Ontology Driven Interoperability – M&S Applications

Andreas Tolk, Ph.D., Saikou Y. Diallo, Charles D. Turnitsa
Virginia Modeling Analysis & Simulation Center (VMASC)
Old Dominion University, Norfolk, Virginia
atolk@odu.edu, sdiallo@odu.edu, cturnits@odu.edu

ABSTRACT

Interoperability is not a cookie cutter function and in fact, it can be achieved on several layers. The levels of conceptual interoperability model (LCIM) identify six layers of system interoperation: technical, syntactic, semantic, pragmatic, dynamic, and conceptual. Standards such as the Distributed Interactive Simulation (DIS, IEEE1278) and the High Level Architecture (HLA, IEEE1516) are very efficient part-solutions that address some layers of the LCIM. However, there is still a need for tools and frameworks that span across all layers. The current research on ontologies – an attempt to formulate an exhaustive and rigorous conceptual schema within a given domain – has the potential to become an overarching solution embracing existing working solutions.

This tutorial will start by presenting the six levels of the LCIM and showing where current interoperability solutions such as DIS and HLA fit and to what degree they are lacking. After giving an overview of the ontological spectrum, the paper will introduce some current developments in the ontology domain, and give an overview of frameworks and methods such as the Resource Description Framework (RDF) and the Ontology Web Language (OWL). The third section will demonstrate some M&S enhancements and applications of ontological ideas to increase interoperability of M&S applications. Finally, the tutorial will show how the different aspects can grow together to become a framework for interoperable solutions covering aspects of all six layers.

ABOUT THE AUTHORS

Andreas Tolk is Associate Professor in the Faculty for Modeling, Simulation, and Visualization at the Engineering Management Department of the College of Engineering and Technology at Old Dominion University (ODU) in Norfolk, Virginia. He has over 16 years of international experience in the field of Applied Military Operations Research and Modeling and Simulation of and for Command and Control Systems. He is affiliated with the Virginia Modeling Analysis & Simulation Center (VMASC). His domain of expertise is the integration of M&S functionality into real world applications based on open standards. He received a Ph.D. (1995) and an M.S. (1988) in Computer Science from the University of the Federal Armed Forces in Munich, Germany.

Saikou Y. Diallo is a Ph.D. candidate at the Virginia Modeling Analysis and Simulation Center (VMASC) of the Old Dominion University (ODU). He received his B.S. in Computer Engineering (2003) and his M.S. in Modeling & Simulation (2006) from ODU. His Ph.D. research under Tolk focuses on the domain of Model Based Data Engineering and web services for M&S applications.

Charles D. Turnitsa is a Ph.D. candidate at the Virginia Modeling Analysis and Simulation Center (VMASC) of the Old Dominion University (ODU). He received his B.S. in Computer Science (1991) from Christopher Newport University (Newport News, Virginia), and his M.S. in Modeling & Simulation (2006) from ODU. His Ph.D. research under Tolk focuses on the domain of dynamic and fractal ontology models for M&S interoperability.

Ontology Driven Interoperability – M&S Applications

Andreas Tolk, Ph.D., Saikou Y. Diallo, Charles D. Turnitsa
Virginia Modeling Analysis & Simulation Center (VMASC)
Old Dominion University, Norfolk, Virginia
atolk@odu.edu, sdiallo@odu.edu, cturnits@odu.edu

INTRODUCTION

Interoperability is not a cookie cutter function. There are several layers of interoperation possible between information technology systems. Several approaches have been published to capture applicable layered approaches, among them the Levels of Information System Interoperability (LISI) model and the NATO Interoperability Model. Modeling and Simulation (M&S) systems are special due to the fact that they are based on a model, which is defined as a "*purposeful abstraction of reality*." To deal with the special aspects of M&S, the term composability is often used. Petty and Weisel (2003) offer the following working definition: "*Composability is the capability to select and assemble simulation components in various combinations into simulation systems to satisfy specific user requirements. The defining characteristic of composability is the ability to combine and recombine components into different simulation systems for different purposes.*"

Both currently accepted IEEE standards for distributed simulation systems, the Distributed Interactive Simulation (DIS, IEEE1278) and High Level Architecture (HLA, IEEE1516), only support these ideas to a limited extent. They target interoperable solutions on the implementation level, not the modeling level. However, as stated in Tolk (2006): "*Interoperability of Simulation Systems requires Composability of Conceptual Models!*" This, as a logical next step, requires engineering methods to document the ideas, assumptions, and constraints underlying the purposeful abstraction. While this documentation is very helpful to developers, the ultimate goal is to capture the resulting metadata in a machine-understandable form, so as to enable systems to interoperate in a dynamic and semi-intelligent fashion.

Emerging from the ideas of the semantic web, such as described by Daconta et al. (2003), ontologies have the potential to provide a way forward. Daconta et al. describe ontologies as a solution for semantic interoperability on the web, and introduce an ontological spectrum of solutions ranging from weak to strong semantics. Nonetheless, these solutions are static in nature, since they only address the sphere delimited by the predetermined information exchange requirement between systems. This disadvantage is overcome by combining ontological ideas with web services.

Research in the Semantic Web area has given birth to a series of applicable tools and framework some of which are partly standardized and others are partly de facto standards in their domains. Some of the frameworks are the extensible Mark-up Language (XML), the Research Description Framework (RDF), the Defense Advanced Research Project Agency (DARPA) Agent Markup Language (DAML), the Ontology Web Language (OWL), and OWL for Services (OWL-S).

While ontologies are important for the semantic web ideas, they become essential for M&S applications. As pointed out earlier, conceptual models document ideas, constraints, and assumptions derived from the purposeful abstraction of reality in the modeling process. Ontologies therefore have the potential to describe the reality valid within a model. In order to reach the desired composability of conceptual models, the first step is to align and capture these different views of reality using engineering methods. However, the ontological spectrum needs to be extended to support all layers of interoperation and this paper discusses potential applicable extensions.

Finally, there are several research initiatives going on in related domains, such as ontologies and the Model Driven Architecture (MDA) of the Object Management Group (OMG), or composition of web services based on ontological constraints. This paper will present some of the more pertinent ones.

In summary, ontologies are not a silver bullet or the golden solution to composability and interoperability, but they offer new solutions to reach higher levels of interoperations that are not sufficiently supported by current M&S standard solutions.

LAYERS OF INTEROPERATION

The ability to exchange information between systems is paramount not only in the M&S world but also in the business world. Systems interoperate at different levels and it is important to first clearly differentiate between systems exchanging bits and bytes and systems exchanging concepts. The ultimate goal is to attain a level of conceptual interoperability and ontology is a path to get there.

In order to cope with the different layers of interoperation, the Levels of Conceptual Interoperability Model (LCIM) was developed at the Virginia Modeling Analysis & Simulation Center. Tolk and Muguira (2003) presented the first version of the LCIM during a Simulation Interoperability Workshop. Other scientist and researchers refined the model and contributed to its current form. In particular Page et al. (2004) suggested defining composability as the realm of the model and interoperability as the realm of the software implementation of the model. In addition, their research introduces the notion of *"integratability"* when dealing with the hardware and configuration side of connectivity. Following this categorization, we recommend the following distinction when dealing with interoperation:

- *Integratability* contends with the physical/ technical realms of connections between systems, which include hardware and firmware, protocols, etc.

- *Interoperability* contends with the software- and implementation details of interoperations; this includes exchange of data elements based on a common data interpretation.

- *Composability* contends with the alignment of issues on the modeling level. The underlying models are purposeful abstractions of reality used for the conceptualization being implemented by the resulting simulation systems.

Figure 1 shows the current LCIM including the relation to the ideas described in Page et al. (2004) and the layers for modeling/abstraction, simulation/ implementation, and network/connectivity. The currently used LCIM version distinguishes between the following layers:

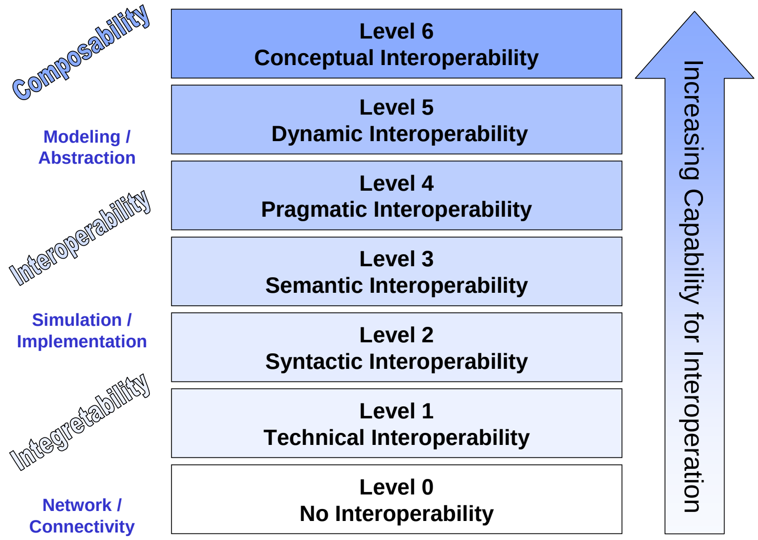

Figure 1: Levels of Conceptual Interoperability Model

- Level 0: Stand-alone systems have *No Interoperability*.

- Level 1: On the level of *Technical Interoperability*, a communication protocol exists for exchanging data between participating systems. On this level, a communication infrastructure is established allowing the exchange of bits and bytes; the underlying networks and communication protocols are unambiguously defined.

- Level 2: The *Syntactic Interoperability* level introduces a common structure to exchange information, i.e., a common data format is applied. On this level, a common protocol to structure the data is used; the format of the information exchange is unambiguously defined.

- Level 3: If a common information exchange reference model is used, the level of *Semantic Interoperability* is reached. On this level, the meaning of the data is shared; the content of the information exchange requests are unambiguously defined.

- Level 4: *Pragmatic Interoperability* is reached when the interoperating systems are aware of each other's methods and procedures. In other words, the use of the data – or the context of its application – is understood by the participating systems; the context in which the information is exchanged is unambiguously defined.

- Level 5: As a system operates on data over time, the states of that system changes along with the assumptions and constraints that affect its data interchange. At the *Dynamic Interoperability* level, interoperating systems are able to comprehend and take advantage of the state changes that occur in the assumptions and constraints that each other are making over time. Simply stated, the effect of the information exchange within the participating systems is unambiguously defined.

- Level 6: Finally, if the conceptual models – i.e. the assumptions and constraints of the "purposeful abstraction of reality" – are aligned, the highest level of interoperability is reached: *Conceptual Interoperability*. This requires that conceptual models be fully documented based on engineering methods enabling their interpretation and evaluation by other engineers. In other words, we need a "fully specified but implementation independent model" as requested in Davis and Anderson, 2003, and not just a text describing the conceptual idea.

The LCIM has been applied for Verification and Validation for Coalition Crisis Management and for Energy Management purposes and was even referred to in the final report on System-of-Systems Interoperability evaluations conducted by the Carnegie Mellon University (Morris et al., 2004).

When evaluating current solutions targeting simulation systems interoperability, it becomes obvious that they focus on the implementation level and not on the modeling level. In fact, the M&S community has directed a lot of attention and effort on simulation interoperability and less on modeling composability. Tolk and Muguira (2004) and Zeigler et al. (1999) have examined the contribution of the two dominant standards for M&S applications, DIS and HLA, to the layers of interoperation and concluded:

- The IEEE 1278 standard does not define any infrastructure software. However, the way information has to be exchanged between participating systems is very well defined in the form of Protocol Data Units (PDUs). While technical interoperability (level 1) is assumed, these PDUs support syntactic and semantic interoperability (levels 2 and 3). The information is unambiguously defined; however, there is no way in DIS to support pragmatic or higher interoperability. The DIS standard explicitly states that every simulator is responsible for how the information is used within the system. There is no central government, no central node. Some ideas, in particular dead-reckoning algorithms point to wards support of pragmatic and even dynamic interoperability, but these ideas are necessary, not sufficient.

- The IEEE 1516 standard overcame one problem of the PDUs by adding more flexibility: While the information exchange was unambiguously defined by their use, each new data element requires the standard to be extended. Instead of standardizing the information that systems exchange, HLA defines rules, software interfaces, and software services to describe how information is to be exchanged. While it supports unambiguous information on higher levels of interoperation, it does not ensure it. As a result, HLA only supports the layers up to the syntactic level of interoperability. If the Federation Development and Execution Process (FEDEP), is applied consistently, and all results are documented in machine-readable protocols and artifacts, higher layers of interoperations can be supported, however this is not the rule. The Extensible M&S Framework (XMSF) research team made several recommendations, but the current version of the standard is still implementation driven.

In summary, the M&S community has focused successfully on the interoperability of simulation systems, but now is the time to reinforce the research on the composability of models.

With DIS and HLA and the advent of XML most systems can interoperate at the semantic level. In order to move towards the ultimate goal of conceptual interoperability systems must describe their underlying concepts and processes in a fashion understandable by humans but most importantly by machines. The methods and tools developed for the semantic web seem to be a good core to start from and move forward. Since these methods and tools are designed to express ontologies, the next section will take a more in-depth look at ontologies really are.

INTRODUCTION TO ONTOLOGIES

Within this section, we introduce the main ideas underlying the concept of ontologies. In order to fully comprehend the concept of ontologies, it is important that they are examined as part of an ontological spectrum ranging from weak to strong semantic interoperability. The balance of this section will first describe the spectrum before focusing on tools and frameworks for developing ontologies.

Ontologies must be evaluated in the context of the so-called Semantic Web initiatives. While the Internet based on the Hypertext Mark-up Language (HTML) was a "*web of documents*" focused on the distribution of documents and their independent display and presentation to a human user, it changed into a "*web of data*" with the advent of the extensible Markup Language (XML). Consequently, the idea of a semantic web in which web services are able to dynamically communicate data and its meaning to each other, as envisioned among other by Daconta et al. (2003) now seems possible. The ideas borne out of this vision are directly applicable to support higher layers of interoperation currently not addressed by DIS or HLA. The authors are convinced that these and similar engineering principles for conceptual modeling must be applied in order to support higher layers in a way that

- On the short term the documentation can easily be shared between federation developers, and
- On the mid term metadata to capture these ideas can be developed enabling machine-parseable and machine-understandable definition of information exchange requirements between M&S applications.

However, while ontologies target the unambiguous description of the information sphere understandable by M&S applications – including the ability to manipulate this sphere in accordance with M&S algorithms – there are additional dynamic aspects of the system that ontologies do not capture because of their static nature. Furthermore, ontologies need to be coupled with new approaches in order to deal with multiple levels of resolution (including aggregation and disaggregation functions between them), multiple concepts of time, and other M&S federation specific challenges.

While it is important to understand what ontologies are, it is crucial to know that building ontologies is a process. In fact ontologies are part of a spectrum that is introduced next.

Introduction to the Ontological Spectrum

The ontology spectrum was introduced by Daconta et al. (2004) and describes a range of semantic models of increasing expressiveness and complexity. Concepts often perceived to be independent are put into a common ontological context, such as taxonomy, thesaurus, conceptual model, ontology, and logical theory. Furthermore, there is a lot of confusion concerning the terminology, such as how to distinguish between:

- Syntax, semantics, and pragmatics,
- Data objects, classification objects, terminology objects, meaning objects, and relations
- Intension and extension,
- Ontology and epistemology, and
- Term, concept, and real world referent.

Some of these terminology confusions such as syntax (structure of data), semantics (meaning of data), and pragmatics (context of the use of data) have been dealt with earlier in this paper. The ISO/IEC standard 11179 specifies data objects, classification objects, terminology objects, meaning objects, and their relations in detail. The standard defines the following terms to describe registries of data for reuse.

- *Conceptual Domains* define sets of categories, which are not necessarily finite, where the categories represent the meaning of the permissible values in the associated value domains. These are the concepts representing objects of the real world within the conceptual model ("military units" for example)

- *Data Element Concepts* describe the contextual semantics, i.e., the kinds of objects for which data are collected and the particular characteristic of those objects being measured. They describe the classification domain, e.g., that "unique unit identifiers" are necessary to describe the concept of military units.

- *Value Domains* comprise the allowed values for an associated data element. They comprise symbolic information on the representation level; in other words, they describe the valid terminology.

- *Data Elements* are the basic containers for data such as used in data models. Data elements comprise contextual information on the representation level. These data are terms representing the concepts within the M&S application.

The terms intension and extension capture the underlying motivation behind a modeling effort. Intension refers to the set of all possible things that a word could represent. *Modeling by intension* means that the model should list all of the enumerated types that could ever exist. Extension refers to the set of all the actual things that a word could represent. *Modeling by extension* means including all of the actual types necessary to describe entities.

Ontologies are the result of modeling efforts, which means that they are used to abstract from reality. The intention of ontologies is to capture an abstraction of the real world. The real world referent is modeled by concepts of the ontology. The application captures these concepts by terms. Terms and their relations are captured as epistemologies, which are typologies of ontologies. While *ontologies deal with concepts* (or meaning objects) at the modeling level, *epistemologies deal with terms* (or data entities) at the implementation level. Ontologies and epistemologies are as often confused as concepts and terms or conceptual domains and representing data elements. The lack of a concise use of concepts and terms often results in not being able to understand the problems of the other side. Among these typical problems are

- *Synonyms* are not identified, because they use different value domains (typically unambiguous names and codes or alternative keys, all being identifiers), or
- *Aggregates* are not identified, because the underlying concept structure is not identified explicitly (such as unit readiness status connected to ammunition, personnel, and petrol-and-liquids data of higher resolution).

The Ontological Spectrum

The ontological spectrum of the Semantic Web is an expressive, comprehensive, and powerful form of data engineering. It is not a radically new concept, but it builds on traditional data modeling techniques and combines and transforms them into powerful ways of expressing rich relationships in a more thoroughly understandable manner. It is therefore essential to understand the underlying concepts in order to understand the importance of ontological ideas for M&S applications and composability/interoperability issues.

In order to align higher concepts, their meaning, and their relationships, several methods can and must be applied, that all deal with a slightly different view of the challenge. Daconta et al. (2003) introduced the idea of the ontological spectrum, which we use here as well. We present a simplified version using only a subset of applicable methods. Figure 2 shows a simplified version of the ontological spectrum used in this paper.

The methods are ordered in the spectrum from weak to strong semantic interoperability. As mentioned before, the ontological spectrum uses methods and tools of traditional data engineering and applies and enhances them in the context of the Semantic Web. The ontological spectrum starts with simple lists and enumerations and goes via thesauri and taxonomies to logical theory. All methods are used to formalize the specification of the underlying concepts, which is the working definition for ontologies:

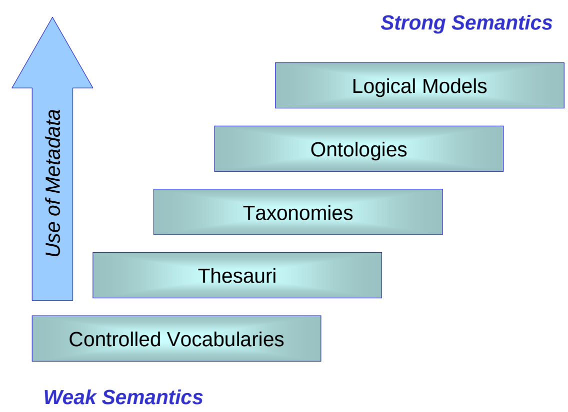

Figure 2: Simplified Ontological Spectrum

Ontologies are formalizations of specifications of conceptualizations! All three parts of this definition are important:

- The objective of ontologies is to document the *conceptualization*, which is another word for the result of the modeling process.
- This is done in a *specified way*, which means the application of engineering methods guided by rules and methods.
- The result is *formalized*, which means that machines and computers can not only read the result, but also make sense out of it in the context of their applications.

As formulated in Tolk and Blais (2005) for practical applications: *“If a formal specification concisely and unambiguously defines concepts such that anyone interested in the specified domain can consistently understand the concept’s meaning and its suitable use, then that specification is an ontology.”*

Controlled Vocabularies

Dictionaries and glossaries are lists of controlled vocabularies and are among the weakest semantics in the ontological spectrum. All allowed terms and their meanings are completely enumerated, well-defined and controlled by a common registration authority. Sometimes, additional information, such as pronunciations, etymologies, and variant spellings, are given or cross-references are included, but the overall structure is a flat list. However, for some applications, such a list of allowed values may be sufficient (see, e.g., the list of allowed PDUs in IEEE1278). Furthermore, these lists enumerate terms, not underlying concepts.

Thesauri
Thesauri are controlled vocabularies arranged in a known order and structured so that equivalence, homographic, hierarchical, and associative relationships among terms are displayed clearly and identified by standardized relationship indicators. The primary purpose of thesauri is to facilitate retrieval of documents and achieve consistency in the indexing of written or otherwise recorded documents and other items. As with controlled vocabularies the focus is on the terms, not the underlying concepts.

Taxonomies
Taxonomies are tree structures of classifications for a given set of objects. At the top of these structures are single classifications, which are the root nodes that apply to all objects. Nodes below these roots are more specific classifications that apply to subsets of the total set of classified objects. The main purpose is the classification of terms. The higher a term, the more universal it is; that means that leaves are the most specific terms of taxonomies. Taxonomies are the first form reflecting the idea of concepts.

Ontologies
Ontologies formulate an exhaustive and rigorous conceptual schema within a given domain. Although these are typically hierarchical data structures containing all the relevant entities, they are not necessarily trees. In addition to entities, ontologies contain relationships and rules (such as theorems and regulations) within those domains. Ontologies capture the meaning of the underlying concepts.

In practice it is agreed that ontologies should contain at a minimum not only a hierarchy of concepts organized by subsumption relations, but also other 'semantic relations' that specify how one concept is related to another. The main purpose is the definition of entities and their relationships.

Logical Models
Logical models are semantically the strongest methods of the ontological spectrum. Description logic, first order logic, and modal logic belong into this group. Furthermore, logical models can be separated into frame-based and axiomatic models. Frame-based models use an associated-node structure representing the logical expressions. Entity classes, attributes, properties, relations/associations, and constraints/rules are in the center. Axiomatic approaches make axioms and rules explicit, which means that they use languages exposing logical expressions.

The selection of a suitable method to semantically describe and align services and applications is driven by the constraints of the applications themselves. However, it should be pointed out that even the strong semantics support the LCIM only up to the level of pragmatic interoperability. Beyond that, ontologies and logical models do not sufficiently capture the dynamics of M&S systems so far. The methods described so far summarize the state-of-the-art in semantic alignment. The following section will present some of the currently used tools and framework supporting the ideas presented earlier.

Ontological Tools and Frameworks

The goal of ontological tools and frameworks is to create a language for systems to self-describe. The availability of ontologies makes it possible for machines to make inferences and deduction about each other. In order to achieve that goal, systems must expose their underlying structure to each other by describing classes and their properties as well as the relationships between classes in a standardized way.

As a result, recent efforts led to the creation of a family of XML-based frameworks and language that build on top each other. One of the main driving forces behind the motion towards ontology development has been the idea of a Semantic Web, as envisioned by Berners-Lee and others (2001). It has been clear from the start that a depiction of the conceptual meaning behind the data is required for systems and agents to make effective use of it. Ontology based tools and frameworks ensure interoperation between systems, collections of data, and agents, enabling the confluence of both. As discussed above, the specification of such a conceptualization is ontology.

Berners-Lee (2001) further envisions an automatic interaction between systems and agents navigating the web will interact with each other. To enable this idea, the specified conceptualization – the ontology – must be available at the system/data level, which means, ontologies must be readable and understandable by machines.

Systems have relied on data describing data for years. It is commonly referred to as metadata, and is often designed for a specific purpose, and hence, with a specific format. In order to enable semantic web ideas, metadata must represent the underlying ontology in a standard format. Moreover, the format must be in some way self-describing, as it is not practical to expect all of the possible descriptions for all possible data to exist in some known list bound by intension. Rather, a self-describing system must exist, so that all possible descriptions can exist by extension.

Given the three characteristics of what is required for the semantic web, a number of tools and methods have been developed that address what is needed for a formalized specification of a conceptualization that is portable along with the data, and is self-describing. Examples are defined in the following sections.

Extensible Markup Language (XML)

XML resulted from improvements – mainly simplifications allowing the easier application, of the Standard Generalized Markup Language (SGML). SGML itself was developed out of an IBM project, from the 1960s, for inserting tags that could be used to describe data and evolved into the ISO Standard 8879. Different features or sections of a document could be marked as serving a particular function. One of the most successful descendants of SGML, prior to the development of XML, is the Hypertext Markup Language (HTML). HTML is the language that currently makes most of the World Wide Web (WWW) documents possible.

Since its introduction in 1998, XML has become widely and almost universally adopted by all levels of data and system modelers and developers. It has a number of very attractive strengths that have induced widespread adoption, but it also has a few weaknesses.

XML is important to the development of portable ontologies because it is extensible (the markup tags are self describing) universally readable (it is based on Unicode), and highly portable (it can be transferred via almost any medium, and its self-contained and embedded nature make it a perfect partner to the data it is describing).

While XML is a good framework to start from, it is lacking in two main areas when it comes to expressing ontologies:

- The first missing element is a corollary to the fact that XML is self-describing. As a result, each document has a tag set valid only within a context. XML resolves this issue partly by introducing the notion of namespaces. However there is still a need for a framework for describing namespaces.
- Second, XML only deals with the description of data documents and not the data needs and capabilities of agents or systems.

The first missing element namely a system for organizing the open, self-describing, nature of XML for ontological purposes is addressed almost by RDF.. The description of data needs and capabilities for both agents and systems starts with DAML, and builds increasingly to OWL-S.

Relying on XML as a technology to enable interchange of data between systems does not, in itself, satisfy any of the requirements for reaching the levels of the LCIM, above syntactical interoperability (level 2). It is possible to reach semantic interoperability (level 3) with the alignment of tag sets between the different systems.

Resource Description Framework (RDF)

Because of XML's inadequacies the W3C developed RDF as a standard framework to capture relatively simple ontologies. By definition, RDF is a standardized method for describing resources. In simple terms, a description is a statement that relates what one is describing (the subject of the description) to a statement defining it (the object of the description). In RDF a description is made of three parts. The "subject" (what you are describing), the "object" (the definition) joined together by the "predicate." The predicate links the subject to the object thus giving a mode to the relationship. The set of subject, object, and predicate is commonly referred to as an RDF triple.

RDF triples mostly rely on Universal Resource Identifiers (URI) to provide a physical address for each member of the RDF triple. A URI, although originally envisioned as being quite useful, is not used universally within RDF. Other possibilities include simple terms, literals, and probably in the future Extensible Resource Identifiers (XRI).

It is worth pointing out that RDF was originally described as a graphical description method, depicting an RDF triple as a subject node and an object node, and a directed arc as the predicate. Since a directed graph is not adequate for automated consumption, a machine-readable format was developed under the W3C. This resulted in the creation of an XML-based RDF Schema commonly known as the RDF/XML schema, or RDFS.

RDFS describes resources using URIs, simple terms, and literals. Literals are, essentially simple terms, however they may be typed. In that case, they also include a reference to a description of their type. Literal types in RDF are usually similar to those found within programming languages – integer, string, *etc.*

RDFS follows the basic RDF triple structure. However, it is useful to consider the predicate as a property, and the object as a value for that property. Figure 3 shows a comparative view of both. One of the effects of being able to show how objects, or property-values, can be related to subjects is that those property-values (or objects) can be treated as subjects themselves. A machine can then infer that certain properties are transitive

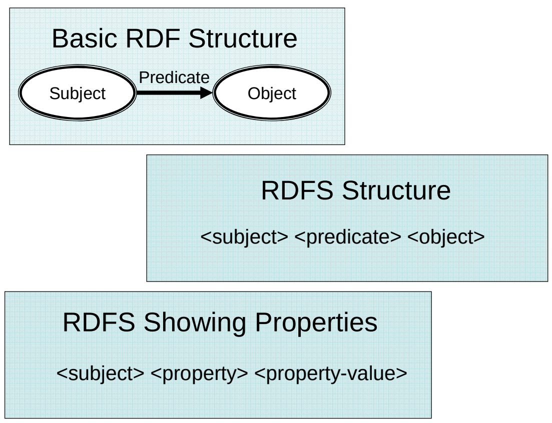

Figure 3: RDF and RDFS Structure

Considering RDFS in this way shows how it can describe complex documents. So far, RDFS can only establish a number of triples, each giving a one-to-one establishment of some property/property-value pairing to a subject. However, there are a few standard RDFS properties that open up new possibilities. Two of the most important are

- *Containers:* they allow for higher order (higher than binary) pairings taking place by replacing the property-value with a list of property-values. This creates a situation where a one-to-many pairing is possible.

- *Types*: are even more powerful, for they allow for the establishment of categorization, including classes. Along with classes, also subclasses are supported. This allows for the construction of complex relationships where, for instance, the property/property-value pairings that describe a class are also inherited by the sub-class, allowing for the sorts of structuring that is required in the ontological spectrum for thesauri, taxonomies, and the higher levels.

Relying on RDFS as an enhancement to XML for system-to-system data interchange has the potential to increase the level interoperation between systems. If the RDF structures are well formed and complete enough to help describe the semantic meaning of the data being interchanged, and each system is capable of making use of those RDF structures – by operating over a similarly structured RDFS – then the semantic level (level 3) of interoperability can be reached within the LCIM.

Ontology Web Language (OWL)

The purpose of OWL, similar to that of RDFS is to provide an XML-based vocabulary to express ontologies (classes, properties and relationships among classes). However, RDFS does this at a very rudimentary level and is not rich enough to reflect the complex nature of many systems.

DARPA tried to overcome these shortcomings with the development of DAML, an RDFS-based language that makes it possible to describe systems at a higher level of detail. DARPA later combined DAML with the European Community's Ontology Interface Layer (OIL) to create DAML+OIL, a full-fledged ontology modeling language. A revision of DAML+OIL, lead by the W3C resulted in the creation of OWL, a new standard for expressing the ontology of a system. Some of OWL's main capabilities include:

- *Defining property characteristics*: RDFS defines a property in terms of its range (possible values), its domain (class it belongs to) and as a "subProperty-Of" to narrow its meaning. OWL makes it possible to describe the nature of properties by defining them as symmetric, transitive, functional, inverse, or inverse functional.

- *Object property versus data type properties*: In owl, as opposed to RDFS, object properties and data type properties are members of two disjoint classes. Object properties are used to relate resources to one another, while data properties link a resource to a literal (number, string, etc…) or a built in XML schema data type.

- *Property restriction*: OWL classes have a higher level of expressiveness than RDFS classes from which they are inherited. OWL classes allow restrictions on global properties. A property can have all of its values belonging to a certain class, at least one value coming from a certain class or simply have a specific value. OWL also allows restrictions on the cardinality of properties, by simply specifying cardinality, minimum cardinality, or maximum cardinality.

- *Class relationships*: OWL classes have most of the properties of sets in set theory (union, disjoint, complement, etc…)

In summary, OWL increases the power of inference that systems can make about one another. However, in order to reach a higher level of interoperability above the pragmatic level, OWL itself is not sufficient.

OWL for Services (OWL-S)

OWL provides a powerful framework for expressing ontologies. OWL-S is not a new method or tool but an application of OWL to describe services in a much more detailed fashion than the current Web Service Description Language (WSDL).[1]

Figure 4 presents the ontology of a service, which is comprised of three components: profile, model, and grounding.

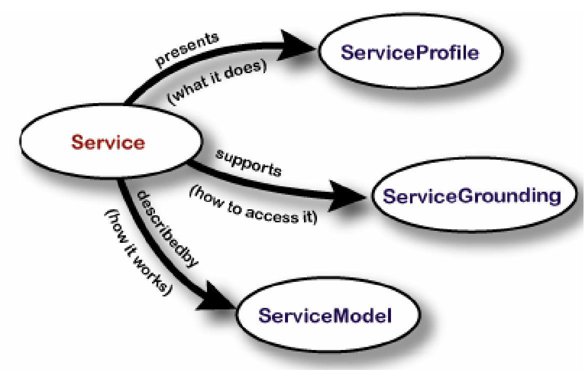

Figure 4: Ontology of a Service

- The *service profile* provides a concise description of the capabilities implemented by the service (What it does). It allows clients and search agents to determine whether the service fulfills their needs.

- The *service model* describes the behavior and state-changes of a service (How it works). To do this, it specifies the inputs, outputs, preconditions, and effects (IOPE).

- The *service grounding* defines how to make use of a service (How to access it). Because WSDL is suitable to express the grounding of a service, such as formats and protocols, OWL-S applies these ideas as well.

The combination of OWL methods for the service profile and the service model and WSDL method for the service grounding results in the "best of both worlds." OWL-S provides a semantic description of services while WSDL specifies how to access the services. Potential clients can use the service profile to discover the service, the service model to understand the behavior of the service at the abstract level, and finally use WSDL to identify the protocols to bind and interact with the service at the implementation level. Figure 5 shows the interplay.

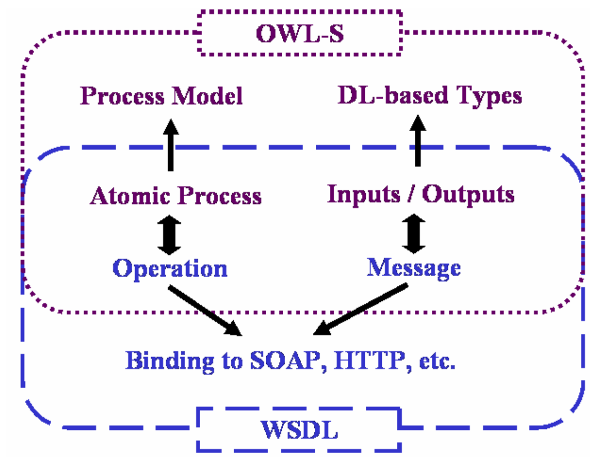

Figure 5: Interplay of OWL and WSDL Methods[2]

OWL-S enables a new level of interoperation for services. It makes it possible to automatically discover and invoke web services. Most importantly it supports service composition and interoperation thus allowing more complex tasks to be performed in an automated fashion. While OWL itself remains on the pragmatic level of interoperability, the use within OWL-S clearly tends toward dynamic interoperability. In particular the Service Model in OWL is a first step towards the standardized formal representation of the underlying conceptual model required before.

The next chapter will describe how these capabilities are applicable to the M&S world. In order to do so however, the special requirements for M&S applications must be considered. As stated in Tolk (2006), the

[1] WSDL is a W3C standard that describes a "protocol-and-encoding independent mechanism for web service providers to describe the means of interacting with service."

[2] Figure 4 is courtesy of http://www.daml.org. See link for more information on OWL-S specifications

M&S community having focused successfully on solving interoperability challenges on the simulation/implementation level must now focus on composability challenges on the modeling/abstraction level. The core methods implementing the ontological spectrum are a promising start to this endeavor.

APPLICATION TO M&S

Mizoguchi (1996) stated that ontology is an important area of research and application for various areas in information science where the specific and unambiguous meaning of data needs to be captured. Entities within a domain can be understood in terms of their conceptual meaning as well as their relationships to each other. This supports system designers, system integrators, and system evaluators.

While this all-encompassing definition may or may not be too broad for all of computer science, ontology is certainly relevant for M&S. The importance of ontology work for M&S can be seen in two areas. The first area is related to the fact that within a simulation system, reality is completely as the system defines it. A simulation system is not only responsible for reporting on the behavior of objects, it is responsible for creating that behavior, and also creating the interactions between the object and its environment. A strong conceptualization of those behaviors and interactions is required. Since this conceptualization needs to be accessed, it should also be well specified.

The relevance of ontology work to M&S is also evident is the area of conceptual meaning. Simulation systems are required to exchange information with each other concerning their objects, behaviors, and environments. This information must be understandable by all systems involved in the exchange. Due to potential problems with misalignment of conceptual meaning behind terms, and even misalignment of terms for the same conceptual meaning, a clear representation of the underlying concepts (which should have near universal meaning to a domain area, whereas terms do not have this universality) is required. A strong case for the better modeling of concepts for M&S is made in Robinson (2006).

To support these two areas, there are some aspects of ontology application that are of particular interest to M&S. These are listed below, and are in fact supported by two lines of research being conducted at VMASC.

Concepts and Rules

The traditional view of an ontological model is that it consists of entities and relations. Given the requirements for M&S that have been described above, however, there may be room for more within an ontology system serving this application area. By introducing concepts and rules into the ontological description, we can begin to address these requirements.

The ability to describe the concepts behind the entities (terms) of the ontology alleviates the problem of conceptual misalignment between systems. It also eliminates the misalignment of terms applied to the same concepts within several systems, because of the nature of concepts – they are generally accepted universals of meaning within an application area.

The ability to model the rules of a system gives us a peek into the dynamic nature of a simulation system. The relations of the ontology show which of the entities of that ontology are either related to each other – as a parent-child relationship, or related with some semantic meaning for the application area. By understanding the rules of the ontology, we can see when these semantic rules are to be applied, and to what extent. We can also define situational relations.

Grammar and Web Services Profiles

In terms of M&S, systems are composed of many subsystems which interact with each other in a fashion hidden to the user. In fact an M&S tool is a system of systems that dynamically interoperate with one another. This concept is very similar to that of a series of services distributed around the world, communicating via a global network. By treating systems as services, it is possible to use OWL-S to describe the ontology of a system in terms of its profile, process, and grounding.

However, OWL-S does not provide a way to link systems/services together to form a family of systems/services. A family of systems is defined as a set of systems working together to perform a given task. In order to specify that a set of systems belong to the same family and provide ontology for the family, a grammar is needed. Wikipedia defines a grammar as "an abstract structure that describes a formal language precisely, i.e., a set of rules that mathematically delineates a (usually infinite) set of finite-length strings over a (usually finite) alphabet. Formal grammars are so named by analogy to grammar in human languages."

This sort of service orchestration is currently a manual and time-consuming effort that requires developers to examine each system separately and decide how to combine them in a meaningful way. In this case, defining a grammar for the formal language (OWL-S) would automate this process and allow systems to interoperate the same way services would in the semantic web.

In addition to an OWL-S description of each subsystem, we have identified three levels of system access:

- *Atomic level*: Individual systems are described at the most basic level. The description of the concepts and processes at this level does not include any other concept and process. In terms of a grammar atomic systems are the words of the language (PDUs in DIS for example)

- *Composed level*: At this level, systems describe themselves in term of a family of subsystems that are semantically linked. For instance in order to implement a "MOVE", the system tells a soldier to start moving while if it is a vehicle, a "MOUNT" has to be performed first. In this case the "MOVE" order for vehicles is executed at the composite level. This type of scenario is known in OWL-S as a precondition

- *Aggregate level*: The aggregate level can be viewed as the system of systems. The system is described at the top level, exposing only the parts that are pertinent for immediate consumption.

Defining and applying a grammar at the atomic level, enables pragmatic interoperability. Moreover, it becomes possible to dynamically compose and orchestrate subsystems into temporary or permanent families of systems.

In order to see how these two application areas complement each other in their support of M&S, let's consider the levels of data access described above – atomic, composed, and aggregate. If the Grammar and Web Service systems are allowing systems to access each other's data with ontological meaning, it becomes clear that the three levels of data access should correspond in some way to the components of a system's ontology. The atomic level of access allows for the handling of the data elements that represent the ontological entities of a system. The composed level of access allows for the handling of data elements at the classic ontological view of a system – entities and relations. Once we move into the realm of aggregate level access, we are now introducing the effects of the rules of the ontology.

Relying on this combination of application areas (1) ontological description involving concepts and rules and (2) grammars and web services has some interesting implications for our system interoperability. With conceptual understanding of entities, as well as with aggregate level access, we can be firmly grounded within the pragmatic level (level 4) of the LCIM. The foundations are laid for research into dynamic interoperability, and perhaps conceptual interoperability (levels 5 and 6); however these remain the area of future research.

SELECTED RELATED RESEARCH

The Semantic Web initiatives and the ontological spectrum are tightly connected with Internet-based applications, of which web services are of special interest, as they are enabling service-oriented architectures being of special interest for distributed simulation applications. The current developments on Runtime Infrastructure improvements are looking into this direction and early prototypes have been demonstrated. Therefore, we will focus on work on choreography and composition of web services and the ontological implications of this work.

- In their overview on current solutions for web-service composability, Srivastava and Koehler (2003) show that the functionality of web services needs to be described with additional pieces of information. They recommend "semantic annotation" of what the web service does or by a "functional annotation" of how it behaves. Current solutions based on the Resource Description Framework (RDF) are generally not sufficient.

- Lopes and Hammoudi (2003) show how the use of frameworks, as provided by the OMG's Model Driven Architecture (MDA), can support the composition of web services on higher levels.

- Concepts like the Web Service Conversation Language (WSCL) can enable services to negotiate their composition, as discussed in Banerji et al. (2002). This approach requires a semantically rich environment for orchestration as well.

- Agerval et al. (2005) recommend a framework to represent the underlying concepts in the form of a common ontology mainly focusing on strong semantic methods of the ontological spectrum.

- Arpinar et al. (2005) recommend a framework using ontological descriptions of web services to discover and compose services into higher services. While the platform uses semantic similarities, the higher levels of the LCIM are not taken under consideration by the semi-automated mapping efforts.

These approaches are just a few examples to show current research domains of immediate interest. It should be pointed out that the M&S specific challenges of conceptual modeling are not in the mainstream of ongoing discussions, but the necessity to capture underlying assumptions is getting to be discussed increasingly, not only within the M&S community.

SUMMARY

This paper can only give a first introduction to the ideas of ontology driven interoperability and its M&S applications. Nonetheless, the potential to contribute significantly to the semantic alignment of models should be obvious. However, to support all layers of interoperation defined in the LCIM extensions are needed. The objective of the Semantic Web initiatives is to change the Internet from a "web of documents" into a "web of data." M&S applications, however, require more than aligned data. In order to achieve interoperability on the simulation level, composability on the modeling level is required. This means that underlying assumptions and constraints identified in the process of modeling, i.e., building a purposeful abstraction of reality, must be captured and documented based on engineering methods. Conceptual modeling must evolve from being an art into a discipline. The methods of the ontological spectrum enhanced with methods as proposed in the application section of this paper have the potential to significantly contribute to fulfill these requirements, but additional research is necessary, in particular concerning the applicability of results in related domains such as system-of-systems dynamics, modeling languages (such as UML/SysML), and model based approaches (such as MDA).

ACKNOWLEDGEMENTS

The Joint Advanced Training Technology Laboratory (JATTL) of the US Joint Forces Command predominantly funded the underlying research work leading to the results presented in this paper. We also like to thank our colleagues from George Mason University and Naval Postgraduate School for the fruitful discussion and contributions in the course of writing this paper, in particular Mr. Curtis L. Blais.

REFERENCES

Agarwal, S., Handschuh, S., and Staab, S. (2005). Annotation, Composition and Invocation of Semantic Web Services. Journal on Web Semantics 2 (1): 1-24

Arpinar, B., Zhang, R., Aleman-Meza, B., and Maduko, A. (2005). Ontology-driven Web services composition platform. Journal on Information Systems and E-Business Management 3 (2): 175 – 199

Banerji, A., Bartolini, C., Beringer, D., Chopella, V., Govindarajan, K., Karp, A., Kuno, H., Lemon, M., Pogossiants, G., Sharma, S., and Williams., S. (2002). Web Services Conversation Language (WSCL). W3C Note Mar 2002

Berners-Lee, T., Hendler, J., and Lassila, O. (2001). The Semantic Web. Scientific American, May, 2001

Daconta, M., Obrst, L., and Smith, K. (2003). The Semantic Web: The Future of XML, Web Services, and Knowledge Management. John Wiley, Inc.

Davis, P.K., and Anderson, R.H. (2003). Improving the Composability of Department of Defense Models and Simulations. RAND Corporation

IEEE Standard Group 1278: Distributed Interactive Simulation (Revision 2002). Obtainable via http://standards.ieee.org/catalog/olis/compsim.html [last visited June 2006]

IEEE Standard Group 1516: High Level Architecture (Revision 2000), obtainable via http://standards.ieee.org/catalog/olis/compsim.html [last visited June 2006]

International Standard ISO/IEC 11179: Information technology – Metadata registries (MDR), Part I-VI

Lopes, D., and Hammoudi, S. (2003). Web Services in the Context of MDA. Proceedings International Conference on Web Services, IEEE CS Press

Mizoguchi, R., and Ikeda, M. (1996). Towards Ontology Engineering. Osaka University Technical Report AI-TR-96-1, I.S.I.R.

Morris, E., Levine, L., Meyers, C., Place, P., and Plakosh, D. (2004). System of Systems Interoperability (SOSI): Final Report. Carnegie Mellon Software Engineering Institute, Technical Report CMU/SEI-2004-TR-004 ESC-TR-2004-004

Obrst, L. (2006). The Ontological Spectrum & Semantic Models. Open, International, Virtual Community of Practice on Ontologies, http://ontolog.cim3.net (last visited June 2006)

Page, E.H., Briggs, R., and Tufarolo, J.A. (2004). Toward a Family of Maturity Models for the Simulation Interconnection Problem. Proceedings IEEE Spring Simulation Interoperability Workshop, IEEE CS Press

Srivastava, B., and Koehler, J. (2003). Web Service Composition - Current Solutions and Open Problems. Proceedings ICAPS 2003 Workshop on Planning for Web Services

Tolk, A. (2006). What comes after the Semantic Web – PADS Implications for the Dynamic Web. Proceedings of the ACM/IEEE/SCS Principles of Advanced Distributed Simulation Conference (PADS 2006), IEEE CS Press

Tolk, A., and Blais, C.L. (2005). Taxonomies, Ontologies, and Battle Management Languages – Recommendations for the Coalition BML Study Group. Proceedings IEEE Spring Simulation Interoperability Workshop, IEEE CS Press

Tolk, A., and Muguira, J.A. (2003). The Levels of Conceptual Interoperability Model (LCIM). Proceedings IEEE Fall Simulation Interoperability Workshop, IEEE CS Press

Tolk, A., and Muguira, J.A. (2004). M&S within the Model Driven Architecture. Proceedings Interservice/Industry Training, Simulation, and Education Conference (I/ITSEC)

Zeigler, B., Ball, G., Cho, H., Lee, J., Sarjoughian, H. (1999). The DEVS/HLA Distributed Simulation Environment and its Support for Predictive Filtering. In SIMULATION: Special Issue on The High Level Architecture, Volume 73, No. 4
www.wikipedia.com

