A Semantic Framework for Systems Engineering Standards

David Price
TopQuadrant
1730 Varsity Drive
Suite 220
Raleigh, NC 27606
dprice@topquadrant.com

Allison Barnard Feeney,
Dr. Albert Jones
National Institute of Standards and Technology
100 Bureau Drive, Stop 8260
Gaithersburg, MD 20899
Allison.BarnardFeeney@nist.gov
Albert.Jones@nist.gov

This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States.
Published and used by INCOSE with permission from the authors.

Abstract. Systems engineers and asset managers create and maintain models of components and systems associated with long-lived, engineering assets. Component models come from many domains, disciplines and applications. System models typically require integration of these component models. This paper presents the results of an investigation into the convergence of three distinct themes: the growing numbers of standards for engineering data, the maturing of Semantic Web technology, and a framework for architecting and relating engineering vocabularies and data. This investigation sought to determine whether these themes, used together, could enable industry to make significant progress toward model-based systems engineering in the areas of collaboration, knowledge management, data integration and timely decision making. This paper argues that semantic technologies have matured to the point where they should be the preferred choice for developing future standards and frameworks.

Introduction

It is common practice in many organizations to create and maintain computer models and engineering data associated with long-lived engineering assets. These assets are complex systems comprising many individual components and subsystems. Typically, the models and data related to these components and subsystems are created for and maintained by different departments within the organization, with each using the best available tool for their purpose. Yet, organizations still desire an asset-level model, which requires the integration of those individual models and data. Standards that support much of the problem space are available, but they are not widely used. The companies that are integrating their data at the system level are using custom point-to-point solutions. With this investigation, our goal was to leverage the growing number of standards for engineering data, the maturing Semantic Web technologies, and a proposed framework for architecting and relating engineering vocabularies and data to address model-based systems engineering. Using the results of our investigation, we sought to determine if it is possible to make major, rather than incremental, improvements over the current practice by applying Semantic Web technology, while also supporting the reuse and alignment with existing applications and standards such as Systems Modeling Language (SysML) (OMG 2012).

In a 2004 paper (Price and Bodington 2004), the authors described a vision in which future Semantic Web technologies would be used as a basis for that integration. Semantic Web technologies have matured considerably since 2004 and we believe that integration from these technologies is a capability here today. Nevertheless, few organizations use Semantic Web technologies to implement that vision. Have these technologies matured to the point where they can represent real-world assets accurately and completely? This paper describes the current state of those technologies and argues the technical merits and technological capabilities to support engineering data and models associated with long-lived assets.

Capabilities in Semantic Web technologies are driven by two key ideas: anyone can say anything about anything and what they say is computer-interpretable. Just these two ideas lead to many of the interesting characteristics of Semantic Web technologies. The specific Semantic Web technologies discussed in this paper are the World Wide Web Consortium's (W3C) international standards:

- Web Ontology Language 2 (OWL) (W3C 2009b)
- Resource Description Framework (RDF) (W3C 2004)
- SPARQL Query Language for RDF (SPARQL) (W3C 2008)
- SPARQL Inference Notation (SPIN) (W3C 2011)
- Simple Knowledge Organization System (SKOS) (W3C 2009a)

To demonstrate the capabilities of these technologies, we focus on the scenarios involving data exchange, data sharing, and data integration. In the remainder of this section, we discuss these scenarios and how they might be accomplished using current international standards for systems engineering and asset management.

Data Exchange

Most data exchanges involve two kinds of data structures: those that capture generic reference data and those that specify the details of the engineering items and their properties. Three standards have been developed to facilitate those exchanges: ISO 10303-233 (AP233) (ISO 2012) and ISO 10303-239 (AP239) (ISO 2005) and OASIS Product Life Cycle Support (PLCS) (OASIS 2013). ISO 10303-233 specifies an application protocol for the representation of systems engineering data associated with the design process. It defines the context, scope, and information requirements for various development stages during the design of a system. OASIS PLCS enables structured data exchange to support complex assets throughout their complete life cycle - from concept to disposal. Figure 1 shows data exchange as described in the PLCS standard.

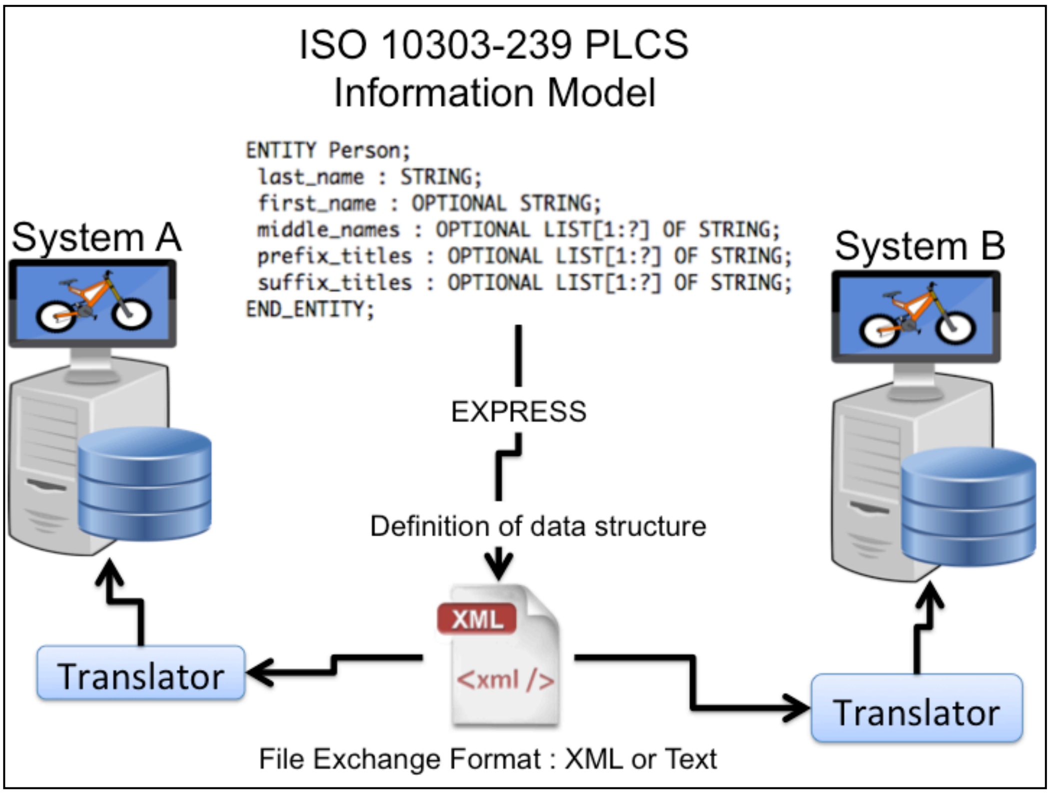

Figure 1. OASIS PLCS TC Data Exchange Diagram

The schemas for both standards are written using ISO 10303-11 (EXPRESS) (ISO 2004) and the Reference Data are written using OWL. These standards are actually implemented using clear text and Extensible Markup Language (XML) (W3C 2012) methods. In many industries, a simpler architecture where standard schemas are written directly in the W3C XML Schema language is employed.

Data Sharing

Data sharing means that the sender and receiver applications have the same understanding of the data. For many industrial organizations, creating or supporting a data-sharing environment is essential. Many standards, including PLCS, suggest that the data exchange capabilities offered by the standard can be the basis for that shared environment. Figure 2 shows a conceptualization of that environment as described by the OASIS PLCS Technical Committee.

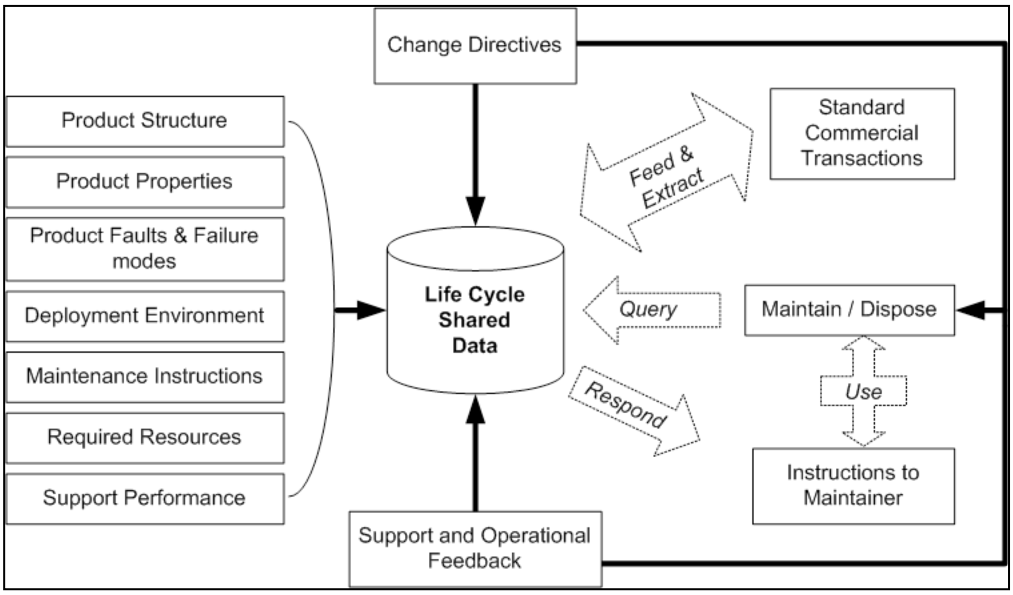

Figure 2. OASIS PLCS Shared Data Diagram

The problem with this conceptualization is that the underlying schema language and schemas for AP233 and PLCS, EXPRESS, do not support sharing because they are designed to capture a snapshot in time and must make all data explicit. Additionally, the complex relationships in data exchange schemas are perhaps better hidden from application users. A better technology and broader framework are required. We discuss these topics in the next section.

Data Integration

Data integration, or data federation where data sources remain more loosely coupled, means that applications can understand and use data from a variety of sources or standards. Although AP233 and other similar standards have fairly broad scopes, they cannot cover all the data requirements of an organization creating or supporting complex engineering assets. Therefore, data federation or integration using multiple standards is an approach that might be of interest to the user community. Examples of this might be AP233 and OMG SysML, or an organization's internal data schemas or relational databases. This is not possible with the technologies currently in use. How Semantic Web technologies address this will be discussed later in this paper.

IDIOM Architecture

Industrial Data Integrated Ontologies and Models (IDIOM) (Price et al. 2011) is an architecture and an information technology (IT) framework for representing and exchanging industrial data using technologies, methodologies, and approaches that are current industry best practice (see Figure 3). IDIOM has put ontologies specified using logic-based languages at the center of that IT framework. Our investigation used that framework and Figure 3 shows a high level view of the scope of our investigation: natural language technical dictionaries, data models and ontologies.

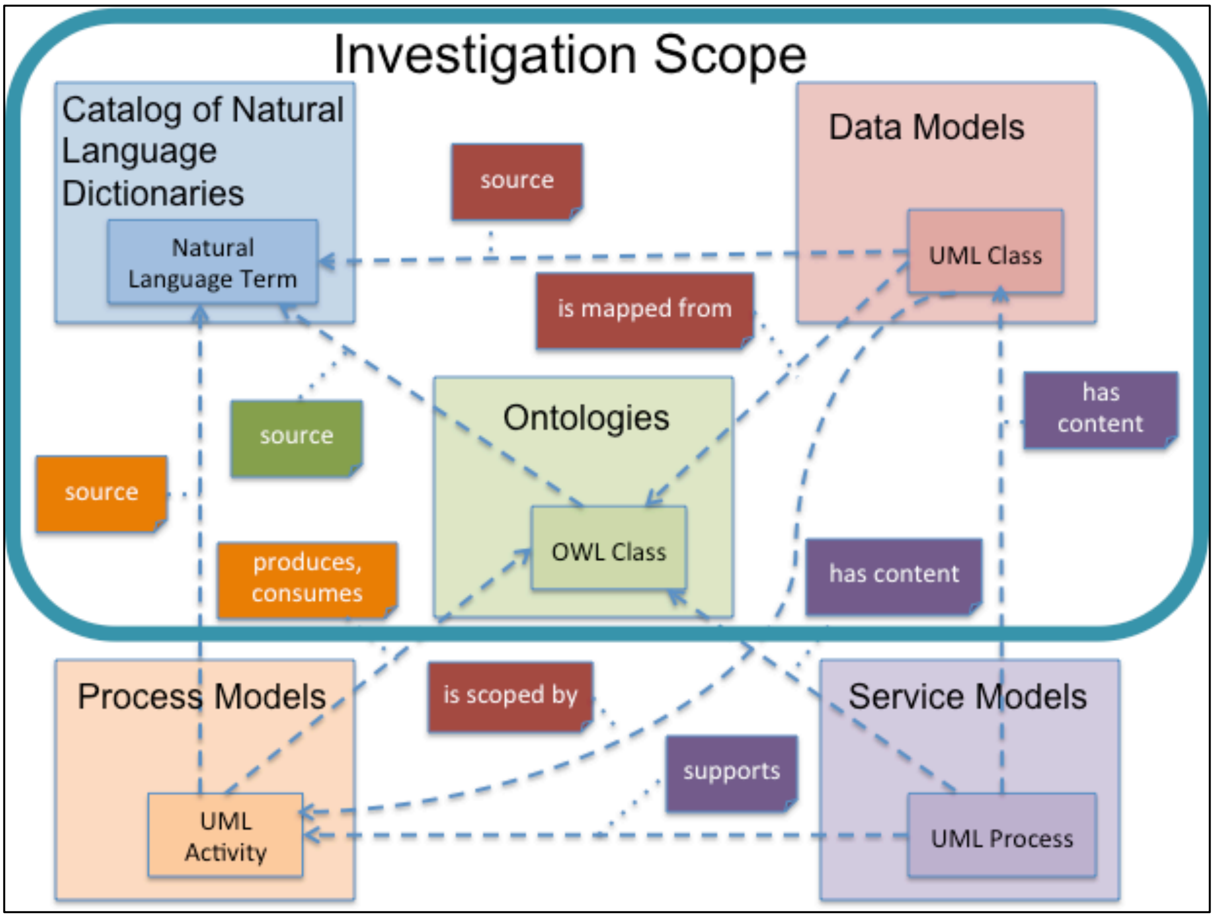

Figure 3. The IDIOM IT Framework and Investigation Scope

The primary driver behind IDIOM was to develop a means of leveraging the experience and knowledge embedded in standardized information models developed by domain experts, such as those within the ISO subcommittee on Industrial Data (ISO 2013). These information models currently serve as a robust, neutral, file format that in 2001 was projected to save approximately $1 billion per year by reducing interoperability problems in the automobile, aerospace, and shipbuilding industries (Gallagher et al. 2002). However, the standards are written in a purpose-built information modeling language called EXPRESS. EXPRESS does not have widespread acceptance and few software engineers learn EXPRESS in school. To prevent industry's significant investment in developing these information models from being lost, leadership of SC4 approved development of the IDIOM architecture for harvesting the knowledge in its information models into new, more widespread technologies. IDIOM is agnostic as to the technology chosen for a particular architectural component; the framework recommends using the best available technologies and tools for the purpose.

In the following sections, we present the results of our investigation. We discuss the benefits of the Semantic Web technologies chosen for use in this investigation. We also provide various examples to demonstrate the utility of both the framework and semantic technologies.

Natural Language Dictionaries

The role of the Catalogue of Natural Language Dictionaries component in the IDIOM IT framework is to provide a more semantically complete information sharing, exchange, or integration solution for industry. Typical engineering standards are stand-alone documents that contain definitions of terms, sometimes including terms by normative reference or by copying terms from other engineering specifications. The natural language dictionaries in IDIOM are intended to provide a means for model elements, in particular ontology elements, to be more closely matched to terms used in the real world. It also enables queries to find where natural language terms are used as a means for finding specifications related to particular subjects.

This component of the IT framework is realized as a collection of natural language dictionaries, largely published elsewhere, augmented by users of the framework as desired.

IDIOM includes technical dictionaries developed with SKOS, the Simple Knowledge Organization System. SKOS is addressing an area of work developing specifications and standards to support the use of knowledge organization systems such as thesauri, classification schemes, subject heading systems, and taxonomies within the framework of the Semantic Web. SKOS allows different groups to publish terminology, while allowing them to link to one another using uniform resource identifiers (URIs). SKOS enables organizations to:

- identify a Concept by a URI
- define one or more Terms for that Concept
- group Concepts into Concept Schemes
- relate Concepts and reuse Concepts between Concept Schemes
- annotate Concept Schemes and Concepts
- extend the standard as required for specific use cases

The dictionary of interest in this paper is the INCOSE Systems Engineering Handbook (Haskins 2011). As with any engineering handbook, there are defined terms, references to terms from other dictionaries and many terms implicitly defined within the text of the handbook. We used *Appendix D: Terms and definitions* that makes reference to terms from *ISO/IEC 15288:2008 – Systems engineering – System life-cycle processes* (ISO and IEC 2008).

Terms and Concepts. According to the SKOS Primer (W3C 2009a) terms and concepts are not the same. Instead, a Concept is the idea for which there may be a variety of terms in a variety of languages. The following example shows how the Concept of System from the INCOSE handbook might be identified and labeled.

```
<http://incose.org/vocabulary/syseng-handbook/System>
        a                   skos:Concept ;
        skos:prefLabel      "System" .
```

The example is written in the W3C standard text format, Turtle (W3C 2013), and makes the following statements.

- There is a thing identified by the following uniform resource identifier (URI) that is a SKOS Concept: http://incose.org/vocabulary/syseng-handbook/System.
- That thing has a SKOS preferred label of “System” that is a character string.
- The language of the string can be specified; (e.g., "Système"@fr for French).

In addition to preferred labels, a Concept may have alternative and hidden labels. The documentation related to a Concept specifies using definitions, several kinds of notes, and examples. Technically, even definition is a kind of note and other kinds of notes are change notes, editorial notes, scope notes, and history notes. For example the Concept of a ‘Facility’ might include a definition and example as follows.

```
<http://standard.iso.org/15288/terms#Facility>
        a                   skos:Concept ;
        skos:definition     "the physical means or equipment for facilitating the
        performance of an action" ;
```

```
        skos:example            "buildings, instruments, tools" ;
        skos:prefLabel          "Facility".
```

Grouping Terms and Concepts. Concepts are grouped together into what SKOS calls Concept Schemes. A Concept Scheme can be used for purposes such as compiling a specific vocabulary, representing the terms and definitions found in a specific dictionary or specification, representing the terms in a corporate master reference data dictionary, or specifying a discipline-specific taxonomy. Concepts can be related to any number of Concept Schemes and so can be shared between vocabularies and dictionaries.

As an example, the INCOSE Handbook defines a set of terms and also relates them to some terms drawn from ISO/IEC 15288. To support this, two Concept Schemes were defined as follows. Note that the Concept Schemes are also identified using a URI and have a preferred label. SKOS also supports other information about the Concept Scheme. First the INCOSE Handbook:

```
<http://incose.org/vocabulary/syseng-handbook/INCOSESysEngHandbook>
        a                       skos:ConceptScheme ;
        skos:prefLabel  "INCOSE Systems Engineering Handbook" .
```

Then ISO/IEC 15288:

```
<http://standard.iso.org/15288/terms#SystemsEngineering-SystemLifeCycleProcesses>
        a                       skos:ConceptScheme ;
        skos:prefLabel  "Systems engineering - System life-cycle processes" .
```

Relating Terms and Concepts. SKOS supports two important relationships between Concepts themselves: semantic relations and mapping relations. Semantic relations are defined for Concepts in the same Concept Scheme. They can be broader/narrower, transitive or not, and related. Broader and narrower are used to directly link Concepts into hierarchies, represent whole/part relationships, and represent class/member relationships. Their transitive flavors are used to make relationships farther up or down the hierarchy. Related is used to link Concepts that are peers or siblings where one is not broader than the other.

Mapping relations match Concepts, typically from other vocabularies and Concept Schemes. Matches can be (1) exact meaning they have the transitive property, (2) close meaning they are nearly the same, but not transitive, and (3) broad or narrow, meaning they are simply related such as peers or siblings. A few examples will make these ideas more clear.

The INCOSE Handbook ‘Supplier’ concept ‘has broader’ the ‘Organization’ concept.

```
<http://standard.iso.org/15288/terms#Supplier>
        a                       skos:Concept ;
        skos:broader     <http://standard.iso.org/15288/terms#Organization> .
```

To add clarity when reading this, the SKOS Primer suggests that the word "broader" should be read here as "has broader concept …". One may read “narrower” in the same way.

The INCOSE Handbook ‘System’ concept and the ISO/IEC 15288 ‘System’ could be specified as an exact match. The INCOSE Handbook ‘System element’, which has alternative label ‘Subsystem’, could be specified as a close match to the ISO/IEC 15288 ‘SystemElement’, which has an alternative label ‘Element’.

SKOS Extensions. SKOS was designed to be lightweight and extended when used in practice. We found it useful, for this work, to extend SKOS in two ways: subclasses and sub-properties. We define subclasses of the SKOS Concept for discipline-specific classes to partition or categorize concepts found in the INCOSE Handbook. Because these subclasses are INCOSE-specific they are in the INCOSE system model namespace; but, we place them in separate SKOS concept schemes to enable modularity and reuse. Figure 4 shows how we can partition the INCOSE Handbook Table of Contents into subclasses.

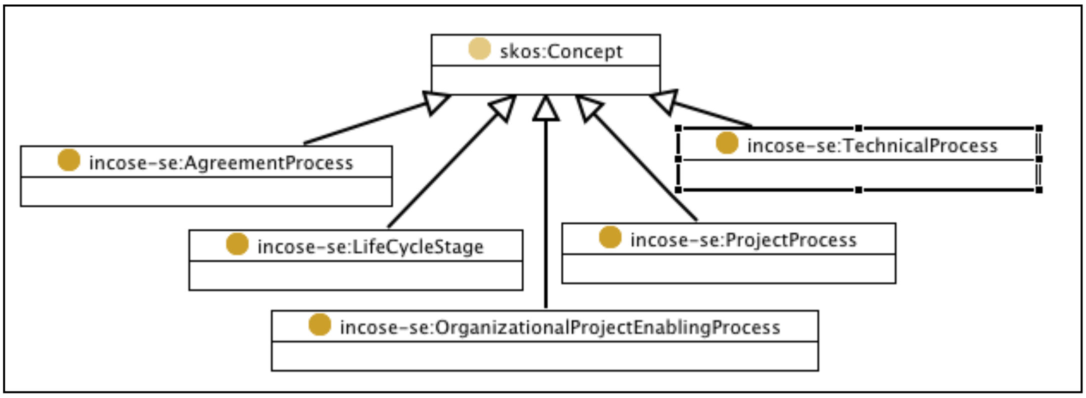

Figure 4. INCOSE-specific subclasses of SKOS class Concept

We define sub-properties of SKOS properties to distinguish isKindOf from isMemberOf. These sub-properties are not discipline-specific; they are in the IDIOM namespace. Figure 5 shows two additional properties that we introduced to distinguish between *is kind of* and *is member of* (or *is instance of*). This distinction is important for engineering applications.

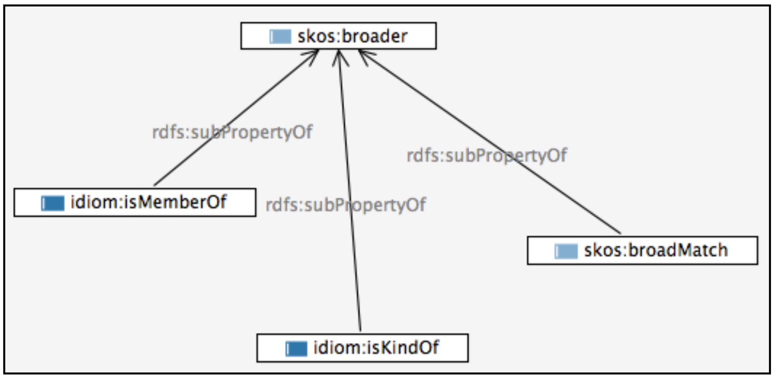

Figure 5. Engineering-specific sub-properties of SKOS property broader

SKOS, with minor extensions, works well for publishing engineering technical dictionaries. SKOS allows organizations to define business concepts precisely, as with current document-based specifications. Since, however, it is on the Web, or at least on an organization's intranet, sharing, modularity and reuse are possible. It enables the simple reuse of one dictionary within another, by reference, without copy and paste. Blending, merging, or aggregating new concepts and data is much simpler. The benefits compound as more linked data is made available on the Internet.

Data Models

The purpose of the Data Models component of the IDIOM IT framework is to support processes such as interchange, integration, archiving and preservation, service content specification, and database structure specification. Data models are often scoped for use in a particular process. Data models can provide additional constraints not possible in the Ontology; and, Ontologies may specify the semantics of the concepts that are modeled in a data model. Data Models related to Ontologies are also capable of providing a 'closed world' view of the knowledge in the 'open world ontology' should that capability be needed.

Within the IDIOM IT framework, data models may be specified in implementation technology-independent or abstract data modeling languages such as UML, SysML and EXPRESS. Data models may be tied to one or more implementation languages or bindings such as XML Schema. Bindings are typically mappings to various file formats, such as text and XML files. They are also useful in model-driven approaches such as the ISO 10303-28 EXPRESS to XML Schema mapping standard (ISO 2007). However, a model-driven approach to bindings is not a requirement on all data models. For example, a UML class diagram with definitional text that helps describe a hand-crafted XML Schema is also a possible approach to standardizing a data model for data exchange. Semantic technologies offer many potential benefits when using ISO AP233 and OASIS PLCS for data exchange and data sharing. In the following sections, we describe several of those benefits.

Proxy Ontologies. As stated earlier, ISO AP233 and OASIS PLCS schemas were developed using the ISO EXPRESS modeling language, however most industrial applications use XML Schemas directly. However, simply using a file format, such as XML schema, for exchange can cause serious data exchange problems with respect to interoperability and reuse and does not support addressing data sharing problems. To address these concerns, we developed an approach based on the use of a *proxy ontology.*

A proxy ontology is an OWL ontology that has the same structure as another schema, model, or database. In the *reeper* project (NIST 2011) NIST implemented an EXPRESS XML-to-OWL RDF/XML transform as an example of producing a proxy OWL ontology from EXPRESS. This effort maintained as direct an alignment between the EXPRESS and OWL as possible. No attempt was made to leverage any aspects of OWL that might be used to support reasoning. The sole purpose of the ontology is to be the basis for data exchange where the exchange schemas and related data are available as OWL/RDF. Doing so provides four main advantages:

- Data exchange translators can be written using the SPARQL language regardless of the schemas of the data source.
- Use of a consistent URI generation strategy for data instances can enable data about a System, for example, to be integrated more easily.
- In cases where OWL/RDF is used for the taxonomy or reference data, direct links can be made to those domain-specific definitions.
- Use of OWL for data exchange enables the use of the SPARQL and SPIN languages to perform validation.

SPARQL and SPIN. SPARQL is the widely implemented RDF query language. Its ASK capability can be considered a data validation constraint. For example, the following will return False unless there is someone named "Alice" who has the email address "alice@work.example."

```
ASK  { ?x foaf:name  "Alice" ;
       foaf:mbox  <mailto:alice@work.example> }
```

SPIN, which has an open source implementation available, includes a vocabulary that links SPARQL queries to OWL classes. Unlike OWL reasoners, SPIN 'closes the world' for validation. So, SPIN can treat SPARQL queries as data validation constraints. Additionally, SPIN is a very capable data mapping or transformation technology. One major advantage of using SPIN over pure SPARQL is that SPIN specifies an ontology for SPARQL queries. This means that SPIN constraints and functions can be stored with the proxy ontology in the RDF database.

Semantic Web languages support the layering of ontologies and constraints so that they can be applied only when necessary. Figure 6 shows one such arrangement with three layers.

These layers show how business-specific constraints can be layered on top of more general or standard constraints.

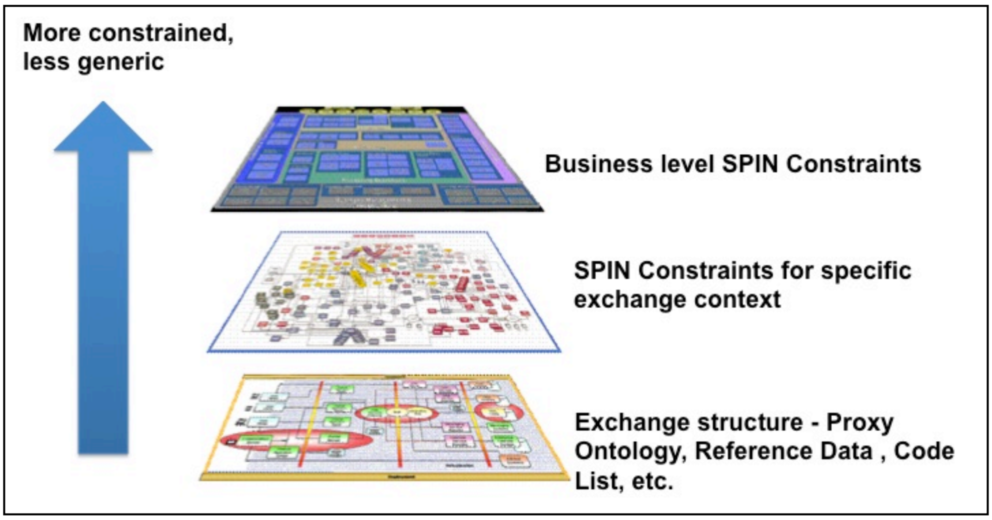

Figure 6. Three layers of Ontologies and Constraints

Multi-schema data exchange. The EXPRESS-based exchange file standards do support data from multiple schemas in the same file; however, that capability is not widely implemented or used. The same can be said of XML Schema-based exchange standards. In contrast, OWL ontology-based data being based on multiple ontologies is such a common occurrence that it's often overlooked as being trivial. So, mixing AP233 and PLCS data with STEP-based geometric shape data in the same OWL exchange file is simple. There is no need to send the AP233 file with string-based pointers to STEP geometry files that might get separated – all the data can be in a single file.

Note that this can be a useful capability in a long-term data retention scenario. It means that all the schemas and data from multiple exchange protocols can be contained in a single file for retention. That's actually true even in situations where the data is not linked or related at all. For example, data can be combined for other reasons, such as to protect confidential data. Because of the distributed nature of the environment in which the Semantic Web languages were expected to operate, they have powerful capabilities for relating elements of different ontologies or schemas that can be employed as well. OWL sameAs, for example, can be used to state that two individuals are the same and OWL equivalentClass can state that two classes have the same membership.

Data federation and integration. As mentioned in the Introduction, AP233 and other engineering standards cannot cover all the data requirements associated with creating or supporting complex engineering assets – particularly when those requirements change over time. Data federation or integration using multiple standards, AP233 and OMG SysML for example, or internal data schemas or even relational databases are viable alternatives. Figure 7 shows the core idea of data integration from multiple sources based on Semantic Web technologies - this happens every day in numerous industrial settings.

Semantic Web technologies enable interchange, integration, archiving and preservation, service content specification, and database structure specification. EXPRESS schemas for most engineering information can be replaced by ontologies that look like the EXPRESS. In fact, nearly every use of EXPRESS and related data exchange can be replaced by OWL and SPARQL/SPIN. Replacing EXPRESS with OWL proxy ontologies means that "translators" can be written by modelers who know SPARQL/SPIN and the source and target OWL ontologies, regardless of the original data schema language.

While our investigation studied business-level EXPRESS standards, it should be noted that we did not study those that required complex geometric capabilities and so those cases may fall outside of our results. For example, modeling nested arrays are a known weakness of the OWL language itself. For data exchange and sharing in domains that do not require dynamic customization, the extensibility provided through this approach may not be needed.

Links

The final component of the IDIOM IT framework within scope of this investigation is the links between framework components. The IDIOM IT framework takes advantage of the Semantic Web's inherent ability to relate elements to each other. Links are the mechanism by which models, Natural Language Dictionaries, and model elements in the various components of the framework are related. An example of a link includes the W3C OWL/RDFS seeAlso annotation. The following section describes how to link between a data model a technical dictionary, and the advantages that such linkage brings.

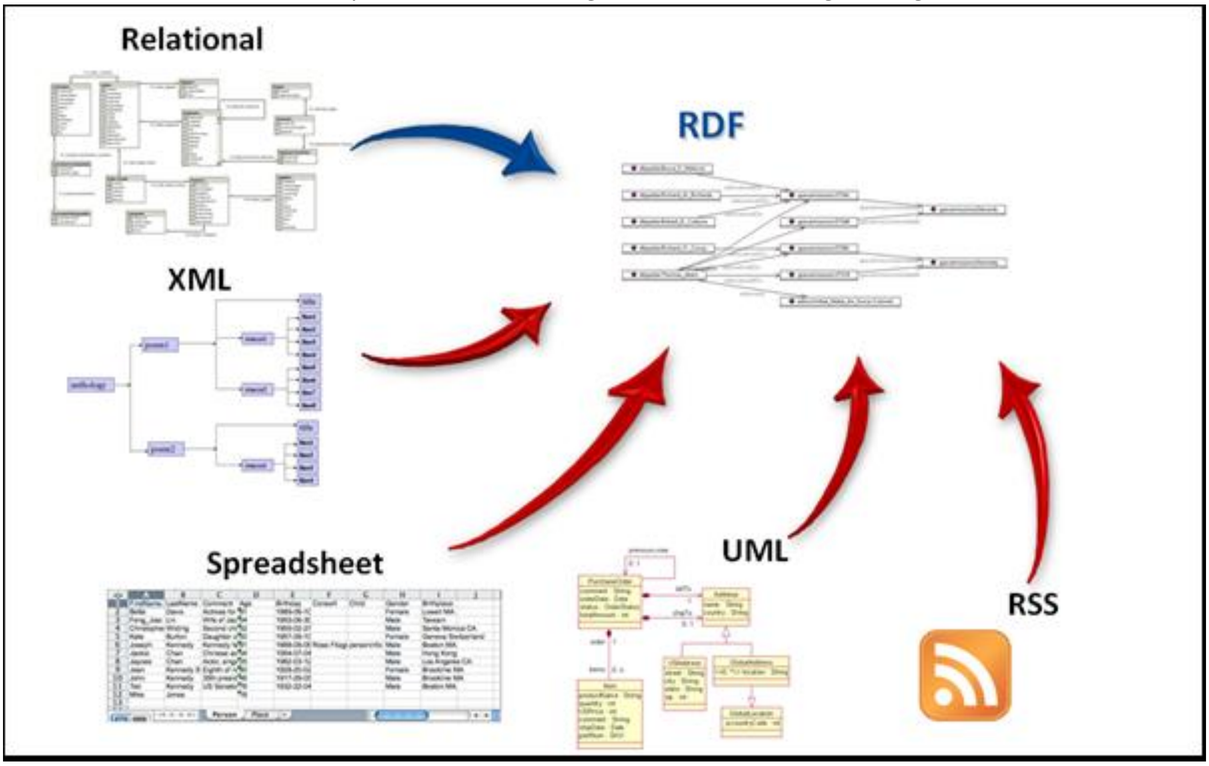

Figure 7. Semantic Web-based Data Integration

OWL Proxy Ontology-SKOS Technical Dictionary Links. Links between an OWL proxy ontology and a SKOS dictionary can be made quite simply. The OWL language includes a property defined in RDF Schema called 'seeAlso'. The property rdfs:seeAlso specifies a resource that might provide additional information about the subject resource. This property may be specialized using rdfs:subPropertyOf to more precisely the nature of the information the object resource has about the subject resource. The object and the subject resources are constrained only to be instances of the class rdfs:Resource.

In Figure 8 the seeAlso property has been used to link AP233 proxy ontology classes to a term from the INCOSE Handbook definition known as SKOS.

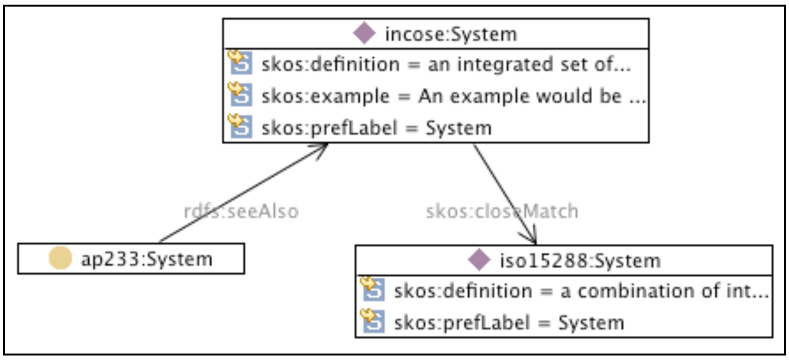

Figure 8. Using seeAlso

Since SKOS Concepts can be related, the usefulness of a link to a SKOS concept can grow in value over time. In the example, the INCOSE System concept records the fact that it is a closeMatch to the ISO 15288 System concept. The AP233 proxy-ontology user gets that extra information 'for free' because of the power of the links that SKOS supports.

If seeAlso is thought too general, then a subproperty of seeAlso can be created. In Figure 9 an IDIOM seeDictionaryTerm connects the AP233 Requirement class and the INCOSE Requirement concept. The property has a range of SKOS Concept and its domain is left undefined so that it can be used to relate any RDF-based item to a SKOS Concept.

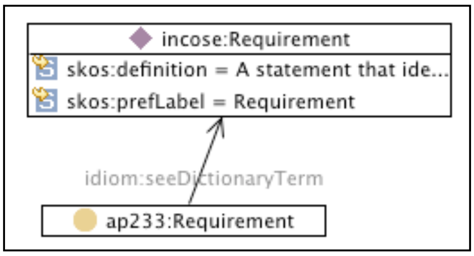

Figure 9. Specializing seeAlso

Our property is defined as a sub-property of rdfs:seeAlso and many tools such as SPARQL endpoints allow the user to specify whether or not simple RDF Schema inference is to be applied prior to executing the query. Therefore, both a general query using rdfs:seeAlso and a specific query using idiom:seeDictionaryTerm would return the same link.

Federation of Links. Due to the nature of Semantic Web links, simply federating the linksets and then navigating them can often produce interesting results. For example, in our investigation we considered the AP233 proxy ontology and a bespoke bike ontology developed for demonstration purposes. It was straightforward to link elements in each ontology to the INCOSE Handbook SKOS after the ontologies were federated by OWL import statements. That enabled queries from the SKOS technical dictionary concepts outwards. Unlike many other technologies, RDF links are easy to navigate in the reverse direction. Figure 10 shows the links to the INCOSE SKOS Requirement concept from AP233 and from the bike ontology. The References tab shows a 'Where Used' view of the item in focus.

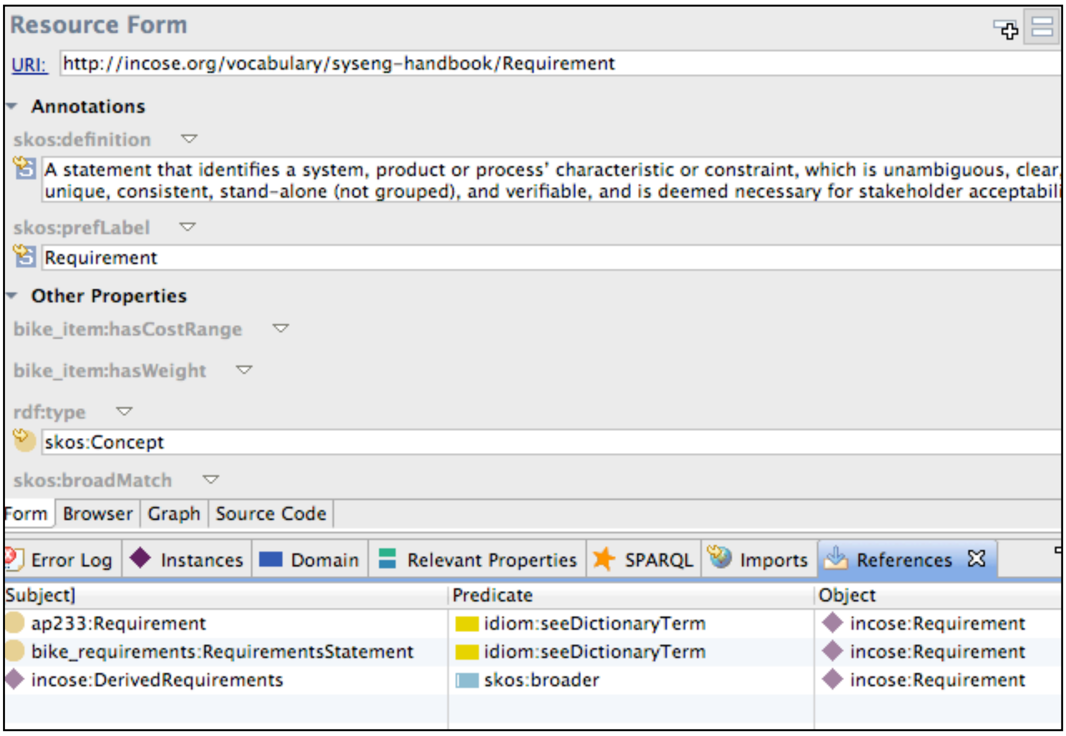

Figure 10. Chasing Links

As you would expect, the "lnks between models" are far simpler using Semantic Technologies than with anything else.

Conclusion

At the outset of this investigation, we sought to determine if a significant advantage could be realized for model-based systems engineering if we combined advances in engineering standards, advances in Semantic Web technologies, and a framework developed with the goal of harvesting value from international engineering data exchange standards.

We introduced the IDIOM architecture, which was developed to enable strong relationships to be made between information technology artifacts, and throughout the paper we have discussed Semantic Web technologies and how they can be used to fill the roles of components in the IDIOM IT framework including the links.

The aspect of this investigation into SKOS for technical dictionaries showed that with minor extensions it is an ideal solution for the publication and sharing of engineering technical dictionaries. This is true even without the advantage of being able to link easily to other artifacts that were part of the investigation.

The aspect of this investigation into OWL proxy ontologies showed that it is indeed a viable approach, specifically:

- Semantic Web technologies and the industry needs that drove the development of systems engineering and engineering asset management align remarkably well. Success with standards in these disciplines like SysML and PLCS has been achieved, and reusing them or recasting them into a Semantic Web approach can lower barriers to implementation even more. Using Semantic Web technologies to harvest standardized domain models in lesser-known languages like EXPRESS can lower the barrier to widespread adoption of the concepts in the standard schemas.
- This approach can be executed without any loss of the current model interchange and data exchange capabilities. Additionally, interoperability between Semantic Web and traditional technologies such as XML is straightforward. Semantic Web-based

implementations add significant benefit as they are a step towards enabling the shared data usage scenario that is not well supported today.

The standards and languages supporting Systems Engineering and Engineering Asset Management were developed over the past decade. As this investigation shows, taking advantage of the available technology developments in and around the Semantic Web can enhance application of this standards and languages, bringing significant benefits to engineering organizations today.

References

Haskins, C., ed. 2011. *Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities.* Version 3.2.1 Revised by M. Krueger, D. Walden, and R. D. Hamelin. San Diego, CA (US): INCOSE.

ISO (International Organisation for Standardisation). 2004. ISO 10303-11. *Industrial automation systems and integration – Product data representation and exchange – Part 11: Description methods: The EXPRESS language reference manual.*

———. 2005. ISO 10303-239. *Industrial automation systems and integration – Product data representation and exchange – Part 239: Application protocol: Product life cycle support*

———. 2007. ISO 10303-28. *Industrial automation systems and integration – Product data representation and exchange – Part 28: Implementation methods: XML representations of EXPRESS schemas and data, using XML schemas.*

———. 2012. ISO 10303-233. *Industrial automation systems and integration – Product data representation and exchange – Part 233: Application protocol: Systems engineering.*

———. 2013. "TC 184/SC 4 Industrial data." Last accessed 5 March. http://www.iso.org/iso/standards_development/technical_committees/list_of_iso_technical_committees/iso_technical_ committee.htm?commid=54158

ISO and IEC (International Organisation for Standardisation and International Electrotechnical Commission). 2008. ISO/IEC 15288. *Systems engineering – System life-cycle processes*.

National Institute of Standards and Technology. 2011. "Reeper." http://www.nist.gov/el/msid/reeper.cfm.

OASIS. 2013. "OASIS Product Life Cycle Support (PLCS) Technical Committee Frequently Asked Questions Page." Last accessed 3 March. http://www.oasis-open.org/committees/plcs/faq.php.

OMG (Object Management Group). 2012. *OMG Systems Modeling Language, Version 1.3*. Last modified 1 June. http://www.omg.org/sysml/1.3/.

Price, D., D. Leal, and A. Barnard Feeney, eds. 2010. "IDIOM Architecture Specification." Last modified April. http://futurearch.wikispaces.com/Documents.

Price, D. and R. Bodington. 2004. "Applying Semantic Web technology to the life cycle support of complex engineering assets." *The Semantic Web – ISWC 2004. Lecture Notes in Computer Science*, Volume 3298/2004, 812-822. doi: 10.1007/978-3-540-30475-3_56.

Rhizomik.net. 2013 "ReDeFer" Accessed 3 March. http://rhizomik.net/html/redefer/.

W3C (World Wide Web Consortium). 2004. *Resource description framework (RDF): Concepts and abstract syntax*. W3C Recommendation 10 February. http://www.w3.org/TR/rdf-concepts/.

———. 2008. *SPARQL query language for RDF*, W3C Recommendation 15 January. http://www.w3.org/TR/rdf-sparql-query/.

———. 2009a. *Simple knowledge organization system (SKOS) Reference*, W3C Recommendation 18 August. http://www.w3.org/TR/skos-reference/.

———. 2009b. *OWL 2 web ontology language, structural specification and functional-style syntax*, W3C Recommendation 27 October 2009. http://www.w3.org/TR/owl2-syntax/.

———. 2011. *SPIN - Overview and Motivation,* W3C Member Submission 22 February. http://www.w3.org/Submission/spin-overview/.
———. 2012. "Extensible Markup Language" Last Modified 24 January. http://www.w3.org/XML/.
———. 2013. *Turtle - Terse RDF triple language*, W3C Candidate Recommendation 19 February. http://www.w3.org/TR/2013/CR-turtle-20130219/.
Gallagher, M, A. O'Connor, and T. Phelps. 2002. *Economic Impact Assessment of the International Standard for the Exchange of Product Model Data (STEP) in Transportation Equipment Industries.* Planning Report 02-5. National Institute of Standards & Technology. Last modified December. http://www.nist.gov/director/planning/upload/report02-5.pdf

Biography

David Price has 30 years experience in software and engineering data management as Managing Director and Consultant at TopQuadrant Limited, the UK subsidiary of TopQuadrant Inc. Previously David was an IBM Senior Software Engineer and Eurostep, Inc. Principle Consultant. David has experience in creating and applying standards to government and industry problems in the computing, oil and gas, aerospace and defense industries. David led TopQuadrant's software-as-a-service delivery of an Oil and Gas reporting semantic database and has provided services to NASA, NIST, a major agribusinesses company and a defense industry consortium on asset management.

Allison Barnard Feeney is a Group Leader in the Systems Integration Division at the National Institute of Standards and Technology, a group dedicated to the solution of national problems related to measurements and standards supporting systems interoperation in the manufacturing sector. Allison has worked in the areas of manufacturing standards implementation, conformance testing, product data standards, and systems integration for the last 25 years. She has been a key participant in the development of the STEP product data standard (STEP - Standard for the Exchange of Product Model Data, ISO 10303). Ms. Barnard Feeney was awarded the Department of Commerce Silver Medal in 2005.

Dr. Albert Jones has spent close to thirty years at the National Institute of Standards and Technology (NIST). Currently, he is the Scientific Advisor for the Systems Integration Division in the Engineering Lab. Before this, he managed the Enterprise Systems Group for more than ten years. The Group focused on supply chain integration, management, and logistics. Prior to that, he was Deputy Director of the Automated Manufacturing Research Facility at NIST. Dr. Jones has published more than two-dozen journal papers and fifty conference papers. He is on the Engineering Advisory Boards at Morgan State University and Loyola University. Before coming to NIST, he held faculty positions at Loyola University and Johns Hopkins University.