Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 153 (2019) 268–276

Procedia Computer Science

www.elsevier.com/locate/procedia

17th Annual Conference on Systems Engineering Research (CSER)

The Ontology of Systems Engineering: Towards a Computational Digital Engineering Semantic Framework

Douglas Orellana[a]*, William Mandrick[a]

[a]*Science Applications International Corporation, 12010 Sunset Hills Road, Reston, VA 20190*

Abstract

The goal of implementing an enterprise digital engineering strategy is to improve data sharing throughout system conceptualization, development, manufacturing, operations, sustainment and their supporting organizations. Data sharing is a critical element of interoperability of models and other digital artifacts supporting the systems engineering process. Achieving interoperability within systems engineering requires eliminating differences of syntax and semantics to take advantage of automation, augmentation, and artificial intelligence capabilities that will increase process efficiency and artifact quality. This paper describes the development of a process-centric Systems Engineering (SE) reference ontology, based upon ISO/IEC/IEEE 15288 Systems and Software Engineering – Life Cycle Processes [1]. The proposed SE reference ontology extends from the Basic Formal Ontology (BFO), which is a very small top-level ontology (TLO) designed specifically to support the integration of more specific ontologies that extend from it [2]. A reference ontology is intended to provide a comprehensive representation of the entities in a given domain encapsulating the terminological content of established knowledge. Unlike other systems ontologies, which focus on the target systems, this reference ontology is focused on the systems engineering process, providing context and purpose for the digital artifacts being built and linking the content of artifacts together into one lexicon.

Keywords: Systems Engineering; Ontology; Digital Engineering; Model Based Systems Engineering; MBSE; Digital Thread; Systems Modeling; Knowledge Management; Basic Formal Ontology; Information Content Entity;

* Corresponding author. Tel.: +1-310-535-6335;.
E-mail address: douglas.w.orellana@saic.com

1877-0509
10.1016/j.procs.2019.05.079

1. Introduction

The amount of information needed for developing today's complex systems is growing rapidly. For the past decade with the emergence of model based methods, information regarding systems development, manufacturing, operations and sustainment are creating a wide array of digital artifacts that need to be traced, cross referenced, and made widely available. Due to distributed sources of information, engineers are unable to piece together a holistic picture of the system due to differences in syntax and semantic content of the information being provided. Within the systems engineering discipline, it has become a harder problem as a digital transformation is occurring where what traditionally was done using Microsoft Office products is now taking a more formal approach, using a set of modeling tools. As environments change from the traditional ecosystem to this new ecosystem, information regarding the system are on information systems that do not share data with one another due to a lack of machine understanding—this is both a semantic and technological interoperability issue. Even within the modeling environment, disparate modeling languages, frameworks, and implementations have slowed down interoperability of the information systems supporting the systems engineering process.

In order to overcome the information interoperability gap, a systems engineering (SE) reference ontology is needed. Unlike other systems ontologies being developed, instead of focusing on the systems being built, the ontology discussed in this paper is focused on the systems engineering process. The reason for this distinction is the fact that traditionally the systems engineering artifacts were driven by process or contract deliverables, and in order to understand the purpose of these artifacts, there needs to be a tie back into the activities in which these artifacts are being developed and delivered. Moving forward there also needs to be a re-engineering of the process outputs to better align to a model based ecosystem.

In what follows we describe the development of a process-centric SE reference ontology, based upon the ISO/IEC/IEEE 15288 Systems and Software Engineering—System Life Cycle Processes [1] and Model-Based Systems Engineering (MBSE) terminology. The classes and relationships for this reference ontology are derived directly from the accumulated knowledge and methods of the SE discipline as described in the authoritative literature. The result is an exhaustive classification of the process aggregates, discrete activities, person roles, information content entities, and relationships between these things. The classes and relationships are arranged in compliance with Basic Formal Ontology (BFO), and are in a computable format (owl-rdf and xml) facilitating:

- The formal representation of objects, object attributes, processes, and relationships that make up the SE domain
- Enhanced human reasoning in the SE domain
- Computer assisted reasoning (e.g. query and inference) in the SE domain
- The semantic integration of SE sub-domains
- The generation of new knowledge gained through semantic integration

2. Background on Ontologies

The purpose of ontology is to identify and categorize (classify) the continuant objects, object properties, occurrent processes and relationships that make up the world we inhabit. The result of the ontological process is an ontology, which is defined as, "a representational artifact, comprising a taxonomy as proper part, whose representations are intended to designate some combination of universals, defined classes, and certain relations between them." [2] Both the ontological process and the resulting representations serve to improve human reasoning about the world, and more recently also serve as computational frameworks. Figure 1 shows the levels of granularity for a suite of interoperable ontologies. Interoperability is gained through a top-down and bottom-up approach to ontology development that will be explained in the following sections.

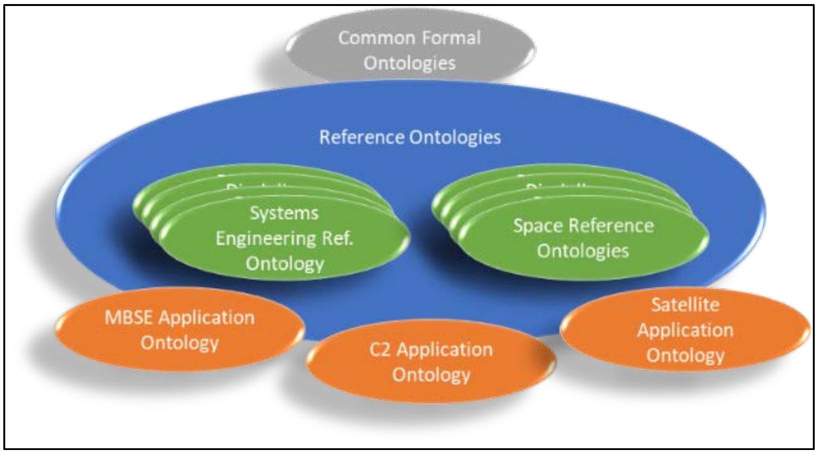

Fig. 1. Ontology Hierarchy

As already stated, ontologies are distinguished by their level of granularity. A Top-Level Ontology (TLO) consists of highly general categories and relationships, and serves as a unifying framework for the lower-level (more specific) ontologies that extend from it. The Basic Formal Ontology (BFO) is a very small TLO that conforms with ISO/IEC 21838-1 (Information technology—Ontologies—Top-Level Ontologies) [3]. BFO consists of only 34 of the most general class terms for continuant objects and occurrent processes, and is designed to support the development of more specific domain-level ontologies in a way that promotes ontological consistency through coordination between different groups. BFO is currently in the final stages of ISO certification as ISO/IEC CD2 21838-2 [4].

Ontologies can also be distinguished by their function. A Reference Ontology extends from a TLO and, "…is intended to provide a comprehensive representation of the entities in a given domain encapsulating the terminological content of established knowledge of the sort that is contained in a scientific textbook" [2]. An application ontology is more specific in that it is created in order to accomplish some task of local significance—e.g. to track Space Objects or Engineering Processes.

Application ontologies are built through a bi-directional (top-down and bottom up) approach. For example the Space Object Ontology (SOO), was designed to provide a consensus-based (realist) framework that includes the class terms and relationships required for establishing Space Domain Awareness (SDA) [5]. It was built as an extension of the Basic Formal Ontology (BFO) and the suite of Common Core Ontologies (CCO) [6], which respectively form its upper and mid-levels—the top-down part of the approach. The top-down part of the approach provides the organizational framework and logical structure for the lower-level application ontologies, meaning that any application ontology that conforms to the TLO will be interoperable with all other conformant ontologies. The bottom-up approach then examined the authoritative classes, relationships, and data elements that make up the Space Object domain in order to create a well-formed and conformant SOO.

3. Current Systems (Engineering) Ontologies

The array of systems (engineering) ontologies has varied in focus and detail. A growing number of organizations and researchers see the benefit of ontologies in systems engineering to maintain informational coherence, integrity, and interoperability [7-9]. [10] introduces an ontology spectrum that ranges the strength of ontologies being built. From the lower left to the top right of this spectrum the semantic richness increases, increasing the level of machine interpretation (rather than machine processing). The ontologies being built today for or within the systems engineering discipline span across the spectrum. The SE process ontology described in this paper is striving to be built with first order logic to be able to be reasoned by a machine.

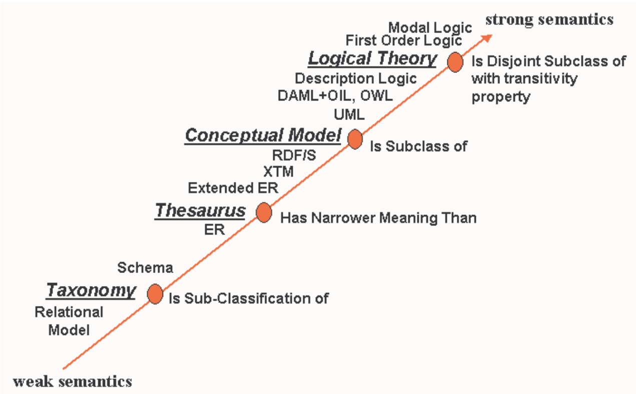

Fig. 2. The Ontology Spectrum [10]

The different attempts at building systems (engineering) focused ontologies have been mainly driven by efforts to strengthen model based methods within the systems engineering discipline. Due to the fact that the traditionally model based systems engineering has been associated with architecting and the systems modeling language (SysML), many of these ontologies are focused domain application ontologies, better referred to as system ontologies. A system ontology represents “a pattern for what constitutes a system: the parts and connections, identity, dependence, unity, and for engineered systems, replaceable components” [11]. [12] built a suite of ontologies to support cyber physical systems. The main purposes for this suite of ontologies is to support decision support and strengthening model driven approaches to developing cyber physical systems. A more commonly known effort at the NASA Jet Propulsion Laboratory (JPL), [13] has developed an organizational systems ontology to support the system to be developed at JPL. This ontology also builds a suite of ontologies. These ontologies are built in order to provide reasoning for requirement traceability, interface consistency, and viewpoint consistency for modeling artifacts within an architecture tool using SysML. This effort provides higher fidelity coupling of the ontologies with what is being developed within the models by taking the ontologies and creating SysML profiles that allow the instantiation of the model elements into a machine interpretable format. [14 -16] have taken a different approach to how they have built their ontologies by including process. [14] looks at developing a modeling framework for SysML that is very specific to autonomous systems and how the architectural models should be organized. [15] attempts to put some pedigree to their ontology by claiming adherence to ISO/IEC/IEEE 15288, but there is no inherent traceability back to the process. [16] shows a top level taxonomy of SE functional entities that more closely align with ISO/IEC/IEEE 15288 but has not gone beyond an initial taxonomy defining SE entities types by high level SE functions and SE objects. For example SE functions types are defined by the top level processes (Technical Management and Technical Execution), while SE Objects types are defined by actors and products with no relationships defined between the SE function and SE objects types.

Although there has been a resounding interest in using ontologies for aspects of systems engineering, there has been limited effort towards employing a top level framework from which these ontologies can consistently extend from so they can complement each other. BFO was developed specifically for integrating disparate, but related, lower level ontologies. Many of the ontologies being built today are for a specific purpose or organization versus being developed as a baseline reference framework for the systems engineering domain/discipline. As described in section 2, a layered ontological approach allows for a suite of semantically interoperable ontologies to be built and maintained, while being derived from BFO, from which interoperability can be standardized. Without wide use of a top level systems engineering ontology, achieving enterprise-level interoperability will just be out of reach.

4. Building a Systems Engineering Reference Ontology

Building a process-centric SE Reference Ontology consists of five processes, and numerous sub-processes, that are based upon best practices from countless past ontology efforts. The five processes are:

4.1. Scope the Ontology

The first step in developing any ontology is scoping out the parameters of what will be included in, and excluded from, the ontology. This is done by surveying authoritative reference materials and conducting subject matter expert interviews, in order to determine the boundaries of the domain. It is also important to answer a series of questions pertaining to who will be using the ontology and what purposes it will serve.

For example, a process-centric SE Reference Ontology is useful to any SE project, serving as a baseline from which to extend into a more specific application ontology—e.g. one concerned with a particular sub-set of SE processes. The reference ontology is useful to any SE project because it includes all of the processes enumerated and described in ISO 15288, as well as the sub-activities, the people participating in those activities, and the informational inputs and outputs of those processes. Excluded from the reference ontology for SE Processes are the environmental elements as well as any particular system created from some SE Process—these may be their own individual domain ontologies.

4.2. Establish a Lexicon

The second step for creating an ontology is to identify the high-frequency terms that will make up the classes (categories) in the ontology, thereby establishing an authoritative lexicon. These include terms for the physical object aggregates, individual objects (components), object properties or attributes, the processes they participate in, and the relationships that logically tie all of elements together in the SE domain. The output of this process is a lexicon of high-frequency terms, complete with their authoritative definitions. These terms will serve as the classes (categories) in the ontology.

4.3. Map to Top Layer Ontology

Once the high-frequency class terms have been identified, defined, and related they are then mapped to the TLO (e.g. the Basic Formal Ontology). This process creates the taxonomical hierarchies for the object aggregates, individual objects (components), object properties or attributes, and processes for that domain or reference ontology. This is done with an ontology editor, such as Protégé or Top Braid Composer, which creates a computable Web Ontology Language (owl) and Resource Description Framework (rdf) file. The TLO is first imported into the ontology editing tool, and the class terms are then mapped to them as sub-classes.

4.4. Establish Axioms

The next Sub-Process identifies and establishes the horizontal (axiomatic) relationships between the classes in the owl-rdf taxonomical hierarchies. For example, all SE Processes will have some Information Content Entity (ICE) as an output. As will be shown below, this step creates the highly useful axiom that SE Process XX has output some information content entity, where XX will be one of the many SE processes.

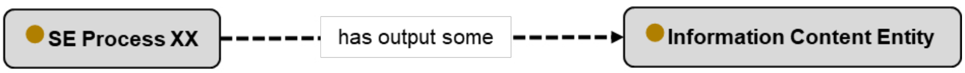

Figure 3. SE Process Relationship to Information Content Entity

The goal at this stage is to establish as many logical axioms as possible between classes, which will result in greatly enhanced query and automated reasoning. Well-formed axioms, are statements accepted as true, which are used as the basis for argument or inference. When axioms are in owl-rdf format, they serve as computable statements that enable powerful reasoning with class terms. When the class terms are associated with instance data, the effect is computational reasoning over vast data.

4.5. Populate with Instances

Once the classes (categories) have been created in owl-rdf, they can be populated with instance data—these are data elements about the real-world entities that occupy space and time. For the SE Process Ontology the instances include the actual people, processes, activities, and information content entities that occupy space and unfold through time. The result is that any axiomatic reasoning between classes (categories), becomes reasoning about actual (instances of) people, activities, and information content entities.

5. The Systems Engineering Reference Ontology and Initial Results

The process-centric SE reference ontology converts natural-language descriptions of SE processes into computable class hierarchies in owl-rdf. The class terms (also called categories) include SE related process aggregates, sub-processes, discrete SE activities, persons in roles (e.g. Chief Engineer, Requirements Manager, etc.) and Information Content Entities (ICEs) such as System Requirements Documents, Models, etc. The addition of first-order logical relationships (e.g. has part, has output, is accountable for, etc.) in owl-rdf facilitates reasoning about the classes—see Figure 4 below. Relationships are the manner in which two or more entities (classes) are associated or connected together. BFO recognizes three basic types of relation:

- Connecting class to class (e.g. Artifact is the 'bearer of' some Function)
- Connecting class to individual (e.g. System Engineer 'has individual' John Smith), and
- Connecting individual to individual (e.g. John Smith is the 'bearer of' John Smith's Engineering Skill).

As a prototype to this process-centric SE reference ontology we have taken seven out of the thirty SE processes defined in ISO/IEC/IEEE 15288 and begun building out the ontology: System Requirement Definition Process, Architecture Definition Process, Design Definition Process, System Analysis Process, Implementation Process, Verification Process, and Validation Process. The reason for starting with these processes over the others identified in the standard was that they most closely aligned with processes where Model Based Systems Engineering is being utilized.

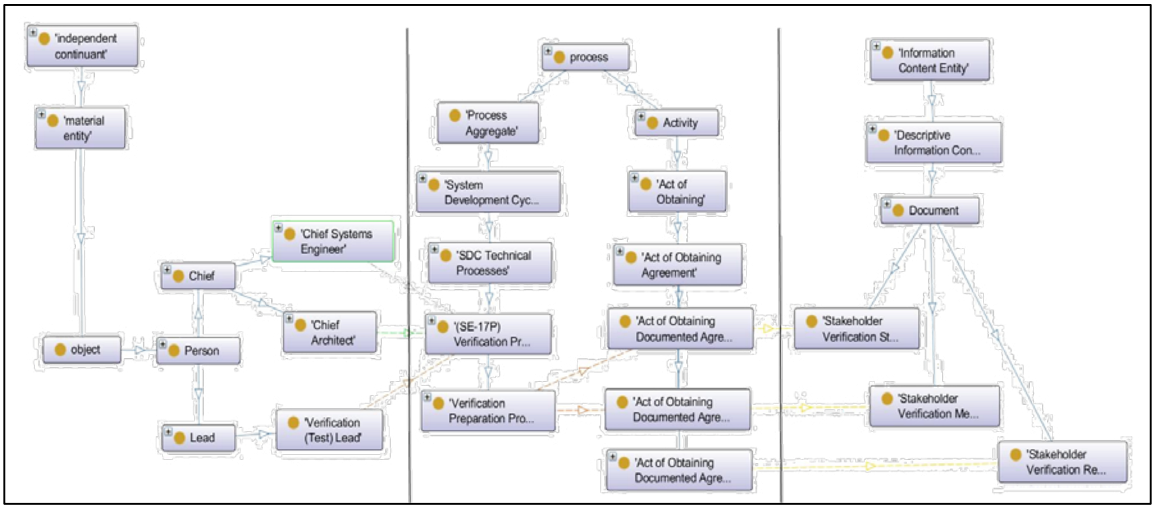

Figure 4. SE Ontology Alignment to BFO Example

As we walked through each process we began to identify patterns for the type of language being used and the content it described. Each process was composed of sub-processes, information content entities, and allocated actor(s). The sub processes themselves were also broken into actions. Interestingly, the actions throughout the seven processes used the same verbs (identifying, defining, selecting, etc.). Our findings are strikingly similar to those made previously by Siang Kok Sim and Alex H B Duff who concluded that, "what is needed is an ontology of design activities so that proponents of models or theories of design and practitioners have a shared understanding of what each specific design activity entails." [17] Such an ontology for SE processes, their constituent activities, as well as informational inputs and outputs would facilitate computationally assisted reasoning across the SE enterprise.

The information content entities were represented as inputs or outputs, were either explicitly stated or implied, and ranged from atomic to aggregate information content entities. Figure 4, shows an example of how we began to align the items identified in the standard to build out the ontology. The actors filled various roles during the SE process. The high level processes were treated as aggregates of activities. The actions were atomic acts that fit the pattern that showed up in the standard. The outputs and inputs described the information content entities, which were all document centric.

Initial analysis of the deconstruction has shown that in order to make ISO/IEC/IEEE 15288 more easily readable is to reduce the amount of similar verbs being used to describe the same types of acts if they are truly the same. Overall the natural language used in the document itself needs to be made consistent to ensure consistency within an ontological framework. As we continue to use this process using a digital ecosystem other changes need to be implemented. The major change that needs to happen is to turn away from the information content entities being documents and identifying atomic and aggregate entities that align with the model based systems engineering information systems that support the systems engineering process. This will more closely reflect the realization of this process and how the process is being used in a digital engineering strategy.

Converting natural language descriptions of the Systems Engineering processes into a computable ontology requires identifying the class terms and relationships that are either explicitly named or implied in authoritative sources. Once identified, the class terms in ISO 15288 for the person roles, processes, sub-processes, and information content entities are then placed into a taxonomical hierarchy in the Ontology Web Language Resource Description Framework (owl-rdf). Computer enhanced reasoning is attained once the logical relationships between classes are identified and built into the owl-rdf ontology. Figure 5 shows the conversion of the Verification Process into an owl-rdf computable ontology with classes and relationships.

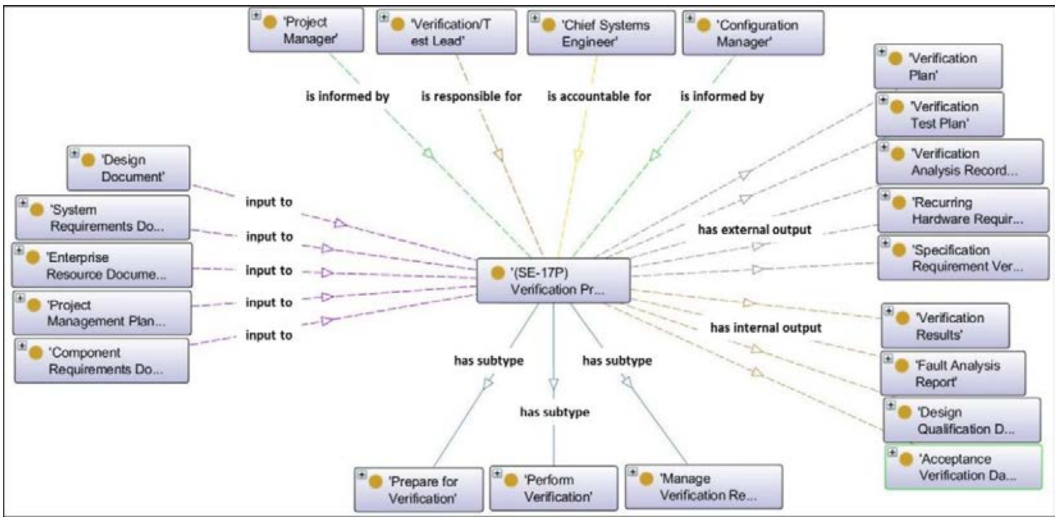

Fig. 5. Verification Process has three distinct sub-processes, which are further decomposed into discrete activities (see Fig. 6). Information Content Entities are inputs to, and outputs from, the process. The Chief Systems Engineer is overall 'accountable for' the process.

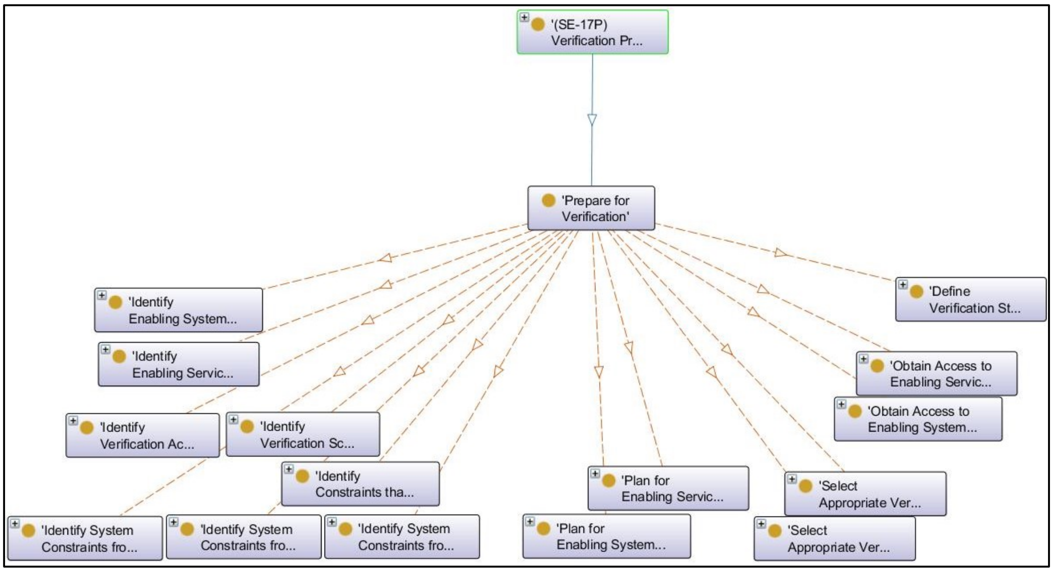

Fig. 6. The 'Prepare for Verification' Process is composed of 15 discrete activities. The + symbol in each box means that that activity can be further decomposed (not shown here due to space constraints) to show the participants (e.g. System Engineer) and the informational inputs and outputs.

With clearly defined class terms and a small set of logical relationships (in owl-rdf format) we are able to reason with and query the ontology. For example, the 'has part' relationship facilitates reasoning about processes, sub-processes, and discrete activities—the reasoning required for understanding workflows and analyzing the composition of process aggregates. The 'input to' and 'has output' relationships facilitate reasoning about the information content entities and their related processes—reasoning required for computer-assisted information management. Participation relationships such as 'is accountable for' and 'is responsible for' allow for reasoning about actors and the activities they participate in—the reasoning required for workforce analysis.

6. Summary and Path Forward

This paper has shown a new approach to building out the systems engineering ontology. This approach is similar to what the biomedical disciplines have used to make their data and discoveries more widely available to a larger audience. The current SE Ontology prototype is built on pedigree by the deconstruction of ISO/IEC/IEEE 15288 and linking the entities identified back to BFO, a standard for top level ontology (ISO/IEC 21838-1). Although we have not completed reviewing and deconstructing the SE processes, the seven processes we have deconstructed have shown patterns and inconsistencies that need to be addressed. The biggest obstacle moving forward will be using a document centric process in a model centric environment with model centric methodologies. In order to provide the greatest value to a digital engineering ecosystem the process inconsistencies will need to be corrected, and application ontologies need to be traced to the SE reference ontology.

The path forward is to continue to build out the ontology with the remaining breadth of the processes. First, we will stay true to the current language of the standard, but once we find deficiencies and inconsistencies we will want to document those areas and correct them as necessary. One area will be to reduce the number of synonyms used (capture vs. collect vs obtain, etc.). Although synonyms could be tied together within an ontology, the ISO/IEC/IEEE 15288 processes, could be misinterpreted by organizational judgement or bias. ISO/IEC/IEEE 15288 also needs to be injected with a modeling mindset to the where reports or viewpoints are aggregates of information content entities described in the processes.

As we move forward we would like an independent organization to endorse this effort so its use becomes more widely available and accepted through the systems engineering community. This will allow for more collaboration between organizations in establishing interoperable set of modular systems engineering ontologies, some which are already being created (e.g. JPL systems ontology).

References

[1] ISO/IEC/IEEE 15288 Systems and Software Engineering – Life Cycle Processes, 2015, Institute of Electrical and Electronics Engineers Standards Website: https://standards.ieee.org/standard/15288-2015.html

[2] Barry Smith, Rob Arp, and Andrew Spear, Building Ontologies with Basic Formal Ontology, MIT Press, 2015

[3] ISO/IEC 21838-1, Information technology—Ontologies—Top-Level Ontologies, International Organization for Standardization website: https://www.iso.org/standard/71954.html (accessed 09 September 2018).

[4] ISO/IEC CD 21838-2 Information technology -- Top-level ontologies -- Part 2: Basic Formal Ontology (BFO) International Organization for Standardization website: https://www.iso.org/standard/74572.html (accessed 09 September 2018).

[5] Alexander P. Cox, Christopher K. Nebelecky, Ronald Rudnicki, et. al. The Space Object Ontology, 19th International Conference on Information Fusion (FUSION), July 2016. Available at the IEEE Digital Library https://ieeexplore.ieee.org/document/7527882. Accessed 18 September 2018.

[6] Rudnicki , Ron. “Common Core Ontology.” GitHub, 20 Feb. 2018, https://github.com/CommonCoreOntology/CommonCoreOntologies .

[7] Madni, Azad M. "The Intellectual Content of Systems Engineering: A Definitional Hurdle or something More?" INSIGHT 9.1 (2006): 21-23.

[8] Blackburn, Mark, et al. Transforming Systems Engineering through Model Centric Engineering. Stevens Institute of Technology Hoboken United States, 2018.

[9] Honour, Eric C., and Ricardo Valerdi. Advancing an ontology for systems engineering to allow consistent measurement. 2006.

[10] Obrst, Leo. "Ontologies for semantically interoperable systems." Proceedings of the twelfth international conference on Information and knowledge management. ACM, 2003.

[11] Graves, Henson, and Matthew West. "Current State of ontology in engineering systems." Internet: OMG (2012).

[12] Petnga, Leonard, and Mark Austin. "Ontologies of time and time-based reasoning for MBSE of cyber-physical systems." Procedia Computer Science 16 (2013): 403-412.

[13] Jenkins, J. Steven, and Nicolas F. Rouquette. "Semantically-Rigorous systems engineering modeling using SysML and OWL." (2012).

[14] Bermejo-Alonso, Julita, et al. "An ontological framework for autonomous systems modelling." International Journal on Advances in Intelligent Systems 3.3 (2010).

[15] Van Ruijven, L. C. "Ontology for systems engineering." Procedia Computer Science 16 (2013): 383-392

[16] Sarder, MD B., and Susan Ferreira. "Developing systems engineering ontologies." System of Systems Engineering, 2007. SoSE'07. IEEE International Conference on. IEEE, 2007.

[17] Siang Kok Sim and Alex H B Duffy “Towards an Ontology of Generic Engineering Design Activities” Research in Engineering Design 14 (2003) 200–223, DOI 10.1007/s00163-003-0037-1