Probabilistic Ontologies for Efficient Resource Sharing in Semantic Web Services

Paulo Cesar G. da Costa, Kathryn B. Laskey
School of Information Technology and Engineering,
George Mason University
4400 University Drive
Fairfax, VA 22030-4444 USA
{pcosta, klaskey}@gmu.edu

Kenneth J. Laskey[#]
MITRE Corporation, M/S H305
7515 Colshire Drive
McLean, VA 22102-7508 USA
klaskey@mitre.org

Abstract. Service Oriented Architecture (SOA) is a key technology to support interoperability among data and processing resources. Semantic interoperability requires mapping between vocabularies of independently developed resources, a task fraught with uncertainty. Probabilistic ontologies enable representation of knowledge in domains characterized by uncertainty. As such, they promise to improve the quality of service descriptions, enable more thorough analysis of service composition opportunities, and provide a theoretically sound methodology for semantic mapping under uncertainty. This paper defines probabilistic ontologies, discusses their application to SOA, and presents a conceptual scheme for using a federation of ontologies (with both common and probabilistic ontologies) as a semantic mapping tool for service oriented information exchange systems with different levels of service descriptions (including legacy and probabilistic enabled descriptions).

1 Semantic Interoperability in the Semantic Web and Service Oriented Architecture Frameworks

A fundamental tenet of the Semantic Web (SW) vision is that adding semantics to web resources can spark a paradigm shift from information-based data exchange to knowledge-based data-exchange. HTML syntax hard-codes a limited single-purpose set of semantic categories. In contrast, the Semantic Web envisions resources annotated with well-defined, explicitly represented semantics that provides the basis for rich description and reasoning. Explicit semantics is essential for appropriate processing of syntactically identical but semantically different terms (e.g., “Washington” the President, the city, or the baseball team). Ontologies, or shared repositories of precisely defined concepts expressed in standardized languages, are a vital tool for enabling semantic interoperability among web resources. Thus, ontologies are a means for transforming the current “Web of shared data” into a “Web of shared knowledge.”

[#] The author's affiliation with The MITRE Corporation is provided for identification purposes only, and is not intended to convey or imply MITRE's concurrence with, or support for, the positions, opinions or viewpoints expressed by the author.

A traditional ontology can at best list multiple possible senses for a word such as "Washington," with no ability to grade their relative plausibility in a given context. This may be sufficient in tightly controlled environments in which a limited set of allowable usages of each term can be strictly enforced. However, this approach cannot be used in an open world where incomplete information is the rule and little control over ontology publishers is feasible. For this reason, traditional ontologies are inadequate for unambiguous semantic matching of independently defined concepts. To fill that gap, probabilistic ontologies are proposed as a means to extend common ontologies, enabling a principled representation of uncertainty and promoting a means to represent the relative plausibility of a specific meaning of a concept based on the context in which it is used.

In parallel with the Semantic Web efforts, Service Oriented Architecture (SOA) has become the leading approach for accessing and using distributed resources developed by independent entities and working with independently developed vocabularies and associated semantics. The advent of SOA marks a transformation from a mostly data-driven Web, with little interaction between requesters and providers of information, into an environment in which information and other resources are accessed and used in a much more dynamic, interactive, and unpredictable fashion.

The supporting technology for the SOA model is composed of XML-based standards and protocols focused on providing a shared understanding of the available services. Currently, accepted standards for developing solutions based on Web Services (the most prevalent implementation of SOA) include SOAP, a message structure used for exchanging XML serializations of content and message handling instructions in a decentralized, distributed environment [1], and the Web Services Description Language (WSDL), which represents messages exchanged when invoking a Web Service [2]. However, these XML-based structures do not have the ability to explicitly formalize the underlying semantics of a given Web Service description, rendering them insufficient to ensure a common understanding of the described Web Service. As pointed out by Paolucci et al. [3], two identical XML descriptions may have different meanings depending on who uses them and when. Because it is unrealistic to expect that all providers and consumers will have equivalent perspectives and knowledge regarding a given service, a common understanding of a given Web Service can be reached only at the semantic level, where the different perspectives and knowledge can be matched.

Not surprisingly, the need for semantic-aware resource descriptions is widely recognized, and is being addressed by research focused on enabling Web Service providers to describe the properties and capabilities of their Web Services in unambiguous, computer-interpretable form (e.g., OWL-S [4], WSMO [5], SWSL [6], and SAWSDL [7]).

This paper argues that progress on both SW and SOA is hampered by the lack of support for uncertainty in common ontology formalisms. We postulate that probabilistic ontologies can fill a key gap in semantic matching technology, thus facilitating widespread usage of Web Services for efficient resource sharing in open and distributed environments. We begin with a general definition of a probabilistic ontology. Next, we present PR-OWL, a language for defining probabilistic ontologies. Finally, we explore possible use cases for applying probabilistic ontologies in a Service Oriented Architecture.

2 Common vs. Probabilistic Ontologies

Since its adoption in the field of Information Systems, the term ontology has been given many different definitions. A common underlying assumption is that the formal foundation for knowledge representation and reasoning would be classical logic. We argue that first-order probabilistic logic is a more appropriate foundation.

The *de facto* standard for developing ontologies geared to the Semantic Web is OWL, a W3C recommendation [8]. 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 [9] and Description Logics, which evolved from the so-called "Structured Inheritance Networks" [10]. 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. In fact, virtually all current ontology formalisms are based on classical logic, and 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 standalone 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," extracted from a book about the American revolution, or alternatively, from a newspaper story about a recent United Nations debate. As another example, the developers of an ontology for military planning [11] identified over a dozen different doctrinal uses of the word "clear" within the United States Department of Defense [12]. 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 (e.g. [13]). 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[1]. 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 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 the pragmatic stance that although our ultimate objective is to seek the truth about Reality as it is, full knowledge 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.

Although interest in probabilistic ontologies has been growing, there is as yet no commonly accepted formal definition of the term. Over the past several decades, semantically rich and computationally efficient formalisms have emerged for representing and reasoning with probabilistic knowledge (e.g., [14]-[15]). Building upon the advances in this area, we have adopted 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.

Definition 1 (from [16]): A probabilistic ontology (PO) 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;

[1] Phlogisten is a hypothetical element that was thought to be present in all flammable materials and was necessary for burning to occur. Becher's theory of phlogiston was dominant until Lavoisier proved that combustion requires oxygen.

- 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 providing a means of representing 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. Ontologies and database schemas are sometimes confused because they are expressed using similar formalisms. 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 built 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.

As a means to develop probabilistic ontologies that might be used in a framework for semantic matching of Web Services, we are using and developing PR-OWL [16, 17], an extension that enables OWL ontologies to represent Bayesian probabilistic models in a way that is flexible enough to be used by diverse Bayesian probabilistic tools based on different probabilistic technologies. That level of flexibility can only be achieved using the underlying semantics of first-order Bayesian logic [14], which is not a part of the standard OWL semantics and abstract syntax. Therefore, it seems clear that PR-OWL can only be realized via extending the semantics and abstract syntax of OWL. However, in order to make use of those extensions, it is necessary to develop new tools supporting the extended syntax and implied semantics of each extension. Such an effort would require commitment from diverse developers and workgroups, which falls outside our present scope.

The major advantages of using PR-OWL are its flexibility and representational power, both inherited from the fact that the language is based on MEBN, a full integration of first-order logic and probability that merges the expressiveness of the former with the inferential power of the latter. 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, any knowledge can be represented in MEBN, provided it can be 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 outside the scope of this work, but the interested reader is directed to [14], [18].

3 The Role of Probabilistic Ontologies in SOA

In order to envision the applicability of POs in SOAs, it is necessary to first understand what kind of uncertainties might be present in a service-oriented environment. SOA, as defined in its reference model [19], is a paradigm for bringing together needs and capabilities to address those needs. It requires establishing an *execution context* (EC), which is an alignment of all technical and policy-related aspects, including vocabularies, protocols, licensing, quality of service (QoS), etc. Much of this specific information is contained in or linked to the service description and/or the description of underlying capabilities. Considering the complexity involved, many forms of uncertainty can be present within a given execution context. For example, uncertainty may arise in the description content (e.g. information annotated with its source but there is no way to verify whether the identity of the source is correct), in the way information is captured as part of a description (e.g. information annotated with its respective source but with no indication of whether it is raw or processed data), or in the applicability of information to current need (e.g., information on recording equipment that does not indicate whether the recorded data fall within a reasonable range for the recording conditions). An ontology that represents statistical information can enable a reasoner to draw inferences about the missing information. For example, consider a report that a given device has recorded an ambient temperature of 28 degrees Celsius was reported at Washington Dulles Airport (IAD) on 23 January. This is a highly unlikely, but not impossible, temperature reading for January in the DC Metropolitan area. Statistical information about climate, sensor reliability, and data recording error rates, if represented in the relevant domain ontologies, could be used to draw inferences about the about the likely temperature at IAD on 23 January that appropriately account for the possibility of various kinds of error. This example illustrates the need for a principled representation of uncertainty in service descriptions, a feature not found in current specifications.

A typical Web Services scenario can be seen as publish-find-bind triangle, in which a service provider publishes a service description, a consumer searches a service registry for a service satisfying his criteria, analyzes the included information (or link to information) on the message structure to be exchanged and the address to exchange it, and interacts with the service to retrieve the resources needed. In this triangle, there are implicit, unspoken challenges for which a principled representation of uncertainty is needed. For example:

- The publisher has to choose vocabulary with which to describe the service (or some other resource related to the service). The vocabulary sets the properties for that class of item. Service ontology developers attempt to define the "right" set and structure of properties for the anticipated users. The consumer must know and understand the semantics of the chosen property vocabulary because these are the properties used to describe the class and its instances, and the consumer must understand and use the same vocabulary or there must be a known and accessible mapping between the properties used for description and those used as search categories. There are many opportunities for uncertainty about intended meanings.
- The publisher uses the chosen property vocabulary as the basis to describe and register instances of that class. This means that the publisher associates values with the properties and registers the instance. But what is the vocabulary for the values? All parties may agree that something has the property *color* and on the meaning of that property, but if the publisher uses only primary colors and the subscriber's search criterion asks for the color *pink,* the latter will never find a match for items the first had catalogued. How does a client's requested value relate to a provider's published values? Do they agree on the vocabulary? Do they agree on the mechanism to mediate vocabulary mismatches?
- The publisher chooses a property vocabulary and creates instance descriptions by associating values. One can infer what properties the publisher considers important by which properties s/he chooses to populate, assuming values are not necessarily assigned for all possible properties. But what of the consumer's priorities when assigning search criteria? If the consumer assigns relative importance, how does the search engine trade off among different combinations of matches across the consumer's search criteria, and how are missing attribute values handled?

Beyond publish-find-bind for a single service, the vision is to provide services at the appropriate granularity, combining atomic services into more complex tasks. For example, suppose a supplier needs to find the dimensions and weight limits for cargo containers for future shipments of items it produces. In today's integration paradigm, the supplier would need to query specific shipping agents directly, and might need to develop special-purpose software interfaces to support interactions with individual shipping agents. In the envisioned architecture, the supplier would invoke a service that (*i*) searches a UDDI registry for shipping agents; (*ii*) queries each for its respective restrictions; (*iii*) compares with the supplier's requirements; and (*iv*) selects a shipper that meets the requirements.

This simple scenario does not include other actions that must be included in such a transaction. For example, security will be needed to authenticate the supplier to the shipping agent and the shipping agent to the supplier. Other actions may be required to establish that each party is authorized to engage in business with the other. The interaction itself may require a guaranteed level of service that would fall into the realm of reliable messaging to guarantee delivery. Additionally, the response from the shipping agent could optionally include video showing details of container packing and handling, and these would not be appropriate to send if the supplier is using a low bandwidth communications link.

Security, reliable messaging, and results dissemination are examples of general-purpose services that could be combined with services for specific business functions, thus alleviating the business service from the need to create and maintain all supporting services. All of these services will have associated service descriptions so that someone composing a robust service combination can identify the appropriate services and the process by which these will work together to provide the higher-level functionality. That said, what are the uncertainties in identifying the correct services and combining these to form a consistent package? Is uncertainty even a relevant concept, or is it a black-and-white issue of whether the pieces fit or not? When trying to decide among several services that appear to satisfy aspects of the same needed function, does the ability to reason under uncertainty come into play in identifying the component services to use and how to combine these?

The above questions do not have simple, universally valid answers. Nevertheless, we expect that there will be problems for which deterministic implementations of SOA elements will suffice to build viable solutions, but it is clear that there are issues that cannot be satisfactorily solved without a principled representation of uncertainty. Probabilistic ontology languages such as PR-OWL can fulfill this requirement.

Providing a detailed account of how to use PO languages to build standards for SOA elements, or even examples of (say) service descriptions with probabilistic elements would require detailed explanation that goes beyond the limits of this paper. Thus, as a means to explore another possible use of POs in a SOA environment, we now present a possible framework using a federation of ontologies (common and probabilistic) for tackling the problem of semantic mapping among concepts used in Web Services (WS) descriptions within a WS repository.

Figure 1 shows a simplified scheme for SOA using probabilistic semantic mapping. As a means to illustrate this scheme, we will devise fictitious examples involving Web Service providers within the geospatial intelligence domain. In this scheme, a service consumer or provider that conveys semantic information (ontology that it abides to, metadata about its requests, parameters, etc.) is called a SOA node Level 1, whereas a SOA node that has no semantic awareness is called a SOA node Level 0.

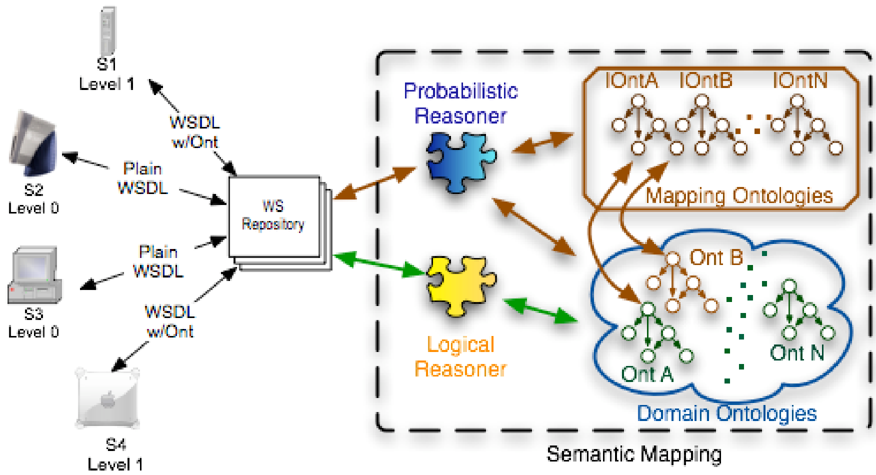

Figure 1 – Probabilistic Semantic Mapping for Web Services

In our first use case, S1 needs to plan a convoy route and requests a service for assessing the possibility of finding guerrilla insurgents in a given region. Being a Level 1 client, S1 sends its request with embedded data about the ontology it references and other semantic information regarding its request (e.g. coordinate system used, expected QoS, etc.). The WS repository, which itself uses an ontology, finds S4, another Level 1 client using the same ontology as S1. This ontology is the PR-OWL ontology "OntB", which represents a probabilistic model of the geospatial domain and has the ability to perform a probabilistic assessment of the requested information. In this case, the request was probabilistic, but the uncertainty involved was related to the service itself (a probabilistic query on a uncertainty-laden domain), and not to the service exchanging process. In other words, the exchange was completed using the logical reasoner alone, since there was a perfect matching in terms of ontologies (both S1 and S4 abide to the same PR-OWL ontology) and the parameters of the requested service, and thus no probabilistic mapping was necessary. (yet, note that S1's query made use of OntB's ability to represent uncertainty about the geospatial intelligence domain.)

In a variation of the previous case, let's suppose that no perfect match between the request and the available providers is found. In this case, the probabilistic reasoner accesses the WS repository to search for the most suitable service given the parameters of S1's request. During that process, it analyses the mapping ontologies related to "OntB" (the ontology referenced by S1) and the domain ontologies related to the services it deemed promising to fit S1's request. In the end, an ordered list of possible providers is built, and the best possible answers will be returned to S1. This simple example shows that there might be many combinations of the use of logical and probabilistic reasoners and ontologies to match the needs of a specific request.

4 Conclusion

Our main objective was to discuss the validity of probabilistic ontologies as a principled representation of uncertainty in a given domain, and its uses in extending the reach of Service Oriented Architecture. Although the concept of a semantic-enabled SOA is in its infancy, we believe much can be achieved by employing both complete and incomplete knowledge to optimize the way resources are exchanged.

References

[1] Mitra, N. "SOAP Version 1.2 Part 0: Primer", W3C Recommendation 24 June 2003, available at http://www.w3.org/TR/soap12-part0/

[2] Chinnici, R.; Moreau, J.; Ryman, A.; and Weerawarana, S. "Web Services Description Language (WSDL) Version 2.0 Part 1: Core Language", W3C Candidate Recommendation 27 March 2006. Available at http://www.w3.org/TR/wsdl20

[3] M. Paolucci, T. Kawamura, T. Payne, and K. Sycara. (2002) "Importing the Semantic Web in UDDI". In C. Bussler, R. Hull, S. McIlraith, M. Orlowska, B. Pernici, and J. Yang, editors, Proceedings of the Workshop on Web Services, E-Business and Semantic Web, CAiSE. Toronto, Canada.

[4] Martin, D. (ed.) "OWL-S: Semantic Markup for Web Services". Available at http://www.daml.org/services/owl-s/1.1/overview/

[5] Domingue, J.; Lausen, H., and Fensel, d. (Chairs) ESSI Web Services Modeling Ontology Working Group. Website, available at http://www.wsmo.org.

[6] Battle, S. *et al.* (2005) "Semantic Web Services Framework (SWSF) Overview". Available at http://www.daml.org/services/swsf/1.0/overview/

[7] W3C (2006) Semantic Annotations for WSDL Working Group. Website, available at http://www.w3.org/2002/ws/sawsdl.

[8] McGuinness, D; and van Harmelen, F. (eds.) "OWL Web Ontology Language Overview", W3C recommendation, 10 February 2004. Available at http://www.w3.org/TR/owl-features/

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

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

[11] Carey, S.; Kleiner, M., Hieb, M. R.; and Brown R. (2001) Standardizing Battle Management Language – A Vital Move Towards the Army Transformation, Paper 01F-SIW-067, in: Proceedings of the IEEE 2001 Fall Simulation Interoperability Workshop.

[12] 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.

[13] Bittner, T.; and Smith, B. (2001) Vagueness and Granular Partitions. In Proceedings of the Second International Conference on Formal Ontology in Information Systems (FOIS 2001). Alpha, NJ: Sheridan Printing Company, Inc.

[14] Laskey, Kathryn B. 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.

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

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

[17] Costa, P. C. G.; Laskey, K. B.; and Laskey, K. J. (2005) 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.

[18] Costa, P. C. G.; Laskey, K. B.; and Laskey (2005) 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.

[19] Mackensie, C. M.; Laskey, K. J.; McCabe, F.; Brown, P. F.; and Metz, R. (2006) Reference Model for Service Oriented Architecture 1.0. Committee Specification 1, 19 July 2006. Available at http://www.oasis-open.org/committees/download.php/19361/soa-rm-cs.pdf