12TH INTERNATIONAL COMMAND AND CONTROL RESEARCH AND TECHNOLOGY SYMPOSIUM

ADAPTING C2 FOR THE 21ST CENTURY

PROBABILISTIC ONTOLOGIES: THE NEXT STEP FOR NET-CENTRIC OPERATIONS

Suggested Tracks:
Track 8 – C2 Technologies and Systems

Dr. Paulo Cesar G. da Costa*
Center of Excellence in C4I
The Volgenau School of Information
Technology and Engineering
George Mason University
4400 University Drive
Fairfax, VA 22030-4444
(703) 993-1644 / 879-6687
pcosta@gmu.edu

Dr. Kathryn Blackmond Laskey
Center of Excellence in C4I
The Volgenau School of Information
Technology and Engineering
George Mason University
4400 University Drive
Fairfax, VA 22030-4444
(703) 993-1644
klaskey@gmu.edu

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

Dr. Edward J. Wright
Information Extraction and Transport, Inc.
1911 N. Ft. Myer Dr., Suite 600
Arlington, VA 22209
ewright@iet.com

*Point of Contact: Dr. Paulo C G da Costa
pcosta@gmu.edu and/or (703) 879-6687

[+] 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.

Abstract

In order to meet the demands envisioned for the battlefield of the 21st century, the DoD is pressing for rapid adoption of the Global Information Grid (GIG), a centerpiece of its transformation towards network-centric operations. Among the enabling technologies being leveraged by GIG-related efforts is the widespread adoption of Service Oriented Architecture (SOA), a powerful approach for effectively connecting consumers and providers of information and data processing resources. However, implementing SOA in the GIG context is a major challenge that requires semantic interoperability among service descriptions. To achieve semantic interoperability, it is necessary to establish mappings between vocabularies of independently developed resources from both providers and consumers. Many research efforts have relied on ontologies as a possible solution to this problem, but with limited success to date. We argue that in such an environment, a principled means for representing uncertainty is needed; something not found in common ontologies. This paper proposes the combined use of probabilistic ontologies and SOA for a Net-Centric framework, and presents a conceptual scheme for battlefield information exchange systems with different levels of service descriptions (including legacy and probabilistic enabled descriptions).

1. Introduction

The DoD's Global Information Grid (GIG) is envisioned as a globally interconnected set of information processing capabilities that will form the technical underpinnings for the doctrine of Network Centric Operations. The GIG architecture (figure 1) marks a radical departure from the old C4ISR model to a service oriented exchange framework in which interconnected units operate cohesively regardless of physical distance. A key aspect of its implementation is the ability to provide reliable and agile information exchange among its components, a huge challenge for which an enabling technology is Service Oriented Architecture (SOA) [1].

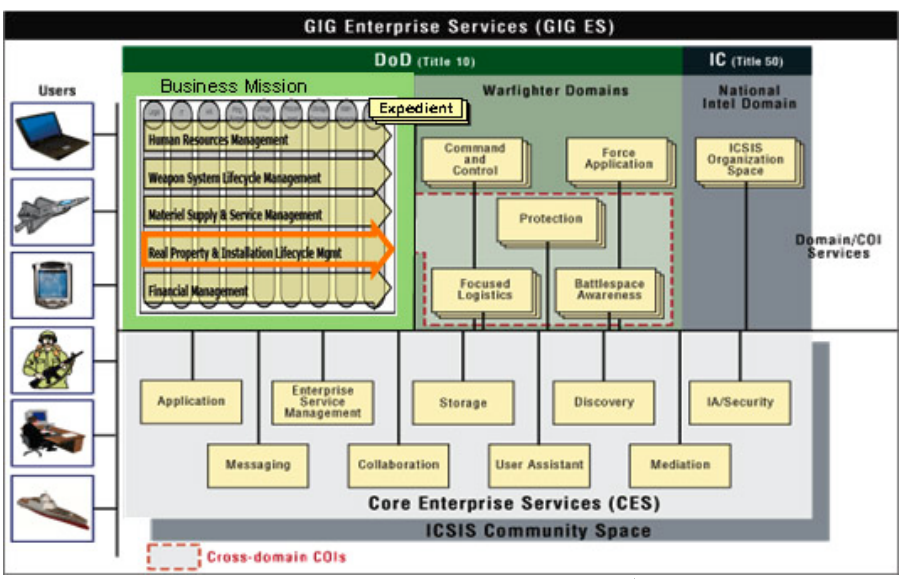

Figure 1 – The GIG architecture[1]

[1] Taken from http://www.acq.osd.mil/ie/bei/gig.htm

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 applications running in an isolated environment, with little interaction between requesters and providers of information, into one in which information and other resources are accessed and used in a much more dynamic, interactive, and unpredictable fashion.

Numerous XML-based standards and protocols are being developed to define Web Services message exchanges that form the basis of much SOA activity. These standards include SOAP, a message envelope structure used for exchanging XML serializations of content and message handling instructions in a decentralized, distributed environment [2], and the Web Services Description Language (WSDL), which represents messages exchanged and the concrete bindings of these messages when invoking a Web Service [3]. However, these XML-based structures do not have the ability to formalize explicitly the underlying semantics of a given Web Service description. This renders them insufficient to ensure a common understanding of the described Web Service. As pointed out by Paolucci et al. [4], 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 the focus of numerous efforts to enable Web Service providers to describe the properties and capabilities of their Web Services in unambiguous, computer-interpretable form (e.g., OWL-S [5], WSMO [6], SWSL [7], SAWSDL [8], and WSDL-S [9]). However, while these efforts provide either semantic models or a means to relate existing semantic models, these do not allow the description authors to specify the degree to which they consider such models complete or applicable. Indeed, SAWSDL "allows multiple annotations to be associated with a given WSDL or XML Schema ... [but] no logical relationship between them is defined by this specification."

We argue that progress on SOA is hampered by the lack of support for uncertainty in common ontology formalisms, and postulate that probabilistic ontologies can fill a key gap in semantic matching technology, thus facilitating more widespread usage of Web Services for efficient resource sharing in open and distributed environments such as the GIG. In the next section, we cover the relevant background information on the need for more powerful data exchange methods, where we present the concept of formal ontology, its extension to capture incomplete knowledge in the form of probabilistic ontology, and our framework for building the latter. Next, we explore the issues related to uncertain, incomplete information in service oriented architectures and present a simple example of how our framework could be applied to such a scenario.

2. The Quest for Semantic Interoperability

2.1. Why Semantics?

Computer are seeing increasingly wide use on the battlefield for storing, exchanging, and working with information. As the availability of information resources increases due to advances in sensor and communications technologies, combatants are starting to face a significant bottleneck in their ability to make use of it. There is an abundance of data but data per se is of limited use to most of our daily tasks until we can transform it into knowledge about the enemy and ourselves. When decision-makers reach their cognitive limit for making sense of incoming data and achieving situational awareness, the result is information overload and suboptimal battlefield performance.

The rapid expansion of connectivity in the field is increasing the problem of information overload. In the race between the availability of data and the decision-makers' ability to transform it into knowledge, many methods for using our ever-growing computational power have been devised to make life easier for the warriors. Yet, in spite of those efforts, there remains a heavy reliance on human cognitive processing for breaking the information to knowledge barrier. This leads us to the question: What is missing for IT techniques to move beyond the information paradigm and begin to work under the knowledge paradigm?

We argue that the answer lies in devising ways for the computers not only to "crunch the bytes" but also to "understand" what those bytes mean. Obviously, computers do not really understand the meaning conveyed by the bytes they "crunch", but this is a widely used metaphor to express the idea that making semantic information explicit and computationally accessible (i.e. better organizing the structure of data) is a powerful, more elegant way of utilizing that data. In other words, if we want to extract knowledge from data, we must develop technologies that allow computers to make use of semantic, contextual information attached to the data being processed. The descriptive information is commonly collected as metadata, and may comprise explicit descriptive elements or links to externally defined information or models.

However, simply adding metadata arbitrarily to military C2 systems would only bring the "Babel Tower" problem to our IT resources. Indeed, when heterogeneous systems need to interoperate in an open world[2], vocabularies that are adequate for a single stand-alone application will break down. This happens because systems developed in isolation from each other will employ different vocabularies originally tailored to different tasks. Inevitably, there is incomplete and partial overlap of terminology and concepts. Even when concepts are clearly defined, inputs available in an open-world system may be insufficient to determine which meaning is most appropriate. Ontologies to describe the individual domains are seen as a means for tackling the problem of semantic inconsistencies among distinct systems.

[2] Due to its sheer size, the GIG can be considered as an open world environment.

2.2. Ontologies to the Rescue

The term Ontology was borrowed from philosophy. Its roots can be traced back to Aristotle's metaphysical studies of the nature of being and knowing[3]. Nonetheless, use of the term ontology in the information systems domain is relatively new, with the first appearance occurring in 1967 ([10], page 22).

One can find many definitions for the concept of ontology applied to information systems, each emphasizing a specific aspect its author judged as most important. With so many possibilities for defining an ontology, one way of avoiding ambiguity is to focus on the objectives underlying its use. For the purposes of this paper, the most important aspect of ontologies is their role as a structured form of knowledge representation. Thus, we use a pragmatic definition of ontologies that emphasizes typical purposes for building and using ontologies.

Definition 1 (from [11]): An ontology is an explicit, formal knowledge representation 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;

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. ■

Ontologies contain a common set of terms for describing and representing a domain in a way that allows automated tools to use the stored data in a more context-aware fashion, intelligent software agents to perform better knowledge management, and many other benefits achieved by a standardized, intensive use of metadata.

Given the need for interoperability among systems based on different schemas and/or ontologies, the ability to exchange data as seamlessly as possible is one of the most desired features of a knowledge representation. Integrating systems created and managed by separate organizations, evolving in different scenarios, and geared to different needs and perspectives is a task that poses many challenges, even when dealing with apparently very similar structures.

Unfortunately, even in tightly controlled settings (e.g. small, closed environments with controlled vocabularies), semantic inconsistencies (such as different concepts with the same name, or different names for the same concept) occur frequently. Current approaches to solve this semantic mapping problem, such as enforcing compliance with standards defined by regulatory authorities (e.g. DOD directives such as 8320.1[4]) or employing generic matching schemes, have consistently fallen short of what is needed to achieve semantic interoperability among systems.

Even though some ontology languages do offer constructs that help to merge ontologies, *they lack a principled means for grading the similarity between concepts or to*

---

[3] The term metaphysics means beyond the study of physics

[4] Available at http://www.defenselink.mil/nii/bpr/bprcd/0039.htm, as of July 6, 2005.

make plausible inferences about the mapping between them. Providing such a means is an important step towards making the semantic mapping problem a less expensive, tedious, error-prone process. In short, the lack of a principled representation for uncertainty in the field of ontological engineering is a major weakness hindering the efforts towards better solutions for the semantic mapping problem. More generally, lack of support for uncertainty management is a serious impediment for more efficient data exchange and, therefore, to make truly interoperable systems within the GIG framework. Clearly, achieving this goal will require more precise semantics and flexible ways to convey information.

Regrettably, for historical reasons and due to the lack of expressivity of probabilistic representations in the past, current ontology languages have no built-in support for representing or reasoning with uncertain, incomplete information. In the uncertainty-laden environment faced by GIG C2 systems, this is a major shortcoming preventing realization of a truly interoperable environment.

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.

2.3. Probabilistic Ontologies

Since the adoption of ontologies in the field of Information Systems, 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. When faced with the complex challenge of representing uncertainty in an ontology, it is a natural tendency to write 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. To understand why this is the case, consider the example of aggregating geospatial information from several databases. Suppose we consult three different databases, all three of which label a particular area as forested. Each of these databases is subject to errors of various kinds. There have been proposals to annotate ontologies with credibility attributes to represent information about uncertainty in different sources of information. If the three reports are independent, annotating each report with a credibility would suffice to aggregate them into a single assessment of the region type. If in our example the three reports agree, standard statistical aggregation technologies would label the region as forested and assign a higher credibility than the three individual credibilities. However, suppose that all three databases obtained their raw data for this area from the same satellite image, and all three applied similar algorithms for assigning a ground cover type label. In this situation, the credibility of the aggregate report is no greater than any of the individual input credibility values. In this case, we need to represent not just a single credibility number, but dependency information about how the credibility depends on the sensor and the data processing algorithm. The approach of attaching a single credibility attribute to each result is insufficient, as too much information is lost to the lack of a good representational scheme that captures structural constraints and dependencies among probabilities. To handle this kind of problem, we need a more expressive formalism for probabilistic ontologies, that is capable of representing subtleties such as correlations due to common sources of error.

Over the past several decades, semantically rich and computationally efficient formalisms have emerged for representing and reasoning with probabilistic knowledge (e.g., [12, 13]). A true probabilistic ontology must be capable of properly representing the nuances these more expressive languages were designed to handle. More formally:

Definition 2 (from [11]): 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 providing a coherent representation of statistical regularities and 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 difference between an ontology and a database schema resides not in the representation language or software application in which they are encoded, but in their intended purposes. Ontologies represent domains in a way intended to 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. Conversely, when a database schema for a given domain is constructed, 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 in a manner that facilitates storage and retrieval of the information needed by the database stakeholders (and their customers), 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 be a primary focus when building a probabilistic ontology for the domain of radar data fusion. Ontology engineers would attempt to express their 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.

2.4. MEBN and 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 is always a trade-off between flexibility and expressiveness among the candidate probabilistic logics. After some careful research (see [11] and [14] for details) we opted for extending the Web Ontology Language OWL to convey probabilistic ontologies, and found that MEBN logic [12] provides a particularly attractive trade-off that made our work easier. This effort resulted in the development of PR-OWL [15].

PR-OWL, shown in figure 2 at its current stage of development, is an upper ontology for probabilistic systems that can be used as a framework for developing probabilistic ontologies. PR-OWL is expressive enough to represent even the most complex probabilistic models and 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. probabilistic relational models, Bayesian networks, etc.).

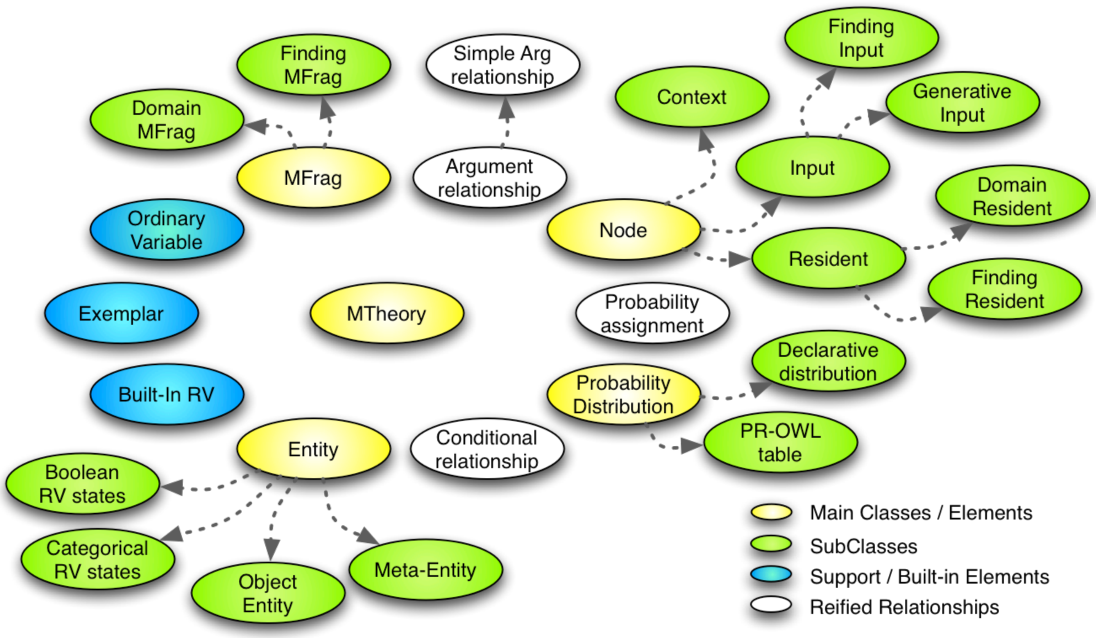

Figure 2 – The PR-OWL Ontology

This level of flexibility can only be achieved using the underlying semantics of first-order Bayesian logic provided by MEBN, 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, and 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, 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 [12] and[16].

3. Applying Probabilistic Ontologies to a SOA-based Application

3.1. The Impact of Uncertainty

In order to envision the applicability of probabilistic ontologies (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 the reference model [17], 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),
- 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
- 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 device has recorded an ambient temperature of 5 degrees Celsius at Rio de Janeiro's Tom Jobin International Airport (GIG) on 23 January. This is a highly unlikely, but not impossible, temperature reading for January near Rio. 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 likely temperature at GIG 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 for SOA services that combine to enable access to such information.

A typical Web Services scenario can be described in terms of a publish-find-bind triangle. A service provider *publishes* a service description. A consumer searches a service registry to *find* a service satisfying his criteria, and analyzes the included information (or link to information) describing the message structure for interacting with the service. Finally, the consumer *binds* to the service using a data exchange protocol and realizes the real world effects of the service. In this triangle, there are implicit, unspoken challenges for which a principled representation of uncertainty is needed. For example:

- The publisher has to choose a 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 criteria. 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 service 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 for the interchange of messages that culminates in the business service of cargo delivery; this may fall into the realm of reliable messaging. 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. Although we expect that there will be problems for which deterministic implementations of SOA elements will suffice to build viable solutions, 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 the 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 registry.

3.2. An Operational Use Case

This section presents a use case to illustrate the benefits of applying probabilistic ontology technology in a military operational scenario. The use case is based on the provision of geospatial data (maps and digital environmental data) and services that include predictions of environmental effects important to military operations. The environmental effects predictions are an important early step in Intelligence Preparation of the Battlefield (IPB) that influences all military planning and operations.

Providing geospatial data and services to warfighters is a complex challenge. The National Geospatial – Intelligence Agency (NGA), Combatant Commands, and the Services, all cooperate to provide data production, data management, data distribution, and geospatial services to support warfighters at every level of command. Dedicated units equipped with state-of-the-art equipment provide support down to the Brigade level. Geospatial services are available in Command and Control software at every level. In addition, the availability of GPS receivers, small format computers and digital cameras, make every soldier – in addition to the specialized units, a potential source for new and updated environmental data.

Managing the data has become a significant challenge in providing geospatial data support. Initial environmental datasets may be of low resolution, old, and incomplete. Once a crisis occurs, NGA and other national assets, civilian organizations, commands and units at every level, and even individual soldiers will all be generating data – for different purposes, using different techniques, in different formats. And since they are all connected, this data is passed around and made available to everyone. Note that the connectivity makes sharing data easier, but makes managing it much harder.

Overcoming the data management challenge requires solving issues with data formats (physical and logical), data representation (raster vs. vector, objects or features vs themes, etc.) and coordinate systems (geodetic datums and map projections). Each of these present potentially complex issues, but are not addressed here. The issue we address has to do with the different vocabularies – or semantics, used by the different data producers. The issue is illustrated with a military use case.

An important environmental effect used to support analysis of enemy and friendly courses of action, is the Cross Country Mobility (CCM) Tactical Decision Aid (TDA). The CCM product predicts the speed that a specific military vehicle or unit can move across country (off roads) based on the terrain. The terrain factors that influence CCM speed are slope, soil type, soil wetness, vegetation and vegetation attributes, ground or surface roughness, and presence of obstacles. For this example, we focus on just soil type. Military CCM algorithms typically use the Unified Soil Classification System (USCS) which defines 16 soils types, based on engineering characteristics that are applicable anywhere in the world [18]. Unfortunately detailed soils maps are not available world wide. In a potential operational area, military spatial data analysts may need to use available data which may come from any number of civilian or government sources (many governments have developed their own unique soil classification systems, tailored to the conditions in their country). Even in a simple case, available soils data may have soil types in the US Department of Agriculture Soil Textual Classification, which has 12 soils classes based on agricultural factors. The challenge then is to translate the vocabularies – or map the ontologies between these two systems.

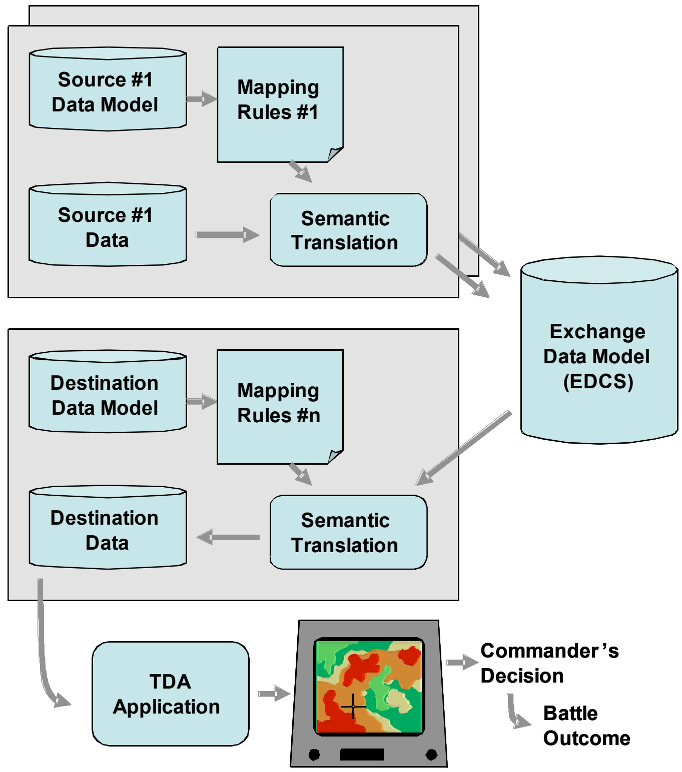

Figure 3 - Process flow for using EDCS to support semantic mapping between multiple source data models and a third destination model.

Within the GIS community, the Open GIS Consortium (OGC) has identified the issue of semantic mapping as a major obstacle to interoperability of GIS data [19]. Commercial software packages have been developed that can be used to perform semantic translation [20]. The military, led by the Modeling and Simulation community, has developed an Environmental Data Coding Specification (EDCS) to be a common environmental ontology to describe environmental "things" important to military operations [21]. Soil types (using the USCS) are part of EDCS as possible values of the soil type attribute for a polygon.

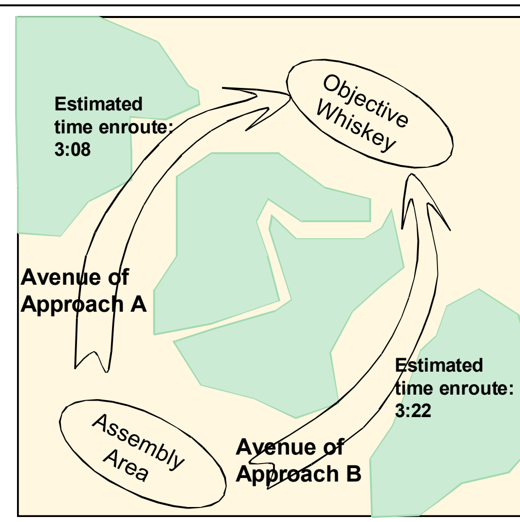

Figure 4 - Traditional CCM analysis, with no accounting for uncertainty.

Figure 3, shows the concept of how EDCS is used to perform semantic mapping in spatial data. The idea is that for each source of data, mapping rules are developed to perform the translation of concepts from the source data model (ontology) into the EDCS data model. If the application requires data in a different destination ontology, another set of mapping rules must be developed for that transformation. In our CCM application, the process proceeds as follows. If soils data for part of the area is available in USCS classes, then no semantic mapping is required to get to EDCS. For the other area, if soil data is available in the USDA system, a set of mapping rules must be developed and applied to yield data in the EDCS system. Because the CCM algorithm uses the EDCS (USCS) soil types, no further semantic mapping is necessary.

The resulting soils data could be used to generate a CCM TDA product, to support analysis of avenues of approach (AA), as shown in Figure 4. All else being equal (ignoring other tactical considerations), since AA Alpha is faster, it is likely to be chosen. So far this looks like an interoperability success story. However if we examine the references to develop mapping rules for translating USDA soil types to USCS soil types there is a problem. As just one example, USDA soil type "Clay" (C), may be classified into USCS as "Inorganic Clays" (CH), "Inorganic silts" (MH), or "Organic Clays" (OH) [18]. These USCS soil classes have very different trafficability characteristics when the soils get wet. Because existing systems provide no way to handle this ambiguity, any mapping rule will be arbitrary and sometimes wrong.

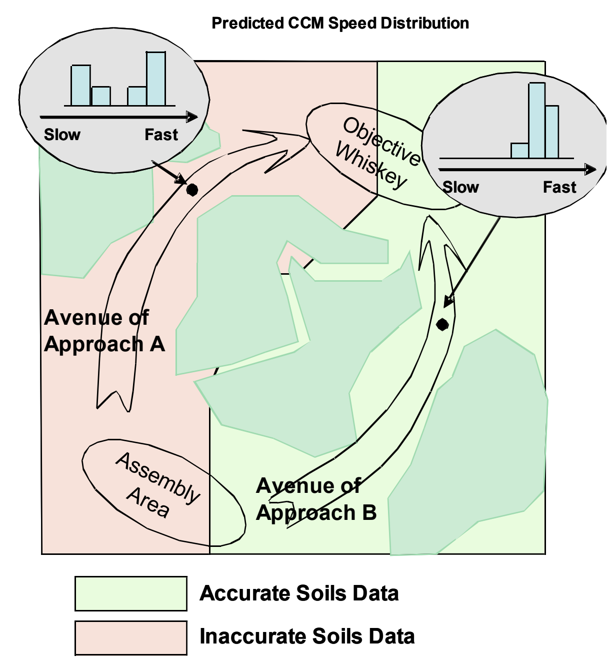

Figure 5 - Understanding the uncertainty and the risks empowers the commander to make better decisions.

Figure 5 shows the same analysis using a hypothetical capability that exploits probabilistic ontologies and supports uncertainty. In this example, the uncertainty is explicitly represented in a probabilistic semantic mapping from USDA soil classes to the USCS classes (see [22] for details). As a result, the visualization can provide a warning that the data in that area is less accurate, and if the user "drills down" to query the underlying probabilistic calculations, the system can show that even though AA Alpha is probably faster than AA Beta, there is a significant risk that AA Alpha will actually be much slower. Based on this awareness, the commander may select the other AA, or may wish to gather more information – by tasking sensors or conducting a recon, to verify the trafficability conditions along AA Alpha. By explicitly representing uncertainty, the commander can understand the risks, and make better battlefield decisions.

The limitation of existing facilities (commercial and military) for performing semantic mappings is that they do not account for uncertainty in the mappings. Use of probabilistic ontologies would overcome this limitation. Probabilistic ontologies offer an additional opportunity. If the probabilistic ontology for soil types is extended to include (probabilistic) relationships between geographic themes that are related to soil type (e.g., vegetation, slope, aspect), then use of this additional information can improve the accuracy of the mapping from USDA soil types to USCS soil types.

3.3. Representing Uncertainty in Services

Implementing this example requires more than just probabilistic ontologies. Also needed are facilities to track the quality of the source data, and to perform automated operations (data integration, and propagation of uncertainty) on a computer representation of the data quality. As we implied above, those are all features that can and should be conveyed via a service registry capable of handling both the semantics of data and its embedded uncertainty.

Figure 6 shows a simplified scheme for SOA using probabilistic semantic mapping in which a commander needs to choose between avenue of approach Alpha and its alternative Beta from figures 4 and 5,. The commander makes the choice with the aid of a computer-based support system, S1, which evaluates the available terrain data and makes recommendations of the best avenues of approach.

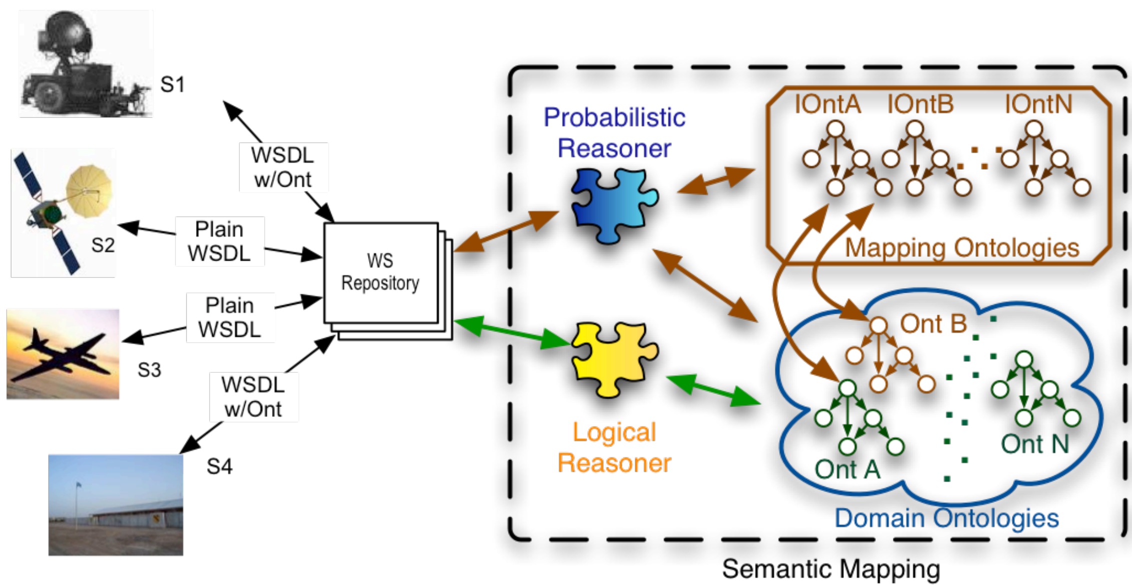

Figure 6 – Semantic Mapping in Web Services

To perform its calculations, S1 needs the best data available at the time. Thus, S1 sends a request with embedded data about the ontology it references and other semantic information regarding its data needs (e.g. coordinate system used, expected QoS, etc.). The WS repository, which itself uses an ontology, finds S4, a network peer that uses 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 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.

Clearly, reasoning with uncertainty introduces additional processing burden, which may affect response times. In any given application, the gain in accuracy or risk reduction must be traded off against the processing burden. It is not always the case that response speed overrides all accuracy concerns. Even when response speed is critical, it may be possible to develop an anytime system that makes use of a rapid initial response, and makes appropriate adjustments when a more complete response becomes available.

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. For a more detailed account, the interest reader should refer to [23]

4. Discussion

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.

Bibliography

1. 2006 Department of Defense Chief Information Officer STartegic Plan, version 1. Available at http://www.dod.mil/cio-nii/docs/DoDCIO_Strat_Plan.pdf.
2. SOAP Version 1.2 and related W3C Recommendations. Latest versions available from: http://www.w3.org/2000/xp/Group/#drafts. SOAP version 1.1 and related W3C Member Submissions available from: http://www.w3.org/Submission/
3. Web Services Description Language (WSDL) Version 2.0. Latest versions available from: http://www.w3.org/2002/ws/desc/#public_drafts. WSDL version 1.1 and related W3C Member Submissions available from: http://www.w3.org/Submission/
4. Paolucci, M., et al. Importing the Semantic Web in UDDI. in Proceedings of the Web Services, E-Business and Semantic Web Workshop, at the 14th International Conference on Advanced Information Systems Engineering (CAiSE'02). 2002. Toronto, Ontario, Canada.
5. Martin, D., et al. OWL-S: Semantic markup for Web services. 2004. Latest version available from: http://www.ai.sri.com/daml/services/owl-s/1.2/.

6. Domingue, J., H. Lausen, and D. Fensel. ESSI Web Services Modeling Ontology Working Group. [cited; Available from: http://www.wsmo.org.
7. Battle, S., et al. Semantic Web Services Framework (SWSF) Overview. 2005 [cited; Available from: http://www.daml.org/services/swsf/1.0/overview/.
8. SAWSDL - Semantic Annotations for WSDL and XML Schema. Latest version available from: http://www.w3.org/2002/ws/sawsdl/#Publications.
9. Akkiraju, R., et al. Web Service Semantics - WSDL-S. [Web Page] 2005 [cited; W3C Member Submission 7 November 2005, Version 1.0]. Available from: http://www.w3.org/Submission/WSDL-S.
10. Smith, B. Ontology: Philosophical and Computational. 2004 [cited 2005 January 15]; Draft]. Available from: http://ontology.buffalo.edu/smith//articles/ontologies.htm
11. Costa, P.C.G., Bayesian Semantics for the Semantic Web, in Department of Systems Engineering and Operations Research. 2005, George Mason University: Fairfax, VA, USA. p. 312.
12. Laskey, K.B. MEBN: A Logic for Open-World Probabilistic Reasoning. [Research Paper] 2005 [cited; Available from: http://ite.gmu.edu/klaskey/index.html.
13. Heckerman, D., C. Meek, and D. Koller, Probabilistic models for relational data, in Technical Report MSR-TR-2004-30. 2004, Microsoft Corporation: Redmond, WA.
14. Costa, P.C.G., K.B. Laskey, and K.J. Laskey, PR-OWL: A Bayesian Framework for the Semantic Web, in Proceedings of the First Workshop on Uncertainty Reasoning for the Semantic Web (URSW2005), held at the Fourth International Semantic Web Conference (ISCW2005). 2005: Galway, Ireland.
15. Costa, P.C.G. PR-OWL Website. 2006 [cited; Available from: http://www.pr-owl.org.
16. Costa, P.C.G. and K.B. Laskey, Multi-Entity Bayesian Networks without Multi-Tears. 2005, The Volgenau School of Information Technology and Engineering, George Mason University. p. 20.
17. Reference Model for Service Oriented Architecture 1.0, OASIS Standard, 12 October 2006. Available from: http://www.oasis-open.org/specs/index.php#soa-rmv1.0.
18. Student Pamphlet: Soil, Trafficability, DMS ST no. 222. 1986, Defense Mapping School.
19. OGC. The OpenGIS Guide, Introduction to Interoperable Geoprocessing. 1996 [cited; Part I of the Open Geodata Interoperability Specification (OGIS)]. Available from: http://members.tripod.com/~ibank/OGIS/guide1.htm#toc.
20. Raghavendran, S., GIS Interoperability: The "Safe Way", in GSDI 2004 Conference. 2004: Bangalore, India.
21. SEDRIS. Environmental Data Coding Specification. 2007 [cited; Available from: http://www.sedris.org/edcs.htm.
22. Wright, E.J., Understanding and Managing Uncertainty in Geospatial Data for Tactical Decision Aids. 2002, George Mason University: Fairfax, VA, USA.
23. Costa, P.C.G., K.B. Laskey, and K.J. Laskey, Probabilistic Ontologies for Efficient Resource Sharing in Semantic Web Services, in Second Workshop on Uncertainty Reasoning for the Semantic Web (URSW 2006), held at the Fifth International Semantic Web Conference (ISWC 2006). 2006: Athens, GA, USA.