Management of C4I and M&S Data Standards with Modular OWL Ontologies

Kevin Gupton
Applied Research Laboratories
The University of Texas at Austin
10,000 Burnet Rd.
Austin, Texas 78758
kgupton@arlut.utexas.edu

Jeff Abbott
Systems Architect
CAE USA Professional Services
3501 Quadrangle BLVD
Orlando, FL, 32817 USA
407-745-2605
jeff.abbott@caemilusa.com

Curtis Blais
MOVES Institute
Naval Postgraduate School
Monterey, CA 93943
clblais@nps.edu

Dr. Saikou Y. Diallo
Virginia Modeling, Analysis & Simulation Center
1030 University Blvd
Suffolk, VA 23435
sdiallo@odu.edu

Dr. Kevin Heffner
Pegasus Simulation
PO Box 47552
Plateau Mont-Royal PS
Montreal Quebec, H2H 2S8
514-600-0141
k.heffner@pegasim.com

Chuck Turnitsa
Virginia Modeling, Analysis & Simulation Center
1030 University Blvd
Suffolk, VA 23435
cturnits@odu.edu

Keywords:
mapping, M&S, interoperability, initialization, C4I, MSDL, JC3IEDM, GFMIEDM, JTDS, OBS, ontology, Semantic Web, standards, reuse, XML, RDF, OWL

ABSTRACT: *Standard information exchange data models (IEDMs), such as the Joint Consultation Command and Control IEDM (JC3IEDM) managed by the Multilateral Interoperability Programme (MIP) and the National Information Exchange Model (NIEM) managed by the US Department of Homeland Security, often are expressed as XML Schema Definition (XSD) documents. This choice of model representation comes with the benefits of a widely adopted format and a well-supported XML toolset and libraries. Although XML, as a technology, has been an enabler in achieving model alignment and interoperability among C4I and M&S systems, several key issues have not been fully addressed. For instance, XML does not provide a standard means for representing semantics. This means that XML expressions generally cannot be interpreted by applications in a meaningful manner unless specific code has been added for this purpose.*

In addition, systems utilizing multiple IEDMs are faced with difficult mapping and model translation tasks that cannot easily be automated. Furthermore, the use of multiple IEDMs creates significant maintainability and scalability challenges associated with the use of the relevant standards and specifications. As the user-base of a data standard grows, the need for distributed management and extensibility becomes critical. Developer communities are driven to make extensions to the data models to support their domain- and system-specific requirements. This results in laborious, manual, often unsustainable extension-management processes. In particular, the MIP-JC3IEDM, SISO Military Scenario Definition Language (MSDL), and US Joint Forces Command (JFCOM) Order of Battle Service (OBS) XML specifications have growing user-bases, and while they share many concepts, no clear maintainable alignment has occurred among them. Although some efforts toward alignment across IEDMs have occurred, no standard, maintainable process or methodology for this has been proposed.

This paper describes a framework for such a model management methodology that endorses the use of the Web Ontology Language (OWL) as the choice for model representation and embraces Semantic Web best practices. This framework shows how the "mappability" of various models can be explicitly expressed using OWL. Beyond the model mapping problem, also shown is how multiple standards can be related and aligned such that they leverage and build upon each other and also provide the basis for model execution—including SISO's suite of standards and products. This is to include MSDL, the Coalition Battle Management Language (C-BML), JC3IEDM, OBS, and Distributed Interactive Simulation (DIS).

1. Introduction

In recent years, there has been a growing interest in "ontology" and the "Semantic Web" and the role they can play in standards development. Ontology is a formal specification of a conceptualization [1], providing a description of what is known about some area of study. The Semantic Web is a vision of a future "web of knowledge" where information on the World-Wide Web (WWW) is self-describing and equally accessible and understandable to both humans and software [2]. Extensive research and development are underway to realize this vision, resulting in numerous tools and techniques for describing and processing data. With respect to standards, the question arises as to what benefits can be obtained from these techniques compared to other technologies for developing, maintaining, implementing, and evolving standards specifications.

The Web Ontology Language (OWL) [3] has emerged in the last decade as a game-changing means for capturing a standardized representation of worldviews as semantic data models. In principle, OWL emphasizes:

- Semantics over syntax – integrate on concept meaning rather than format.
- Application independence – organize models so they may be used in different ways by applications based on different perspectives.
- Web-centric – follows tenets of the WWW for distributed, composable, and extensible management of data models with uniform resource locators (URLs) as globally unique identifiers.
- Standardized representation – OWL is built upon the Resource Description Framework (RDF), leveraging its homogenous, consistent representations and conventions, such that all data and models are read exactly the same way, regardless of domain or topic of those data and models.

1.1. Problem Statement

Model representation formats such as the Integration Definition for Information Modeling (IDEF1X), the Extensible Markup Language (XML)/XML Schema Definition (XSD), and the Unified Modeling Language (UML) have been used successfully in standardization efforts in the C4I and M&S domains. However, significant technical gaps persist in the development and maintenance of standards due to limitations associated with these model representations. Standards that are developed independently and/or dependent upon multiple models result in difficulties compounded by the variations in semantics—as is the case for the JC3IEDM, C-BML, and MSDL standards. Whether these limitations originate in IDEF1X, XML, and UML themselves, in the tools that exist for manipulating models and schemas, or in the conventions used when employing those tools, the gaps that will be highlighted below have impeded progress in standards alignment, maintenance, and implementation.

For example, the prolific use of text, PDF, spreadsheets, graphics, and other human-consumable specification materials broadens the gap between specifications and software implementations that must conform to them. Some rules and constraints may be captured in executable form, but much information is left in unstructured, free text form. This not only prevents machines from leveraging those guidelines, but also allows ambiguity to persist and prevents strong linkage and traceability among those human-readable products.

The Object Management Group (OMG) Model-Driven Architecture (MDA) initiative defines a formal approach for managing models in a comprehensive manner from model definition to the generation of the required code to ensure interoperability across applications. Based on UML, the MDA approach has been experimented with by the MIP for use with the JC3IEDM, forming the basis for the Shared Operational Picture Exchange Services (SOPES) standard. However, MDA generally requires an additional language such as the Object Constraint Language (OCL) that can express constraints and can capture business rules that can be converted readily into code. Even OCL, though, is characterized by usability and language maturity issues concerning, amongst other issues, its semantics [4].

1.2. OWL-based Model Representations

This paper describes an alternative methodology for managing C4I and M&S domain models that simplifies the use and maintenance of multiple products. We will illustrate how OWL and other semantic technologies and conventions provide a resolution to the issues identified and can provide the basis for an approach and a way forward for future standards development.

We will make clear that OWL is not a panacea to all of our problems, but it provides improved capabilities for unambiguous expression of the models, standards, and data intended to be shared, integrated, and reused.

2. Background – Data Interoperability and Reuse

Before considering the application of OWL to the problem at hand, we must understand why semantics are required. What role does semantics play in integration and interoperability? How is it that many of our problems are in fact shortcomings in semantic expression and alignment?

2.1. Where We Are Today

The first wave of DoD Net-Centric Data Strategy (NCDS) implementation emphasized a more service-minded approach to system architecture design. Programs intent on realizing the NCDS built new systems using web services or adapted existing systems to make use of web service technologies. The same enabling technologies prevalent in corporate enterprise and eCommerce have begun to permeate DoD architectures. XML data messages and SOAP-based[1] and RESTful[2] web services are taken as *de facto* standards when deploying service-oriented solutions.

Numerous programs, processes, and systems have migrated to some level of net-centricity. However, many of these successes have been in point-to-point solutions among organizations with pre-existing relationships. The same fixed processes, data exchange channels, and pipelines that existed before have been replicated and modernized using web services.

The reality is that stove-piped legacy solutions often have led to the development and reliance on corresponding stove-piped services. Web services have been built in a deliberate way, focusing technical implementations exclusively on the requirements of specific programs. Limited support for external or third party users often creates difficulties in re-using these point-to-point solutions that were constructed with little or no provision for supporting third party users.

2.2. Supporting the Unanticipated User

Described in the NCDS, the "unanticipated user" is an elusive concept to many implementers. How is it possible to design for the needs of an unanticipated service consumer? It can be argued that the first step is to move from control-centric design that is common in legacy enterprise solutions to a content-centric service design that has contributed to making the World-Wide Web successful. A content-centric interface eases the task of application integration by focusing on the uniform nature of content, rather than the specific controls of any given application [5]. This requires a change in design perspective from the natural tendency to build functionality directly from software requirements to identifying/specifying the data that is to be exchanged and managed. Content-centric design dictates that data should be "exposed" in a uniform (and, as necessary, secure) way to support the consumer requirements, but also supports consumers who have evolving requirements. By simply adopting a content-centric perspective for service design, the resulting solutions enable more consumers to use the products and lower the cost of modifying those services as unanticipated requirements become supported requirements.

The first step to empowering the "unanticipated user" starts with system design; the second step is to address the data representation itself. Even with services and message format standards, conventions, and best practices, we still live in a digital "Tower of Babel". Transparent machine-to-machine communication involving data exchange can be achieved, yet still be characterized by problems concerning how to make data reusable and how to avoid misuse or misinterpretation of data.

In [6] it was shown how different organizations that make use of a common representation of data (same data language, same tags, same objects) can have their own interpretation of the *meaning* (semantics) of those tags. For a business enterprise, meaning comes from the business rules and goals that govern that business. For a military organization, that meaning comes from doctrine. And so on, for other types of organizations. Each has a different culture that lends specific meaning to what may seem on the surface to be common terms or tags. The use of an ontology helps to disambiguate that conflation of meaning by expressing particulars about what a system either expects or interprets from specific data elements.

2.3. A Technical Failure to Convey Meaning

Each of the multitude of information exchange models and data exchange schemas—including hundreds of XML schemas, SQL databases, and IDEF1X models—serves a purpose, be it narrow or broad. The potential for creating specialized XML schemas for each type of

[1] SOAP: Simple Object Access Protocol

[2] REST: Representational State Transfer

data exchange is not necessarily a bad thing, but with no means of aligning the semantics of the payloads conforming to those schemas, extensive divergence of formats and meaning of the data being exchanged often occurs.

Information exchange models are intended for just that: the exchange of information, oriented to what can and cannot be exchanged and for checking the validity of data that has been exchanged. A shared representation is needed for data exchange that is independent of any particular information exchange format or exchange protocol. This is, in fact, the definition of ontology: a formal representation of knowledge as a set of concepts within a domain, and the relationships between those concepts [7]. For the purposes of this discussion, an ontology is the conceptual model from which information exchange formats are derived. An ontology is independent of how some process or system might store, exchange, or manipulate the data, but provides the top-level, universal frame of reference by which data exchange can occur with a common understanding of data meaning and context.

By modeling a domain, ontologies can unify the semantics of dialects of message formats. For example, the Joint Common Database (JCDB), the JC3IEDM data model (which forms the basis for the Coalition Battle Management Language (C-BML)), the Military Scenario Definition Language (MSDL), and Joint Training Data Services (JTDS) Order of Battle Service (OBS) each provide frames for storing or exchanging order of battle (OOB) and military scenario data using database or XML schemas, though each has distinct intended application. Lacking a clear understanding of individual data models, human development of mappings for translating data across these formats can be highly subjective. Automated derivation of such mappings is still an open research area. However, if these specialized formats and models were derived from a common model of command and control (C2) concepts and relationships—i.e., a C2 ontology—it would be easier to understand how data can be translated unambiguously from one model to another. Converting data from MSDL to JC3IEDM to OBS XML would be trivial, if it was known how those formats related to a common C2 ontology. C2 Core, an initiative from the United States Joint Forces Command J8 Directive, provides a common C2 "world view" from which exchange formats can be derived for C2, logistics, M&S, and other communities [8]. C2 Core, and its foundation, the Universal Core (UCore), provide metadata describing an information exchange payload, but do not presume to define the specific structure and content of that payload. Many systems would be able to understand the gist of the payload from the C2 Core/UCore descriptions (digest), but the problem of understanding the semantics of the embedded content remains.

3. Problems and Challenges in Data Standards Management

Many obstacles exist in the standardization of data formats for any domain, and standardizing tactical message exchanges and military scenario descriptions is no exception. The following is a survey of some of the design and maintenance issues that often have been observed by the authors in data standards drafting and system interoperability projects.

Breakdown in Understanding	
Unable to reuse standards across domains	Domains and COIs often reuse models and vocabularies, but with vastly different connotations.
Granularity: Levels of concern	Between the creation of a mission plan and the initialization of C4I/M&S systems, the "scenario" and all supporting data artifacts are created incrementally in many value-adding pipelines. Each incremental step requires finer details than the preceding step.
System-independent vs. system-specific concepts	C4I, M&S, and data management systems often require specialized data views, formats, and syntax for business logic execution.

Schema management	
Extension	Domain, organizational, or system extensions are inevitable. Managing those extensions should be well defined yet simple.
Model mapping & data lossiness	Mappings for translating/mediating between models often are specified in human-only PDFs or spreadsheets. Software mediation is hardcoded, making support of new schemas, versions, or mappings expensive, unsustainable, or costly.
Validation rules and data constraints	Every stage of a data development "pipeline" has its own usage rules and constraints regarding optional/required, accuracy, or granularity, for example.
Pedigree and traceability metadata	The pedigree, revision, and change log of data should be maintained and propagated for consumers to evaluate the validity and appropriateness of the dataset.

Figure 1 depicts the current situation with C-BML, MSDL, OBS, JC3IEDM, and the Global Force Management Information Exchange Data Model (GFMIEDM). The MSDL and OBS XML schemas have weak semantics, where traceability to authoritative definitions is only available through comments in accompanying documents. The JC3IEDM has the strongest semantics of all the aforementioned, but lacks clear guidance on how to "link" to semantics or manage extensions. The GFMIEDM and C-BML schemas use the JC3IEDM directly, but the tools and formats they use do not lend themselves to sustainable standards specifications.

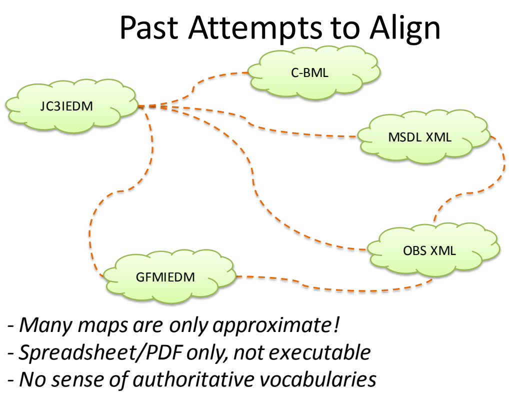

Figure 1 - Lack of Model Alignment

In systems integration, it often can be heard that "Model X is mappable to the JC3IEDM", where "Model X" can be MSDL, OBS, or some other data model. That statement is often only true in theory. In practice, we find that either only a small portion of a model maps, sometimes only the semantics map, while syntax (field format) does not, or the model definitions are too ambiguous to yield confident mappings. Analyzing the degree to which models are "mappable" to each other is pointless if the meaning of "mappability" is unclear. In the authors' opinion, thus far, past evaluations of "mappability" have not improved interoperability significantly.

3.1. Semantics Frozen in Free Text

Machine formats for specifying data models and specifications were identified above; but a far greater concern exists in the free text specification documents. For many government and standards organization products, the most basic and essential semantics needed for consistency and traceability checking only exist in human-readable documents. Glossaries, Terms of Reference (TOR), categorization schemes, and many business rules are only available in human-readable document form, all while many other documents, manuals, and specifications depend on them explicitly. The business logic of our communities often is built upon limited textual foundation documents and referential approximations. Systems are expected to adopt and conform to the guidance documents, yet those logical gaps remain. Instead, we rely on humans in the loop to verify software, models, schemas, and to align standards. This human intervention, of course, is error prone and unnecessarily subjective, as the definitions among coalition partners, standards organizations, academia, and industry may differ greatly and/or diverge due to a lack of coordination and traceability.

While better practices and good design could mitigate some of today's interoperability limitations, the *de facto* structure and design of the RDF syntax and OWL semantic modeling schema provide elegant solutions natively—features inherent to their Web-oriented, semantics-driven design. We illustrate this in the following sections describing the proposed technical approach.

4. Concept for OWL-centric Standards Management and Alignment

OWL and the Semantic Web directly address many of the limitations of today's net-centric data solutions. A trend has been recognized across academia, industry, commerce, government, and DoD that fundamental problems causing communication breakdowns are not being addressed resulting in interoperability failures across our diverse communities.

The approach advocated in this paper can be broken down into three areas: (1) the use of OWL as a common model representation; (2) the use of an upper ontology to ensure minimum consistency of a maximum number of information elements; and (3) the coordinated extension of models toward the goal of establishing lower ontologies from which all identified models can be derived.

For sustainable, successful M&S data standards management, more is required than just infrastructure and governance. We must change the manner in which our information is organized. We must embrace best practices that *promote* convergence and alignment, but without unduly *forcing* convergence.

4.1. Model-based Data Engineering (MBDE)

Similar to the SOPES OMG-MDA approach, the Model-Based Data Engineering (MBDE) process is certainly not a new concept to SISO standards development [9]. As previously stated, great progress in interoperability and reuse has been achieved by converging models and deriving schemas from shared reference models. However, those solutions are based on the use of technologies such as IDEF1X tools, XSDs, SQL, and spreadsheets that complicate execution-time interoperability. While OWL is not a "silver bullet," it provides elegant facilities for specifying and linking standards that are not provided by legacy methodologies, and supports generation of executable solutions for data interchange.

4.2. Proposal for Data Standards Management Framework

This paper advocates the alignment of MSDL, C-BML, GFMIEDM, and JTDS OBS XML schemas with a common extensible C2 and M&S ontology using OWL semantic technologies. This ontology is heavily influenced by the JC3IEDM. Using the benefits inherent in the OWL specification, current misalignments associated with these formats and models can be organized into a framework of modular ontology "layers" where semantic alignment is maintained from the core outward to the end-users' data exchange formats. Such a framework will create a "separation of concern" in model design and support granular composability of standardized and customized ontology modules.

4.3. Why use OWL instead of XML

Information and models in OWL can be expressed directly in XML. However, the additional constraints imposed by OWL allow the semantics of a technical specification to be propagated through a system's design and allow coding to be simplified. The advantage of using OWL over generic XML is that OWL captures and propagates the precise definitions (semantics) of data classes, properties, and instance values. The JC3IEDM includes these data element definitions. Although OBS, MSDL, and other generic XML schemas may include some data element semantics ("annotations") in the schemas themselves or in other related documents, these semantics are not as clear, precise, or complete as the MIP's JC3IEDM.

Moving to an OWL-centric standards framework does not change the complexity of the broader interoperability problem, but it does reduce complexity of implementing solutions. OWL models may be written in the RDF-XML format, meaning it is no more powerful than XML in general. Nevertheless, RDF and OWL embody implementation best practices that make using XML as a data format more precise and consistent and less ambiguous to interpretation.

OWL also enables a modular, composable, distributed approach to modeling, allowing the model space to be partitioned into loosely coupled ontology modules. For example, MSDL then becomes a scenario-specific overlay to a JC3IEDM core. OBS defines a federation and system overlay to MSDL. Common tasking modules also can be extracted from C-BML to support non-military applications such as crisis management, disaster response, and civilian logistics, in a way similar to the SOPES effort.

As data models are reorganized into composed OWL modules, a layering of components will emerge. Application-specific specifications such as MSDL and OBS will build upon semantic cores like the JC3IEDM, GFMIEDM, Universal Core (UCore), or C2 Core. As new community- or system-specific extensions are designed, changes to the core data models are not required, but rather lead to building new semantic layers on top of the core.

Even though OWL provides formats for data storage and data exchange, the development and use of data exchange formats such as MSDL, OBS XML, and GFMIEDM XML will continue. Note that the approach proposed in this paper does not suggest that SQL databases or XML schemas be eliminated or phased out. However, those storage and exchange schemas should be derived from the standard ontologies. Ideally, the alignment and consistency of derivative schemas to standard ontologies should be verified programmatically. MSDL, OBS, and other schemas will need review and adjustment as the standard, consistent C2 and M&S ontologies emerge.

4.4. Leveraging Semantic Technologies

Semantic technologies can benefit more than just data modeling and data format standardization. "Reference data" is also standardized and published for community use. Publishing these standard datasets using OWL also has advantages. For example, DIS enumerations, standard characteristics data, and "known entities" (e.g. ships, people, religions, cultures, factions, organizations, countries, streets, cities, regions, etc.) can all be published as machine-interpretable OWL datasets. The same is true for the wealth of knowledge available from doctrine, policy, and procedures (e.g., NATO Standardized Agreements or STANAGs, Field Manuals, Training Manuals, etc.) but are now only available as PDF documents or Web pages. Encoding in OWL the concepts and relationships from such sources can provide a common format and foundational semantics to simplify system integration.

Finally, the runtime translation and mediation of system data can be simplified and made dynamic with a catalog of modular ontologies and machine-readable mappings to interchange and storage formats. Aligning semantics across our communities is the greater challenge, so when understanding is achieved and documented, translating among syntactic formats will become much more straightforward and more amenable to automation.

5. Framework for Model Management

Before illustrating the inherent benefits of OWL to our alignment and management challenges, we begin by showing how OWL can be used in a standards management framework. What follows is a repeatable process for integrating XML, SQL, text documents, or other format models at the semantic level using OWL.

5.1. Step 1: Migrating semantics out of PDF, XML, and SQL into OWL

In previous sections, we described how syntax, semantics, and context are intermixed in the use of XML or SQL schemas. The effect is a poor "separation of concern" and "mappings" that are inconsistent and not machine-readable or, worse, not machine-executable.

Recently, industry has been successful in adopting a new, more elegant approach to the alignment of distributed structured data. Any model to be integrated—be it XSD, SQL, etc.—should have a corresponding OWL model representation. Any necessary mappings or processes for translating data from the specialized model into OWL should be documented as well. Reference [10] addresses the problem of aligning geospatial feature data dictionaries; the methods applied there are applicable for integrating data models using OWL.

As ambiguity (both accidental and intended) is revealed, difficult, time-consuming questions about intent will need to be answered. For example, many XML schemas fail to specify what relationship exists between nested elements, or whether the order of elements carries meaning. IDEF1X models often neglect to declare entity relationships as association, aggregation, or composition. At this stage, possible oversights and defects in the native model need to be submitted to their respective development groups for consideration; the same is true in later steps of this framework.

The output from this process must be sufficient for completely lossless bidirectional translation of data between the native format and the OWL model. For the problem at hand, this means creation of OWL ontologies that capture the semantics of MSDL, C-BML, OBS, JC3IEDM, and GFMIEDM, respectively. This correspondence map is critical to data mediation into and out of the native format [8] . However, if a full corresponding OWL model is not created, then those compromises and gaps must be documented.

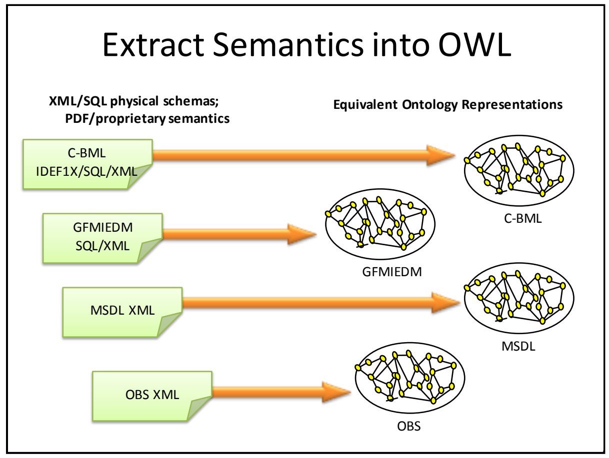

Figure 2 –Extracting semantics into OWL

In this first step depicted in figure 2, the OWL representation does not necessarily have to consider or include other existing OWL models. The intent is to capture the intended semantics and usage guidelines as well as possible in OWL. The next step below requires the deconfliction, normalization, and reuse of existing OWL models.

5.2. Step 2: Constructing modular composed ontologies

Once each model is properly represented in OWL, we can then compare, align, and deconflict them against accepted core ontologies. For the international C2 domain, the JC3IEDM provides the currently accepted semantic core. For instance, the U.S. Army created the draft "C4ISR Data Ontology", which used an earlier Command and Control IEDM (C2IEDM) ontology as a core and added message (variable message format, message text format, tactical data link) and Community of Interest (time-sensitive targeting, battle command) specific extension modules [11].

The JC3IEDM, C-BML, GFMIEDM, MSDL, and OBS models all have similar or even identical concepts, which of course is why we attempt to map them. However, instead of simply mapping the concepts between any two models, thus only creating "umbilical" relationships, OWL allows us to share concepts properly by directly importing other ontologies.

For example, all five models have the concept of a military unit, which—according to the developers of those models—are the exact same concept. However, each has different definitions and syntax for a unit, and different syntax and meaning for a unit name, formal abbreviated name, echelon, etc. By adopting the authoritative, validated definitions embodied in the JC3IEDM, we can directly reference those "normative" semantics within the GFMIEDM, C-BML, MSDL, and OBS vocabularies. Figure 3 presents the modular, composable ontology solution.

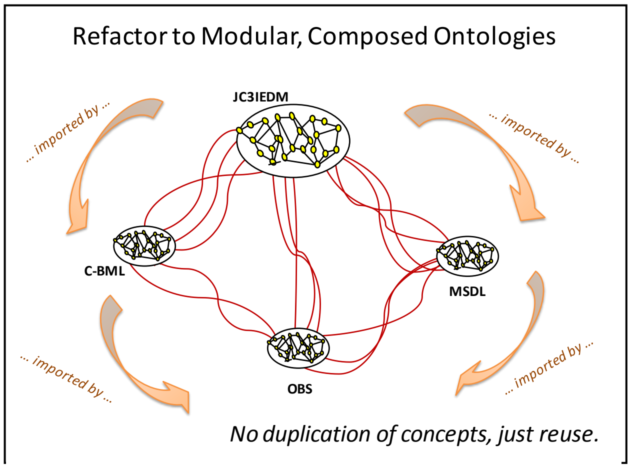

Figure 3 – Refactoring Modular Ontologies

The Unit Order of Battle (UOB) data is common to all four of these models—the military organizations, relationships, and entities (vehicles, aircraft, and personnel) assigned to organizations. Tasks, plans, and orders are part of JC3IEDM and C-BML, and eventually MSDL. Geospatial position/location of units, platforms, and other features is common across several models. All of these shared classes, attributes, and relationships should be unified and imported by all models that use those concepts.

We must also recognize that much of what we model is not unique to our community, thus many robust reference ontologies already exist in the Semantic Web. For example, ontologies exist for representing people, address books, network topology, geospatial locations and reference systems, and action/task/events. As we converge and align our models, we must evaluate and utilize the work done in other technical communities.

From aligning our models at the semantic level in the previous step:

- The GFMIEDM ontology will import the JC3IEDM and needs only to define additional ontology elements for US-specific force management concepts.
- The MSDL ontology will import the JC3IEDM and C-BML ontologies and only extend to include simulation-specific scenario concepts not already included in the JC3IEDM or C-BML.
- The OBS ontology will import JC3IEDM (for C2 concepts), MSDL (for M&S concepts), GFMIEDM (for US force management concepts), and additional US Army ontologies for FBCB2/BFT network elements. The OBS ontology then will only need to define a few system-specific concepts needed (e.g., for initializing JCATS, OneSAF, JDLM, TACSIM, SIMPLE, etc.) instead of a large duplicative model.
- Any other domain could also import and extend any of these models as needed, and define extension ontologies—all without duplicating and maintaining everything already in JC3IEDM.
- The concepts in each model no longer approximately match; they are literally using the same concept definitions.

5.3. Step 3: Deriving schemas from shared reference ontologies

In the previous steps, we describe how disparate XML, SQL, and conceptual models can be interlinked to directly share entity, relationship, and attribute definitions. Ontologies define the "world" described by languages and data models. They represent the foundation over which information is communicated. Through OWL, ontologies can be utilized directly for data storage, business logic, and data exchange in RDF format. However, there are advantages to separating the ontology from the data model. One principle advantage is the flexibility in selecting and using a variety of data representation formats (XML, SQL, RDF, etc.) for the same ontology. Semantics enable the proper use of representation formats in specific contexts without affecting the logical data model.

In contrast, many of today's data models include implicit and explicit logical semantics that cannot be separated from their physical representations. To correct for this, the following procedure can be applied:

- Reverse engineer the semantics from the physical model.

- Establish translation layers that derive/generate physical implementations from the logical semantics of a common ontology.
- Evolve the data standardization processes to align with this paradigm of specifying the logical semantics of an ontology before deciding how to represent the model physically.

Two premises are at play in this step:

1) All physical models (e.g., XML schemas, SQL schemas) should be directly derivable from a conceptual semantic model. The MSDL and OBS XML schemas are not based on conceptual models today.

2) XML schemas are specific to the business process they support. They are intentionally constrained and often not intended for unambiguous data exchange in other business processes.

The US Navy XML Naming and Design Rules (NDRs) and the Universal Business Language (UBL) both build on these premises. The Navy XML NDRs even provide examples of precisely deriving specialized XML message schemas from a generic conceptual model. The National Information Exchange Model (NIEM) also provides guidance for information exchange messages, which also could benefit from being derived from reference domain ontologies (http://www.niem.gov/grants.php).

Working backwards from Step 1 described above, we can obtain semantically aligned XML schemas and XML-to/from-OWL executable mappings. In the case of MSDL and OBS, future versions of each should begin with adjustments to their respective ontologies, which then drive (automatic) changes to their native XML schemas.

As any XML schema becomes aligned with a common reference ontology, some deviations will arise. Many of these deviations are unintended barriers to alignment, perhaps defects that thus far have not been identified by the drafting groups and previous alignment efforts. As mapping efforts proceed, those defects should be reported as proposed changes to respective development groups.

Although the coordination of such change requests across standardization bodies may require significant effort, return on investment will be substantial. Also, a natural evolution of standards can be expected as long as a set of ground rules are followed (see section 6.5).

5.4. Step 4: Lossless translation of data across semantic bridges

Constructing semantically aligned SQL or XML schemas as a result of following this methodology represents great progress and will allow the systems to move toward better data sharing while utilizing the same technologies that are in use today. Current processes of manually mapping any two schemas and hard coding mediation will be more successful with good semantic alignment in the schemas.

However, we can go a step further and actually utilize the modular ontologies as part of the data mediation process. Instead of hard-coding the manual mappings between models (feasible, but unsustainable), we can use the ontologies during the live execution of data mediation services.

Figure 4 conceptualizes how this service might work by illustrating an example. For instance, starting with a GFMIEDM dataset (obtained either in file form or via a web service), both MSDL and OBS XML scenario files are generated.

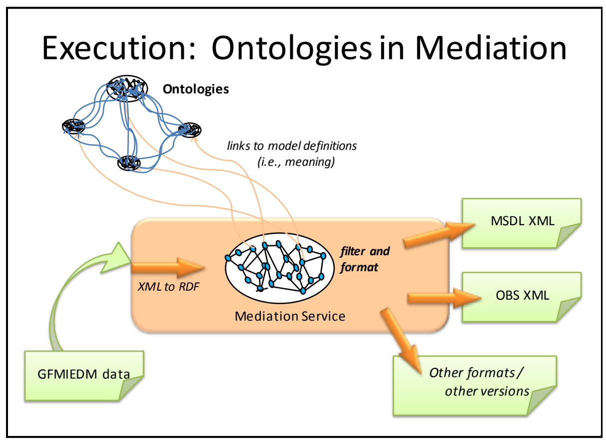

Figure 4 –Execution: Ontologies in Mediation

The process might proceed as follows:

1) Obtain GFMIEDM dataset in XML format.

2) Transform dataset into RDF form using XML-to-OWL derivation mappings.

3) Generate an MSDL XML scenario file from the RDF dataset using MSDL's XML-to-OWL derivation mappings. Unsupported data

fields will be ignored (a report of data loss can then be generated).

4) Generate an OBS XML scenario file from the RDF dataset using OBS's XML-to-OWL derivation mappings. Unsupported data fields will be ignored (a report of data loss can then be generated).

Aggregation or fusion of information is also greatly simplified. Illustrated in figure 5, we can combine two input datasets for consumption by both MSDL and OBS formats. Here we depict the "union" of a GFMIEDM dataset and an OBS XML file to create a combined scenario. The combined dataset can then be exported as an OBS XML file and an MSDL XML file.

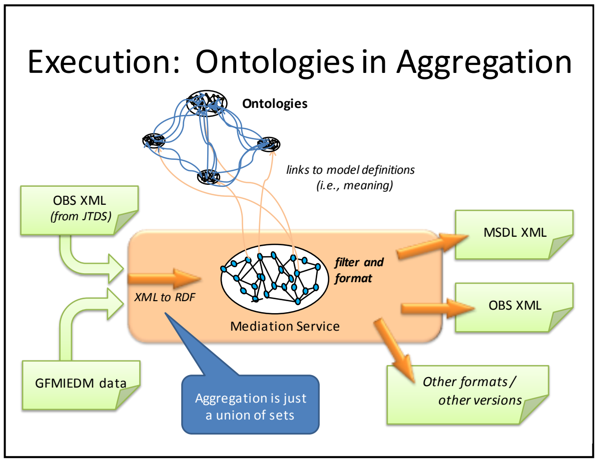

Figure 5 – Execution: Ontologies in Aggregation

6. Benefits Inherent to OWL-Centric Standards

Some of the benefits of an OWL-centric approach to standards management have been explained in the previous sections. The following sections provide a summary of these benefits and discuss how they begin to resolve the standards management challenges identified earlier in this paper.

6.1. Extensibility of Data Models

As an example, OneSAF has made its version of MSDL extensible through the xs:any W3C schema construct.

```
<xs:any namespace="##other"
      minOccurs="0"
      maxOccurs="unbounded"/>
```

This construct not only enables extensions, but also makes them reusable across schemas in cases where the respective systems share understanding of the structure and meaning of the extended content. However, their reuse is currently hindered by the lack of a common underlying semantic model (JC3IEDM). Some also argue that <xs:any/> is too permissive and does not introduce any semantics into the conceptualization. That is, <xs:any/> allows inclusion of new content, but does not imply or support the semantic meaning of that new content.

Extensibility is simplified when the desired extensions already have semantics specified in the corresponding ontology. Models simply need to derive their physical representations from the logical representations, meaning the extensions are already defined in the ontology, but not yet physically represented in the data model. For example, OneSAF has been working on MSDL extensions for the operational environment related to irregular warfare (IW). The extensibility provided by xs:any enables OneSAF to propose and evaluate adjustments to the MSDL specification. Those logical extensions were reverse engineered from existing models already in use in the Joint Nonkinetic Effects Model (JNEM). If an ontology had already been defined for IW, there would have been no need to reverse engineer the ontology.

6.2. Composable and Reusable Data Models

OWL supports the componentization of data models and the reuse of data model components across the Semantic Web through composition. For standards development, this means standards can be built as components, they can import other standards as components, and implementers can import or link to standards within their designs and applications.

This is not a new principle to software engineering. Object-oriented design has long promoted modularity and separation of concern in software design. The various products and documents of standards bodies can also be linked and deconflicted in a similar manner if expressed rigorously in OWL. Doing so will provide a sustainable means of incrementally removing unnecessary duplication and ambiguity across a body of standards products.

6.3. Distributed Management of Data Models

OWL is web-oriented by design; the Semantic Web enables data model components to be distributed across the Internet at different locations. Data models can be managed and maintained by multiple communities without having to co-locate the components. Where componentization enabled separation of concern, the distributed feature of OWL enables separation of responsibility. One community can import the data models of another community by simply linking URLs.

More than just standards bodies benefit from distributed standards management. The implementers, users, evaluators, and contributors of standards can import standard components into developmental models in the same manner. Draft and trial-use models can leverage standards seamlessly.

6.4. Executable Models

Much of this paper has emphasized how OWL is more than a mere data format. OWL is a modeling language rooted in set theory and description logics. OWL can be used for capturing a standardized model and expressing many of the business rules, constraints, and production rules with logical precision. OWL models then can be used directly in source code or source code generation. Models themselves can be validated, checked for consistency and contradictions. Validation can be performed against the "import closure" across all components that a model imports recursively. The programmatic use—or "execution"—of both datasets and data models is enabled by OWL's native representation. However, it is important to note that some of that expressivity brings concerns about computational complexity.

6.5. Natural Evolution of Standards

Distributed management must consider evolution of data models over time. Loosely coupled extensions enable different implementations to compete through trial use. A successful data management approach will have to deal with distributed adaptation and evolution of data models, meaning there will be no centralized control over the direction evolution takes the model. In addition, distributed management cannot continually change data specifications in the model. Additions are good; changes are bad. Therefore, before an extension is integrated it must be stable and accomplish its intended use very well. Preference should be given to building simple constructs first, and ensuring that they are well done. Then build on existing constructs, without changing the earlier ones.

Four distinct factors of distributed management supply ontological models with their character:

- Openness – The absence of imposed centralized control over the ontology.
- Autonomy – The autonomous nature of an ontology's constituent components or subunits.
- Interconnectivity – The high semantic connectivity between the constituent subunits (potential for interoperability).
- Evolution – The nonlinear causality of peers influencing peers to evolve the ontology.

The relative strengths and dominance of each factor are examined from the perspectives of (1) composability/reuse of data models and (2) executable architectures.

Standards developers and users then can shift their focus of adaptation over time from one part of the problem to another; from the ontology to the semantics, to the physical representations of a data model.

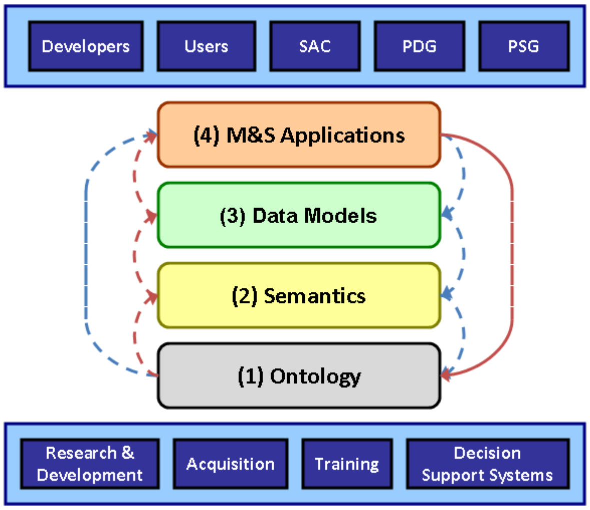

Figure 6

Figure 6 depicts standards development on top (SISO and their stakeholders), with the functional M&S applications of SISO standards on the bottom. The middle layers depict cycles of planning and execution. Planning (blue) is performed by SISO and informed by stakeholders of the M&S applications. Execution (red) is performed by the stakeholders of the M&S applications, and informed by M&S stakeholders on how to apply the standards.

These two groups "inform" each other primarily where the red and blue arrows converge. At these points, decisions of one group affect the actions of the other. Planning informs execution where the arrows converge on the Ontology layer. This represents the fielding of new standards. Execution informs planning where the arrows converge on the M&S Applications layer. This represents gaps that exist in the application of M&S standards. This executable architecture does not properly reflect the formal state of standards today. The layers of Semantics and Ontology are implicit at best.

SISO standards are effective; however, they are also frequently "competed". The formalization of the Semantics and Ontology layers would enable SISO standards to evolve more rapidly in order to adapt to today's emerging needs.

7. A Way-Forward for MSDL and C-BML Coordination

As discussions are revived on how C-BML and MSDL will coordinate and complement each other, the framework described in this paper can at least inform the semantic alignment that is necessary for those standards activities to synchronize. Additional work is also necessary:

- Leverage reference C2 ontology development: Multiple C2 ontology development activities are on-going in the US and internationally. The appropriate SISO PDGs should coordinate with those activities early to ensure those products are usable with C-BML and MSDL.
- Refine MSDL model semantics: As mentioned previously, MSDL lacks traceability to authoritative references for its elements and structure. To have any confidence that MSDL data producers and consumers interpret data content consistently and correctly, a semantic foundation must exist beneath the MSDL syntax.
- MSDL and C-BML artifacts must derive from a common C2 ontology: MSDL and C-BML should do more than cite C2 references; they should import common C2 ontologies and build upon the classes and relationships in them.
- Harmonize MSDL and C-BML scope and architectures: The roles and responsibilities of each product should be explicit such that MSDL and C-BML build upon each other, maximizing model reuse and making composed use of those products simple.
- Define a combined standards development and support plan: Any two products, once aligned, will begin to diverge and conflict if proper maintenance plans are not in place.
- Go back and generate formal ontological representations of the foundation documents as needed; PDG terms of reference, government glossaries and integrated dictionaries (e.g., DODAF AV-2 [12]), and public references should be bootstrapped into ontologies if those communities have not already done so; only then can other new standards begin to build upon old ones.

8. Conclusions and Recommendations

No single standards development group can realize the transition to a new standards development framework such as the one presented in this paper without buy-in from other similar groups. Just as there are levels of (1) physical models, (2) common semantics, and (3) ontology, the standards organization must layer their efforts to align with the layers of the problem. SAC could specify the ontology of SISO. PDGs specify the physical representation of a data model. SAC and PDGs should share in the specification of the semantics that tie PDG models to the ontology of SISO.

Standards roadmaps under development in the SAC document a structure of interdependency and relationship among SISO standards. This could be a start at specifying the ontological scope of SISO. The SAC has strongly recommended that MSDL and C-BML interact/integrate rather than compete. The same is advisable for other standards where relationships can be made explicit, executable, and verifiable.

This framework goes beyond the immediate alignment of JC3IEDM, MSDL, C-BML, OBS, and GFMIEDM. Anyone who uses these specifications can apply the same framework of ontologies to build common extensions for their communities' tangential concerns (e.g., logistics, communications) or custom extensions for their system-specific requirements (e.g., JLCCTC ERF, MATREX, OneSAF). Semantic layers may also arise to partition restricted, proprietary, or classified extensions from public, unrestricted models, as users' requirements dictate.

9. References

[1] T. R. Gruber. A translation approach to portable ontologies. *Knowledge Acquisition*, 5(2):199-220, 1993.

[2] Tim Berners-Lee, James Hendler, and Ora Lassila. The Semantic Web. Scientific American. May 17, 2001.

[3] *OWL 2 Web Ontology Language Structural Specification and Functional-Style Syntax* Boris Motik, Peter F. Patel-Schneider, Bijan Parsia, eds. W3C Recommendation, 27 October 2009, http://www.w3.org/TR/2009/REC-owl2-syntax-20091027/. Latest recommendation available at http://www.w3.org/TR/owl-syntax.

[4] R. Duarte, J. Junior, A. Mota. “Precise Modeling with UML: Why OCL?” *Submitted to the Workshop of Formal Methods, 2003.*

[5] Roy T. Fielding. “JSR 170 Overview: Standardizing the Content Repository Interface.” *Day Software.* March 13, 2005. Day Software AG. http://www.day.com/dam/day/whitepapers/JSR_170_White_Paper.pdf (accessed January 12, 2010).

[6] C. Turnitsa and A. Tolk, Battle Management Language: A Triangle with Five Sides, *Proceedings of the 2006 Spring Simulation Interoperability Workshop*, Huntsville, AL, April 2006.

[7] Wikipedia contributors, "Ontology," *Wikipedia, The Free Encyclopedia*, http://en.wikipedia.org/wiki/Ontology (accessed January 12, 2010).

[8] Basil Krikeles, Fotis Barlos, Daniel Hunter. Inference-based Semantic Mediation and Enrichment for the Social Semantic Web , *Proceedings of the AAAI 2009 Spring Symposium on Social Semantic Web: Where Web 2.0 meets Web 3.0*, 2009.

[9] Kevin Heffner, Fawzi Hassaine. “Using BML for Command & Control of Autonomous Unmanned Air Systems.” *Proceedings of the 2007 Fall Simulation Interoperability Workshop*, 07F-SIW-054, Simulation Interoperability Standards Organization, 2007.

[10] Kevin Gupton, Eric Allcorn, Roy Scrudder, Bruce Carlton. “Mapping Data Models and Data Dictionaries—Removing the Ambiguity.” *Proceedings of the 2010 Fall Simulation Interoperability Workshop*, 10F-SIW-068, Simulation Interoperability Standards Organization, 2007.

[11] *C4ISR Data Ontology Supports Army Net-Centric Data Strategy*. http://data.army.mil/documents/Ontology%20Fact%20Sheet%20-%202sided-%20ANCDS%20Slip%20Sheet.pdf (accessed January 10, 2011).

[12] DoD Architecture Framework, v 1.5 Volume 1: Definitions and Guidelines. http://cio-nii.defense.gov/docs/dodaf_volume_i.pdf

Disclaimer

The views presented in this paper are those of the authors and do not necessarily represent the views of the Department of Defense or its components.

Author Biographies

KEVIN GUPTON is an Engineering Scientist in the Modeling and Simulation Information Management Group at the University of Texas at Austin, Applied Research Laboratories (ARL:UT). He has over 9 years of experience in enterprise system engineering, data modeling, knowledge management, and scientific software engineering. Mr. Gupton has developed net-centric applications and data services utilizing C4I and M&S common data standards and ontologies. He holds a Bachelor of Science in Mathematics and a Master of Science in Computer Science from Texas A&M University.

JEFF B. ABBOTT holds a master’s degree in engineering from the University of Central Florida. Mr. Abbott has worked in the area of modeling and simulation for 22 years. Much of that time has been spent working issues of interoperability related to simulation-based training. Mr. Abbott has been working standards development under SISO in support of interoperability between battle command and modeling and simulation technologies.

CURTIS BLAIS is a Research Associate in the Naval Postgraduate School’s Modeling, Virtual Environments, and Simulation (MOVES) Institute. His principal research activities include application of Semantic Web technologies to improve system interoperability and development of new approaches for human social culture behavior (HSCB) modeling. Mr. Blais has over 36 years’ experience in M&S management, development, education, and application. He holds Bachelor of Science and Master of Science degrees in Mathematics from the University of Notre Dame.

SAIKOU Y. DIALLO is a Research Scientist with the Virginia Modeling, Analysis and Simulation Center –

VMASC – at Old Dominion University. He received his M.S and Ph.D. in Modeling and Simulation from Old Dominion University.

DR. KEVIN HEFFNER holds a BS in Engineering from the State University of NY at Buffalo and a Ph.D from University of Paris VI. He has worked in the field of modeling and simulation for over 20 years. His work includes applying model-centric concepts to flight simulator architectures and interoperability among C2, simulation, and automated forces, including Unmanned Aircraft Systems.

CHUCK TURNITSA is a PhD Candidate at Old Dominion University, in the Modeling, Simulation and Visualization department. He currently works at VMASC as a Senior Project Scientist on projects concerned with knowledge based data modeling. He has authored or co-authored a number of papers concerning the use of ontology and ontological artifacts for modeling and simulation.