can be read and processed by a computer. They also expand the possibilities of standard ontologies by introducing the requirement of a proper representation of the statistical regularities and the uncertain evidence about entities in a domain of application. It is important to emphasize that a probabilistic ontology is not a probabilistic model (e.g. a model built using applications such as Netica, Hugin, or Quiddity*Suite), in the same way that an ontology is not a database application.

The differences in the in-depth underlying concepts and technologies supporting ontologies and database schemas are not easily distinguishable, as the real differentiation between the two resides in their respective intended purposes. Ontologies represent domains in a way that should facilitate interoperability with other representations of that domain (i.e. other ontologies build by different people with different views and interests) or of domains that are not directly related but share some concepts. When a database solution for a given domain is conceived, its primary focus is not in representing all concepts of a domain in a way that makes it interoperable with current or future views of that domain, but in defining the concepts of that domain which would enable storage and retrieval of the information the database stakeholders (and their customers) want to store and retrieve, in a way that best fits their requirements.

In a similar vein, when a probabilistic model is built to solve (say) a radar data fusion problem, the main interest driving its creators is not in making sure that their definitions about radar domain concepts are interoperable with other definitions that might exist for those same concepts. In contrast, interoperability would definitely be a primary focus when building a probabilistic ontology for the domain of radar data fusion. Ontology engineers would attempt to express one view of that domain in a way that others (with possibly different views) may use/understand and thus build applications (databases, decision systems, etc) that are compatible with anything built under that view.

Furthermore, it is not necessary for an ontology to be a running database, yet a database application can be built on top of an ontology. Likewise, a probabilistic ontology does not necessarily need to be a running probabilistic model, yet a running probabilistic model (i.e. an executable application built using a probabilistic package) can be built on top of a probabilistic ontology if that fits the objectives of the application at hand. A subtle difference here is that anything built on top of a traditional ontology can be built on top of a probabilistic ontology, but the converse is not always true, since the latter is an extension of the former that adds the above mentioned features of a probabilistic framework.

1.3. MEBN: The Probabilistic Logic of PR-OWL

To comply with interoperability requirements and at the same time to enable probabilistic model to be built on top of its definitions, a probabilistic ontology has to be based on a very flexible logical foundation. When searching for that framework, we realized that there will always be a trade-off between flexibility and expressiveness among the candidate probabilistic logics. After some careful research (see [7] and [8] for details) we found that MEBN logic [5] provides a particularly attractive trade-off that made our work easier when extending the OWL Semantic Web language.

MEBN is a first-order Bayesian logic that integrates classical first-order logic with probability theory. Classical first-order logic (FOL) is by far the most commonly used, studied and implemented logical system, serving as the logical basis for most current-generation AI systems and ontology languages. MEBN logic provides the basis for extending the capability of these systems by introducing a logically coherent representation for uncertainty. Because a MEBN theory represents a coherent probability distribution, Bayes Theorem provides a mathematical foundation for learning and inference, that reduces to classical logic in the case of certain knowledge (i.e., all probabilities are zero or one).

MEBN represents the world as comprised of entities that have attributes and are related to other entities. Knowledge about the attributes of entities and their relationships to each other is represented as a collection of MEBN fragments (MFrags) organized into MEBN Theories (MTheories) An MFrag represents a conditional probability distribution for instances of its resident random variables given their parents in the fragment graph and the context nodes. An MTheory is a set of MFrags that collectively satisfies consistency constraints ensuring the existence of a unique joint probability distribution over instances of the random variables represented in each of the MFrags within the set. MEBN semantics integrates the standard model-theoretic semantics of classical first-order logic with random variables as formalized in mathematical statistics.

As a full integration of first-order logic and probability, MEBN provides: (1) a means of expressing a globally consistent joint distribution over models of any consistent, finitely axiomatizable FOL theory; (2) a proof theory capable of identifying inconsistent theories in finitely many steps and converging to correct responses to probabilistic queries; and (3) a built in mechanism for adding sequences of new axioms and refining theories in the light of observations. Thus, even the most specific situations can be represented in MEBN, provided they can represented in FOL. Furthermore, because MEBN is a first order Bayesian logic, using it as the underlying semantics of PR-OWL not only guarantees a formal mathematical foundation for a probabilistic extension to the OWL language (PR-OWL), but also ensures that the advantages of Bayesian Inference (e.g. natural “Occam’s Razor”, support for learning from data, etc.) will accrue to PR-OWL probabilistic ontologies. A comprehensive explanation of MEBN logic is not on the scope of this work, but the interested reader is directed to [5], [9].

2. An Upper Ontology for Probabilistic Models

2.1. The Basics of PR-OWL

PR-OWL was developed as an extension enabling OWL ontologies to represent complex Bayesian probabilistic models in a way that is flexible enough to be used by diverse Bayesian probabilistic tools (e.g. Netica, Hugin, Quiddity*Suite, JavaBayes, etc.) based on different probabilistic technologies (e.g. PRMs, BNs, etc.). More specifically, OWL is an upper ontology for probabilistic systems that can be used as a framework for developing probabilistic ontologies (as defined in Section 1.2) that are expressive enough to represent even the most complex probabilistic models.

Ideally, specification of a probabilistic ontology language would follow the steps defined by the W3C [10] to issue an official standard. New tools would need to be developed to support the extended syntax and implied semantics of the probabilistic extensions. Such an effort would require commitment from diverse developers and workgroups, which falls outside our present scope. For this reason, PR-OWL was written as an upper OWL ontology. DaConta et al. define an upper ontology as a set of integrated ontologies that characterizes a set of basic commonsense knowledge notions ([11], page 230). In this preliminary work on PR-OWL as an upper ontology, these basic commonsense notions are related to representing uncertainty in a principled way using OWL syntax. If PR-OWL were to become a W3C Recommendation, this collection of notions would be formally incorporated into the OWL language as a set of constructs that can be employed to build probabilistic ontologies.

The PR-OWL upper ontology for probabilistic systems consists of a set of classes, subclasses and properties that collectively form a framework for building probabilistic ontologies. The first step toward building a probabilistic ontology in compliance with our Definition 1 is to import into any OWL editor an OWL file containing the PR-OWL classes, subclasses, and properties (one is available at http://www.pr-owl.org/pr-owl.owl). In fact, this is exactly what we did when we built the Star Trek probabilistic ontology depicted in [7]. We used the Protégé Ontology Editor (available at http://protege.stanford.edu) import feature to import the pr-owl.owl file we previously downloaded. Figure 1 shows the major elements within that file.

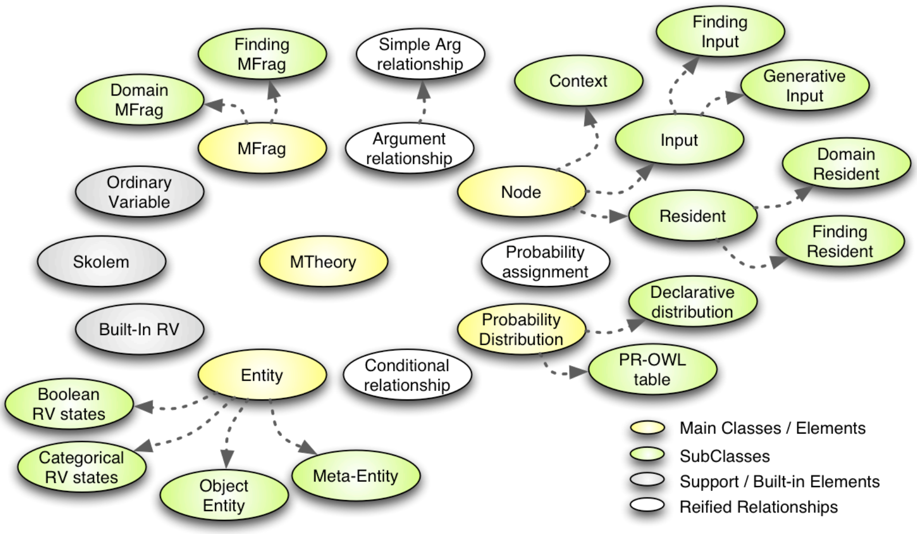

Figure 1 – Main Elements of the PR-OWL Upper Ontology

After importing the PR-OWL definitions, the next step in ontology design is to construct domain-specific concepts, using the PR-OWL definitions to represent uncertainty about their attributes and relationships. As an example, the concepts of the above-mentioned Star Trek probabilistic ontology were either subclasses or instances of the imported PR-OWL upper ontology. Using this procedure, an ontology engineer is not only able to build a coherent generative MTheory and other probabilistic ontology elements, but also make it compatible with other ontologies that use PR-OWL concepts.

Figure 2 shows the initial Protégé screen after importing the PR-OWL ontologies and defining the classes of object entities that will be part of the ontology. In Protégé, concepts of imported ontologies appear with a light colored dot icon and the namespace abbreviation at the left side of the concept's name, as it can be seen in the Asserted Hierarchy window on the left side of the picture.

The darker icons (Starship, Zone, Sensor Report, and TimeStep) correspond to the classes created as a first step in building the Starship probabilistic ontology. PR-OWL object entities correspond to frames in frame systems and to objects in object-oriented systems. The simple model used in this research contains only four object entities; so four classes were created under the PR-OWL ObjectEntity Class (i.e. Starship, Zone, SensorReport, and TimeStep). These are the user-defined classes that convey the equivalent of what a standard ontology would represent about a domain, so its individuals are the concepts and entities that would populate a non-probabilistic description of that domain. In our Starship ontology, the domain instances will be individual zones, sensor reports, starships, and time steps, all represented as individuals of the domain classes created by the user.

The other PR-OWL classes shown in the picture are directly fulfilled by individuals representing the elements of a generative MTheory. The user does not create new classes here, but individuals that convey the information necessary for performing inferences about the attributes of and relationships among instances of the entities represented in the ontology. In other words, these individuals express the probabilistic aspects of the domain MTheory. These individuals can be used by a probabilistic reasoner as templates for building a situation-specific Bayesian network (SSBN) to answer a probabilistic query about specific entities in a given situation (e.g., whether one's own starship is under attack from the starships detected by its sensors).

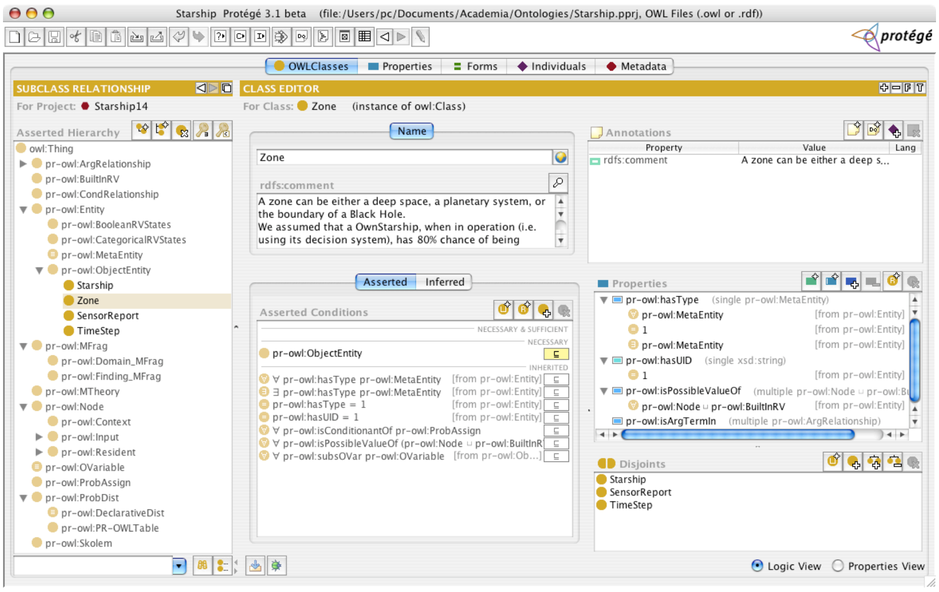

Figure 2 – Main Elements of the PR-OWL Upper Ontology

In our conceptual approach, we considered the question of whether to represent an MFrag template as a class or an instance. It is important to keep in mind that no matter what approach an ontology designer uses in the light of his/her objectives, the structural and logical constraints of MEBN logic will be inherited. Since the other elements of the ontology will also be either instances or subclasses of the imported PR-OWL upper ontology, then all will inherit the structural and logical constraints that collectively enforce the compliance with MEBN rules, thus guaranteeing that such an ontology would be a coherent, logically consistent MEBN Theory.

Although PR-OWL does not enforce a specific resolution of this issue, we considered the pros and cons of modeling our concepts as subclasses or instances of PR-OWL classes in the design of our Star Trek probabilistic ontology. Our experience leads us to conclude that the objectives and characteristics of the probabilistic ontology being built will dictate how to make this choice. Again, representing uncertainty within an ontology is not the same thing as building a probabilistic system. In our Star Trek case study, the generative MTheory was seen as the part of the system that holds the domain knowledge used in this process. In other words, the process of building, working and storing the instantiated MFrags in this case is not part of the Star Trek probabilistic ontology, but a task to be executed by a Bayesian-capable IT system that uses that ontology.

Finally, from our definition it is possible to realize that nothing prevents a probabilistic ontology from being "partially probabilistic". That is, a knowledge engineer can choose the concepts that he/she is interested to be in the "probabilistic part" of the ontology, while writing the other concepts in standard OWL.

In this specific case, the "probabilistic part" refers to the concepts written using PR-OWL definitions and that collectively form an MTheory. There is no need for all the concepts in a probabilistic ontology to be probabilistic, but at least some have to form a valid MTheory. Of course, only the concepts being part of the MTheory will be subject to the advantages of the probabilistic ontology over a deterministic one.

2.2. An Operational Concept for using PR-OWL

At its current stage of development, PR-OWL contains only the basic representation elements that provide a means of representing any MEBN theory. Such a representation could be used by a Bayesian tool (acting as a probabilistic ontology reasoner) to perform inferences to answer queries and/or to learn from newly incoming evidence via Bayesian learning.

However, building MFrags in a probabilistic ontology is a manual, error prone, and tedious process. Avoiding errors or inconsistencies requires deep knowledge of the logic and of the data structures of PR-OWL. Without considering the future paths to be followed by research on PR-OWL (i.e. whether it will be kept as an upper ontology or transformed into a semantic extension to the OWL language), the framework provided by the upper ontology on probabilistic models already enables the development of plugins to current OWL editors for building and using probabilistic ontologies.

Figure 3 illustrates a concept for a possible plugin based on the OWL Protégé editor (which is itself an OWL plugin). It shows a MFrag graphical editor that uses a concept very similar to BN construction GUIs found in graphical packages such as Netica™ (demo available at http://www.norsys.com). To build an MFrag a user has to select the icon related to the kind of node he/she wants to create (e.g. resident, input, context), connect that node with its parents and children, and enter its basic characteristics (i.e. name, probability distribution, etc.) by double-clicking on it or via another GUI-related facility. Nodes would be associated with attributes of entities or relationships among entities, either by clicking on an existing attribute or relationship to name the node, or by naming the node to automatically create the attribute or relationship.

The idea of such a plugin is to hide from users the complex constructs required to convey the many details of a probabilistic ontology, such as the reified relationships, composite RV term constructions (with or without quantifiers and Exemplar constants), and others. In the figure, the Zone MFrag was selected from the combo box in the top of the viewing area, thus information about its nodes is displayed in a graphical format that allows the user to build more nodes, edit or view the existing ones. and then chose node ZoneEShips(z) so it appears highlighted (a red box around it) and all its data is shown in the lower square.

Tedious tasks such as building a PR-OWL table with many cells could be carried out much more quickly and with fewer errors, thus providing a boost in productivity. In the probability table case, the user would only have to fill the probabilities in the correct cells of a CPT's graphical display and the plugin would build their respective PR-OWL constructs.

Another point of usage improvement is the intrinsic syntax check provided by a guided construction. As an example, when writing a composite RV term, the user would not have to actually write the complex reified relations (ArgRelationships, Skolem contants, OVariables, Inner terms, etc). Instead, a menu with the allowed connectives would be available so his/her task would be reduced to enter the arguments of the formula and embed the connectives the way he/she wants. The final result would be a valid formula that would then be transformed in PR-OWL syntax by the plugin.

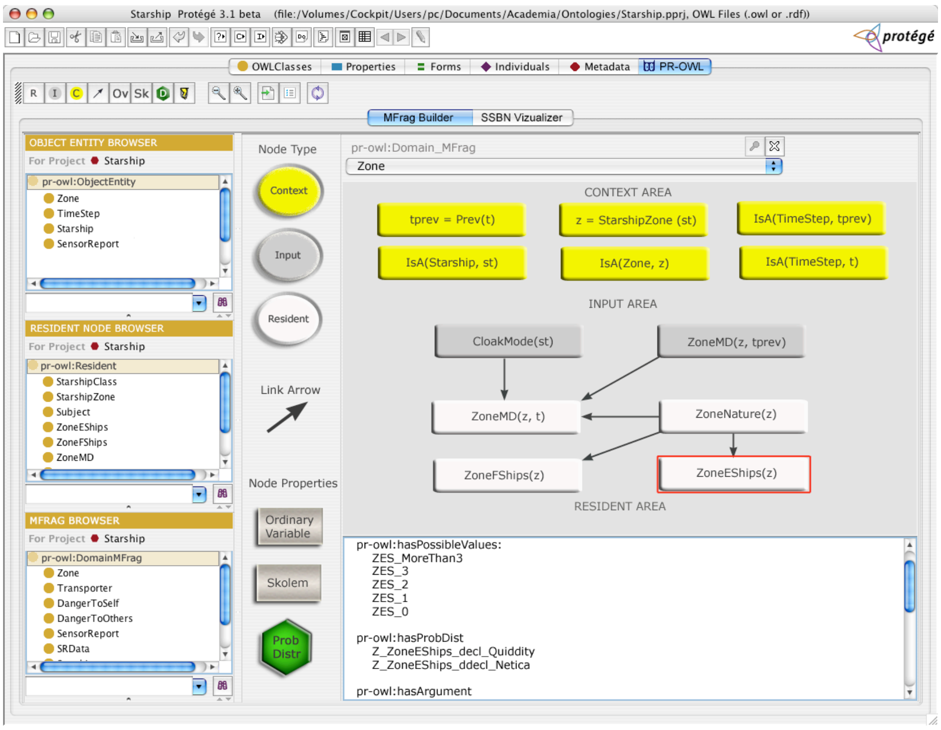

Figure 3 – Snapshot of a Graphical PR-OWL Plugin

It is important to keep in mind that this brief description of an operational concept barely scratches the surface of the many possibilities for the technology presented here, and its purpose is to point out one such possibility. Nevertheless, implementing such a plugin would be an important first step toward making probabilistic ontologies a reality, opening the door to its wide use in many domains.

3. Related Research

One of the main reasons why research in ontology languages is still focused on deterministic approaches has been the limited expressiveness of traditional probabilistic languages. There is a current line of research focused on extending OWL so it can represent probabilistic information contained in a Bayesian Network (e.g. [12]-[13].). The approach involves augmenting OWL semantics to allow probabilistic information to be represented via additional markups. The result would be a probabilistic annotated ontology that could be translated to a Bayesian network (BN). Such a translation would be based on a set of translation rules that would rely on the probabilistic information attached to individual concepts and properties within the annotated ontology. BNs provide an elegant mathematical structure for modeling complex relationships among hypotheses while keeping a relatively simple visualization of these relationships. Yet, the limited attribute-value representation of BNs makes them unsuitable for problems requiring greater expressive power.

Another option for representing uncertainty in OWL is to focus on OWL-DL, a decidable subset of OWL that is based on Description Logics [14]. Description Logics are a family of knowledge representation formalisms that represent the knowledge of an application domain (the "world") by first defining the relevant concepts of the domain (its terminology), and then using these concepts to specify properties of objects and individuals occurring in the domain (the world description). Description Logic divides a knowledge base into two components: a terminological box, or T-Box, and the assertional box, or A-Box. The first introduces the terminology (i.e. the vocabulary) of an application domain, while the latter contains assertions about instances of the concepts defined in the T-Box. Description Logic is a subset of FOL that provides a very good combination of decidability and expressiveness, and is the basis of OWL-DL. Probabilistic extensions have been developed for description logics (e.g., [15][16][17]). Description logics are highly effective and efficient for the classification and subsumption problems they were designed to address. However, their ability to represent and reason about other commonly occurring kinds of knowledge is limited. An example of a restrictive aspect of DL languages is their limited ability to represent constraints on the instances that can participate in a relationship.

Although the above approaches are promising where applicable, a definitive solution for the generic semantic mapping problem requires a general-purpose formalism that gives ontology designers a range of options to balance tractability against expressiveness. Pool and Aiken [18] developed an OWL-based interface for the relational probabilistic toolset Quiddity*Suite, developed by IET, Inc. Their constructs provide an expressive method for representing uncertainty in OWL ontologies. Their work is similar in spirit to ours, but is specialized to the Quiddity*Suite toolset. We employ MEBN as our underlying logical basis, thus providing full first-order expressiveness. Costa [7] presents rules for constructing PR-OWL ontologies in a manner that can be translated into Quiddity*Suite, and for performing the translation.

4. Discussion

Semantic interoperability is a major objective in general IT system development and a necessary ingredient for systems seeking improved knowledge sharing and reuse. In this work, we discussed the role of ontologies in general and probabilistic ontologies in particular as a means to achieve semantic interoperability. We presented a Bayesian ontology language based on MEBN logic that provides the means to express first-order probabilistic theories. We also addressed the inherent complexity of the language and the consequent need of a GUI as a means to make the development of probabilistic ontologies a less intricate task.

Probabilistic ontologies are an increasingly important topic in forums devoted to best practices in systems development. Given the nature of the domain knowledge embedded in their systems, system developers in general would profit most from the advantages of being able to convey such knowledge with a principled treatment for uncertainty. That would allow the proper use of probability information to help devise reliable, more general semantic mapping schemas by using probabilistic ontologies to represent the mappings between two or more ontologies as its instances.

References

[1] Marvin L. Minsky. Framework for Representing Knowledge. In The Psychology of Computer Vision. P. H. Winston (Eds.), pages 211-277. New York, NY: McGraw-Hill, 1975.

[2] Ronald J. Brachman. What's in a Concept: Structural Foundations for Semantic Networks. International Journal of Man-Machine Studies, 9(2), 127-152, 1977.

[3] S. Carey, Martin Kleiner, Michael R. Hieb and R. Brown, Standardizing Battle Management Language – A Vital Move Towards the Army Transformation, Paper 01F-SIW-067, Fall Simulation Interoperability Workshop, 2001.

[4] Field Manual No. FM 1-02 (FM 101-5-1) MCRP 5-12A, *Operational Terms and Graphics*, Headquarters, Department of the Army, Washington, DC, 21 September 2004.

[5] Kathryn B. Laskey. MEBN: A Logic for Open-World Probabilistic Reasoning. The Volnegau School of Information Technology and Engineering. George Mason University, Fairfax, VA, USA. Available at http://ite.gmu.edu/~klaskey/index.html.

[6] Heckerman, D., Meek, C., & Koller, D. (2004). *Probabilistic models for relational data*. Redmond, WA: Microsoft Corporation.

[7] Paulo C. G. da Costa. Bayesian Semantics for the Semantic Web. Doctoral Thesis, School of Information Technology and Engineering, George Mason University. Fairfax, VA, USA, 2005. Available at http://hdl.handle.net/1920/455.

[8] Paulo C. G. da Costa, Kathryn B. Laskey, and Kenneth J. Laskey. PR-OWL: A Bayesian Framework for the Semantic Web. Proceedings of the first workshop on Uncertainty Reasoning for the Semantic Web (URSW 2005), held at the Fourth International Semantic Web Conference (ISWC 2005). November, 6-10 2005, Galway, Ireland. Available at http://hdl.handle.net/1920/454.

[9] Paulo C. G. da Costa, and Kathryn B. Laskey. Multi-Entity Bayesian Networks without Multi-Tears. Draft, Department of Systems Engineering and Operations

Research, George Mason University: Fairfax, VA, USA, 2005. Available at http://hdl.handle.net/1920/456.
[10] Ian Jacobs, Editor. World Wide Web Consortium Process Document. June 18, 2003. Retrieved March 03, 2006, from http://www.w3.org/2003/06/Process-20030618/cover.html.
[11] Michael C. DaConta, Leo J. Obrst, and K. T. Smith. The Sematic Web: A Guide to the Future of Xml, Web Services, and Knowledge Management. Indianapolis, IN, USA: Wiley Publishing, Inc., 2003.
[12] Zhongli Ding, and Yun Peng. A Probabilistic Extension to Ontology Language OWL. in 37th Annual Hawaii International Conference on System Sciences (HICSS'04). Big Island, Hawaii, 2004.
[13] Tao Gu, Hung Keng Pung, and Da Qing Zhang. A Bayesian Approach for Dealing with Uncertainty Contexts. in Second International Conference on Pervasive Computing. Vienna, Austria: Austrian Computer Society, 2004.
[14] F. Baader, et al., Editors. The Description Logic Handbook: Theory, Implementation and Applications. First edition ed., Cambridge University Press: Cambridge, UK, 2003.
[15] Manfred Jaeger. Probabilistic Reasoning in Terminological Logics. Paper presented at the Fourth International Conference on Principles of Knowledge Representation and Reasoning (KR94), May 24-27. Bonn, Germany, 1994.
[16] Daphne Koller, A. Y. Levy, and Avi Pfeffer. P-CLASSIC: A Tractable Probabilistic Description Logic. Paper presented at the Fourteenth National Conference on Artificial Intelligence (AAAI-97), July 27-31. Providence, RI, USA, 1997.
[17] R. Giugno, and Thomas Lukasiewicz. P-SHOQ(D): A Probabilistic Extension of SHOQ(D) for Probabilistic Ontologies in the Semantic Web. in European Conference on Logics in Artificial Intelligence (JELIA 2002). Cosenza, Italy: Springer, 2002.
[18] Michael Pool, and Jeffrey Aikin. KEEPER: and Protégé: An Elicitation Environment for Bayesian Inference Tools, in Workshop on Protégé and Reasoning held at the Seventh International Protégé Conference: Bethesda, MD, USA, 2004.