Information Models and Ontologies to Enable Digital Engineering

RESEARCH WORKSHOP
MAY 23-24, 2023

Sponsored and Organized by:
the Systems Engineering Research Center (SERC) and MITRE

UNCLASSIFIED
DISTRIBUTION STATEMENT A: Approved for public release. Distribution is unlimited.

Table of Contents

EXECUTIVE SUMMARY

The Systems Engineering Research Center (SERC), a University Affiliated Research Center (UARC) for the U.S. Department of Defense (DoD) and MITRE hosted a research workshop on Information Models and Ontologies to Enable Digital Engineering at MITRE on May 23-24, 2023. This in-person workshop gathered key stakeholders and experts from government, federally funded research development centers (FFRDCs), other UARCs, industry, and academia to drive the discussion, propelled by targeted, invited talks focused on promising approaches to design and implement ontologies and existing bodies of knowledge to support the development of ontologies. A total of 66 participants attended this invitation-only workshop.

The objective of this workshop was to first understand the current state of practice and research in the synthesis and application of information models and ontologies in support of Digital Engineering; and second, identify a small number of key research questions that need to be addressed. This was driven by the notion of achieving semantically rich design reasoning across abstraction levels and disciplinary domains. Four invited talks set the stage for this workshop followed by three working groups and a concluding summarizing session at the end of the workshop.

The topics of interest included:

- Capabilities enabled by ontologies in the context of Digital Engineering
- Internal consistency in nomenclature, concepts, and practices across an extended enterprise, and a complex supply chain
- Tension between theoretical correctness and pragmatic implementation of ontologies and information models
- Barriers and enablers to develop and adopt ontologies
- Existing or proposed initial taxonomies in support of Digital Engineering.

Outcomes

The workshop revealed that much of the work being conducted in one DoD organization or another on ontology development and implementation, especially in the applied computational ontology sense, was not widely known by the other organizations. The workshop consequently highlighted a need to convene smaller, focused groups with DoD movers in this space to begin aligning priorities and objectives.

The workshop breakout groups found gaps in our understanding and practice across the following three major areas, which are described in further detail under the Conclusion section of this report:

1. Foundations for applied computational ontology development, evolution, and use across real DoD programs
2. Defining and describing the value of applied computational ontologies to Digital Engineering practice
3. Critical dimensions for operationalizing applied computational ontologies and transitioning them to scale.

INTRODUCTION

The Systems Engineering Research Center (SERC), a University Affiliated Research Center (UARC) for the DoD and MITRE hosted a research workshop on Information Models and Ontologies to Enable Digital Engineering at MITRE on May 23-24, 2023. This in-person workshop gathered 66 key stakeholders and experts from government, FFRDCs, other UARCs, industry, and academia to drive the discussion, propelled by targeted invited talks focused on promising approaches to design and implement ontologies and existing bodies of knowledge to support the development of ontologies.

The workshop objective was to first understand the current state of practice and research in the synthesis and application of information models and ontologies in support of Digital Engineering; and second, identify a small number of key research questions that need to be addressed. This was driven by the notion of achieving semantically rich design reasoning across abstraction levels and disciplinary domains.

Workshop Agenda Structure

Four invited talks set the stage for this workshop:

- **How to – and How Not to – Build Ontologies: The Hard Road to Semantic Interoperability**
  Mr. Chris Partridge, *Chief Ontologist, BORO Solutions, UK*

- **Armaments Digital Engineering Strategy – An Ontology Based Approach**
  Drs. Jason Cook and Eddie Grimes, *U.S. Army Futures Command, Armaments Center*

- **Experiences Using Formal Ontologies in Systems Engineering Practice**
  Dr. Steve Jenkins, *NASA Jet Propulsion Lab (JPL) (Retired)*

- **A Digital Engineering Methodology for Interoperability Using Ontologies**
  Drs. Mark Blackburn and Tom Hagedorn, *Systems Engineering Research Center (SERC)*

Two talks occurred each day and were followed by three working groups and a concluding summarizing session at the end of the workshop. Each working group was moderated with an interactive discussion. For the first day, the three working groups each focused on a different topic:

1. **Ontology Development:** *Best practices to develop an ontology: what is needed, what is good enough, what are critical challenges?*
2. **Ontology Evaluation and Evolution:** *What information/knowledge is best when developing ontologies; where can we find this knowledge, what are potential inconsistencies and/or likely conflicts, what are the existing harmonization initiatives?*
3. **Piloting Ontologies:** *What are the potential low impedance pilots to showcase the value of ontologies?*

For the second day, the three working groups each focused on a common theme, anchored on the concepts of **transitioning to and operationalizing digital ontologies** at scale:

- What **working model and policy considerations** could support applied computational ontology development and evolution?
- What are the primary considerations with respect to policies for **governance** in terms of what is needed, what needs to be developed, and where these aspects might live?
- What are the needs concerning **workforce development** to support applied computational ontology development and implementation?
- What are the critical aspects across the technical, cultural, and leadership dimensions required for successfully **operationalizing applied computational ontologies and transitioning them to enterprise scale**?

Workshop Opening Introduction

The workshop began with an overview of the SERC and its mission, especially as its related research thrusts seek to enable Digital Engineering. Ontologies, specifically as they enable the effective and efficient advancement of Digital Engineering, are key enablers of semantic interoperability for computationally-based data synthesis, analysis, and exploration. The need for interoperability across Digital Engineering efforts extends past traditional engineering into Acquisitions, Test and Evaluation, and the body of Evidence-Based Decision Making across a system's lifecycle. Traditionally, this has been viewed as solvable via automated data sharing based on a common exchange model. Such syntactic interoperability is vital, yet insufficient. We also need the semantic interoperability provided by ontologies.

At a high level, ontologies provide a common vocabulary for representing and organizing data within a domain and define the meaning of the concepts and relationships within that domain. This allows different systems and applications to share and understand data consistently and unambiguously.
An ontology models generalized data, that is, we take into consideration general objects that have common properties and not specified individual entities. As a general data model, an ontology should be reusable framework, where an upper-level ontology describes general concepts and relations and a domain ontology describes concepts and relations in a particular domain.

As highlighted frequently by one of the invited speakers, Mr. Chris Partridge, the value of ontologies lies in what they allow us to communicate. For Digital Engineering, this translates into what data and information ontologies allow us to synthesize, compute on, and reason about. Accordingly, as the focus of this workshop, **digital ontologies for data and model interoperability** will support the practice of Digital Engineering by enabling several key capabilities:

- Integrating data and information across disparate data, information artifacts, models, and disciplines
- Enabling <u>syntactic</u> and <u>semantic</u> interoperability

- Providing the foundations for data and information that are machine-interpretable, readily extensible, and human-understandable.

Ontology-enabled methods for Digital Engineering allow us to make inferences on data and information, that is, to determine new facts, discover previously unseen gaps and relationships, etc. based upon the existing ones in a logical fashion. They also make it possible to uniquely identify data elements so that different data systems can refer to the same concepts without having to pass around or duplicate unwieldy data structures. These "**applied computational ontologies**" (i.e., distinguished from purely philosophical ones) are a pragmatic means to formally describe the relevant entities and relations of a system, implement those descriptions and relationships, and be useful for our specific purposes and objectives.

While it is not uncommon or exceedingly difficult to establish a digital ontology for a given effort, such a point solution will not create an interoperable foundation extensible to other efforts or domains. There are actually several ontology formalisms in existence, with a few in widespread use. While a diverse set of ontologies would seem to offer a readily available "best fit" selection ready to implement, this is actually one of the challenges. Different ontologies express their commitment, or how they see the world, in different ways. In turn, this significantly impacts which ontologies may be the best fits for further development and use in different types of domains.

The invited talks for the workshop were selected to provide a diverse set of views informed by an equally diverse set of experiences with ontology development, implementation, and use in a Digital Engineering context. In turn, these talks exposed the attendees to varied perspectives and lessons learned prior to each day's breakout sessions.

Notably, participants in the workshop tended to use the terms "applied computational ontology" or "digital ontology" interchangeably to mean ontologies developed and implemented for the purpose of achieving data and model interoperability across Digital Engineering and digitally transformed system engineering practices.

FEATURED TALK DAY 1, TALK 1: A DIGITAL ENGINEERING METHODOLOGY FOR INTEROPERABILITY USING ONTOLOGIES

This talk discussed a Digital Engineering (DE) methodology for interoperability using ontologies to support mission, system and discipline specific analyses. The use of ontologies is to support domain ontologies such as armaments as discussed in the fourth talk by our Army sponsor. The described approach uses ontologies and semantic technologies to model data in a tool agnostic environment.

Most people use databases every day, but very few people ever make or program the database. Accordingly, we have developed a unique method that effectively hides the details of the underlying ontologies from a typical user by leveraging descriptive models as a mechanism to "program" the ontology aligned data. In a systems engineering context, mission, system, subsystem, discipline-specific and physics-based objectives and parameters are described using tools like Systems Modeling Language (SysML). In addition, the SysML model defines the parameters needed to characterize the modeled mission. A parametric diagram called the Assessment Flow Diagram (AFD) explicitly describes the relation between these parameters and interfaces for various types of external simulations (e.g., geometry, ballistics, aerodynamics).

The Armament Interoperability and Integration Framework (IoIF) then loads the SysML model, with the AFD effectively used to "configure" IoIF. Based upon the AFD's structure and annotation, IoIF affords a request interface for packages of system or mission data aligned to the specific needs of a given model. These remain linked to the mission and system attributes described in the SysML model and are accessible via both the programmed and query interfaces. When used in a workflow, the IoIF continuously uses and updates its ontology-aligned data repository with outputs generated by the analyses described in the AFD. The workflow produces trade-off values for the mission and system-level objectives.

The talk concluded with a demonstration of the workflow, including the digital thread dashboard that can be used to look at different parameters associated with the trade space objectives, and also demonstrated a decision framework dashboard that allows for the visualization of the mission and system tradeoffs.

FEATURED TALK DAY 1, TALK 2: HOW TO- AND HOW NOT TO- BUILD ONTOLOGIES: THE HARD ROAD TO SEMANTIC INTEROPERABILITY

This talk looked at the digitalization journey that takes one to semantically seamlessly interoperating enterprise systems. A journey that is a hard road to travel – particularly at the later stages – when an ontology is deployed. The journey is difficult (hard) because it requires the adoption of radical new practices.

The talk provides a useful way of understanding these radically new practices through a cultural evolutionary perspective. It explains how this perspective naturally raises important and broad concerns.

In provided examples related to evolutionary contingency, it seems clear that if we don't adapt in the right way, we will not evolve interoperability. The talk noted that we have some idea of what the practices are, what the trajectory of the journey is, but that this is not sufficient for evolutionary success. Further, the community also needs to find the means to (horizontally) inherit these aspects.

The talk highlighted the radical new practices that need to be adopted along the journey, describing three intertwined headings:

- Shifting scope
- Evolution
- Economics of complex accuracy.

Underneath shifting scope, four main practices were described: 1) the lifecycle separation of concerns approach, 2) how to manage form by either managing the formalization process or taking control of the form, 3) the role of a top ontology, and 4) the need to work with *all* of the data. With respect to evolution, the concepts of directed evolution and factoring followed by refactoring were discussed. Finally, with respect to the economics of complex accuracy, the talk discussed the practices of running a paradigm shift factory and scaling down to scale up.

In summary, the talk highlighted that the road to semantic interoperability is hard because it requires the inheritance of a set of radically new practices. Moreover, we are presently working on developing these new practices and, in the future, we will need to shift from practices to a more organized, inheritable praxis.

BREAKOUT WORKING GROUPS SUMMARIES (DAY 1)

Group 1: Ontology Development

The Ontology Development working group started the session with the following questions:

With respect to best practices to develop an ontology:

- **What is needed?**
- **What is good enough?**
- **What are critical challenges?**

Precursors for Ontology Development

The group first identified precursors to ontology development necessary to lay the foundations for successful development and implementation. Namely, a team needs a formal plan for the "journey," i.e., the ontology lifecycle of existing ontological basis selection, further development appropriate for the domain or domains needed, identifying methods for formalization and computational implementation, and proper attention to the synergy between syntactic and semantic interoperability. Examples of previous efforts in related applications can offer guidance and a means to evaluate what success might look like. The entire ontology lifecycle as it complements Digital Engineering activities needs to be operationalized via properly defined and scoped processes; this view is highly concordant with systems engineering.

The group also expressed that it is critical for an organization to have a policy in place regarding ontology development, expansion if needed, and use. This will require professional policymakers to work closely with the ontologists to craft effective and grounded guidance. Context is a critical aspect. The group identified several additional factors important for long-term success: the need for use cases for an ontology, that any ontology and its expression as a living computational artifact should be tool agnostic but also based on a format that supports further development, the need for clear and consistent definitions and a commensurate model of terms, and that an ontology should be specially formulated for an ability to be governed by policy.

Paradigms for Good Practice

The group then sought to answer the question "what is a good practice to determine that ontology and what are the barriers?" The group identified three paradigms, which became the leading topics of sub-group discussion to further explore opportunities and risks: classes, instances, and common core.

In ontologies, a class (or concept) is group of individual entities or "things" that share common characteristics used in a wide sense. In a domain ontology, classes describe concepts in the domain; these can be functions, tasks, processes, physical entities, etc. (In contrast to object-oriented programming, which is concerned with methods on classes where design is contingent on operational properties of a class, ontologies focus on structural properties of a class.) The group discussed the differences in starting from the bottom-up approach versus developing from the top down. An alternative is to find the best synergy in using higher-order SME choices and base data (e.g., from asset systems where data are stored) simultaneously. The group felt that generality should be balanced with parsimony when defining classes, and that a key risk is when schema does not follow the actual data (or, conversely, data does not end up following the schema).

Where an ontology is a conceptualization of a domain, instances add data to that conceptualization. Instances are the specific "things" represented by a class. An ontology together with a set of individual instances of classes constitutes a knowledge base. The group felt strongly that instances should be based on reality and that the problem space for a given implementation of an instance set with an ontology should be well-bounded. The group further discussed risks associated with overfitting to the instances we have (i.e., we need a lot of instances) and having an incomplete representative instance set, which can lead to solving the wrong problem.

The group discussed the need for a core ontology for the US Department of Defense (DoD) because the DoD language is unique and specialized. Moreover, all the Services (e.g., Army, Navy, Marines, Air Force, Space Force) as well as non-service specific DoD groups (e.g., OUSD (R&E), Combatant Commands, etc.) have unique policies and needs. Maturing a core DoD ontology will further our collective ability as a community to establish governance and structural hierarchy. Several challenges associated with developing a common core at a high-level for DoD purposes were identified: attaining proper modularity of the framework, that systems engineering itself does not have a common core, the need to ensure consistency with a top-level ontology, that this approach assumes or may assume that all derivative ontologies have one parent, and defining how the common core ontology could be grounded and tested.

Ontology Development and Design Considerations

The group identified several barriers to successful ontology development. First, it can be quite difficult to communicate precisely what an ontology is and what it will do that is differentiated from existing taxonomy and syntactic interoperability concepts. Often the term ontology is used to mean very different things by different organizations. This can create challenges when communicating the need for development to key decision makers who need to fund the effort; ontology discussions are often not the most exciting and feel much less tangible and actionable for non-ontologists. It is important to think about the return on ontology development for Digital Engineering success and be able to communicate that return.

These concepts relate to the challenge of developing consensus across and enterprise such as the DoD or even an individual Service with respect to how to approach ontology development and use. Meanwhile, in SE there are cases of success that can be adopted. A consensus-based approach and consortium model was suggested, similar to that found in other areas of systems engineering, wherein a stakeholder group gets together to validate an ontology or set of ontologies that are holistic and accurate. These observations relate strongly to the discussions from the Day 2 breakouts.

With respect to other design considerations, the group asked whether an ontology could offer a competitive advantage and how to establish a balance between competition and cooperation. There are many different drivers behind ontology development and use, and these drivers will strongly influence what can be shared (e.g., security considerations) and how each ontology could fit as part of a larger solution space and facilitate reuse. Use cases will be important to help address these aspects. It is also important to think about paradigm shifts in ontology development; as each stage grows, the size does not grow but rather the functionality increases. Appropriate manpower with the time and ontology-related skills are also needed.

The group also discussed that, for traditional engineered systems, sustainment is a "tax." For ontologies, should the focus therefore be design and maintenance or rather continuous design? Reference and domain ontologies tend to be stable across time (at least historically so far, rapid leaps in technology and its capabilities could impact this assumption). It is possible that if a given ecosystem is stable, then one can build context in later.

One group member stated that they found there are ways to organize the world that don't have to be reflected in ontology and this relates to "ontological realism" – being realistic to the point of being pedantic. Due diligence is needed up front; there are development principles that can be used to guide and constrain development of digital ontologies. We also need to accept that there will be continuous change and an ontology should not commit to a "moral framework." Science is a good role model for this and expectations for development are important.

The group concluded this discussion by highlighting the importance of evaluation and finding consistent and effective ways to evaluate ontologies. Key metrics might stem from basic questions such as "can we migrate from this to that?". The group felt that ultimately, trying to determine the right measures would be significantly more difficult than finding the right ontology; we simply do not know how to adequately assess "goodness" yet.

Group 2: Ontology Evaluation and Evolution

The Ontology Evaluation and Evolution working group started the session with the following questions:

- **How do we evaluate an ontology in terms of consistency, correctness for the domain, etc.? Is this consistent across or relative to the domain and problem space?**
- **How do we enable effective extensibility and synthesis across multiple domain ontologies, and still preserve these aspects?**
- **How do the needs for communication and computation drive the needs for ontological development and measures of effectiveness?**

Consistency was discussed in terms of consistency across development efforts, domains, organization, etc. Concepts like internal mathematical consistency were only briefly addressed. Thus, the need for consistency was aligned with aspects like reuse, cross-domain reasoning, and consensus building.

A main point of discussion was the need for intentionally developing the questions we expect ontology-aligned data to answer. One participant noted the knowledge stream in his organization as a model for starting with the question the knowledge graph needs to answer. All development efforts beyond the identification of the question were directed, or "purposeful," toward answering the question. The effort to limit ontology development and tools surrounding the ontologies to specific, useful questions resonated with the group. Other related comments included:

- What is the right "1st question?"
- What are the key use cases?
- What data should we be collecting related to the central questions?
- Is there a maturity metric?
- How precisely was the question answered?
- How consistently is the question answered?
- Are answers to the question machine discoverable?

While there was general consensus around developing ontologies and knowledge graphs with questions in mind, this approach was balanced with the idea of reuse across projects, organizations, and domains. In discussing this, the point was emphasized that we must understand what questions we are allowed to ask the knowledge graph. For example, an excellent aircraft ontology may have been developed for a previous project and available for reuse on a current project. This ontology may be excellently suited to answer questions related to mass, propulsion, or ground communication, but it may be ill-suited to answer questions of airworthiness as it was not designed to answer that question. Consistency and interoperability of form (e.g., common use of a top-level ontology) is only a portion of the effort; proper understanding of purpose and limitations is also key.

The group also discussed implementation and the need for metrics to assess consistency and correctness. For implementation, there was discussion on how closely frontline engineers should be expected to interact with underlying ontologies and how much they should understand the ontological commitments being made versus the toolset and user interfaces abstracting these concepts away from some portion of the design and analysis community involved. In general, tools are needed to aid in use of ontologies and ensuring consistency across ontologies.

Extensibility refers to the ability in ontologies for the data model to evolve. This ability relates to how reusable an ontology is – if it can only answer the question it was designed for and not expanded to include other questions, it has limited reuse potential. There was discussion on balancing usability and extensibility. There is a view that these two metrics are often at odds.

Discussion of synthesis was around top-level ontologies and approaches that may not require total synthesis. There was discussion of how to properly map ontologies that have been developed using different top-level ontologies. A research question that needs further inspection is what types of synthesis are possible between top-level ontologies and what portions of ontologies may be completely incompatible. An approach that does not require complete synthesis of data is model transformations that allow for selected transformation to a common form without a full mapping.

One approach to synthesis and consistency is engaging with tool vendors to develop curated ontologies with interoperability and consistency in mind. A curation group was discussed. Development of non-proprietary ontologies may drive tool vendors towards a common approach. Alternatively, proprietary joint efforts with limited sharing may be further investigated. In relation to the tool vendors, it was generally agreed that a certain mass of practice needs to be present to create the demand signal needed for tool vendors to adopt.

One point that came up was the focus on efficiency: is the main purpose of using ontologies to increase efficiency? There are efficiency advantages of Digital Engineering and ontologies, be that efficiency in design and analysis time cost, model runtime, data cost, search time, etc. However, it was noted that too singular a focus on efficiency can "smuggle in risk." Thus, in looking at the evaluation of ontology use, increase in efficiency should be balanced with other views of quality of design and reasoning efforts.

Group 3: Piloting Ontologies

The group brainstormed piloting opportunities in Digital Engineering ontologies by considering the Digital Engineering problems that need to be solved using ontologies to succeed. These problems often reside across the spectrum of Digital Engineering activities (e.g., digital acquisition, testing and evaluation, etc.) and require data-drive decisions across the activities. The group suggested the following pilot opportunities that could showcase the value of ontologies:

1. The Adaptive Acquisition Framework (AAF) is an authoritative set of acquisition pathways to enable the workforce to customize strategies to deliver better solutions faster. The pilot could

facilitate efforts from reliability or maintainability to product support. The group proposed conducting a small pilot first to develop the ontologies and demonstrate the utility to prove the effectiveness of ontologies.

2. The Electronic Warfare (EW) Office at OUSD wants to address cross-use effects across the DoD Services. For example, capability and requirements as well as their implications may mean something different for the Services. This could be a potential opportunity for piloting ontologies.
3. The Integration Acquisition Portfolio Reviews (IAPR) through mission engineering could be an example for a pilot. Mission-level portfolios are not including ontologies and Digital Engineering. In a joint mission example, a use case could be moving fuel in a contested space (e.g., host nation resources and logistics issue) where various DoD Services have a shared need for accessing the same fuel.
4. Resiliency ontology was also discussed. What does it mean when something degrades, fails, and recover? One example is the energy sector. What is the robustness in terms of cost, schedule, performance, and risk?

Current efforts in developing ontologies seem to have trouble connecting across communities or functions, and ability to assemble the right people to demonstrate a pilot in 6 months. Some gaps discussed include the scale (e.g., tagging in SysML 2, expertise in ontologies, etc.), validation and verification of the ontologies, cross-ontology interoperability, and education (exercise versus application and use in real-world). There needs to be a focus on leveraging artificial intelligence (AI) and machine learning (ML) for computational resources and uses, determining when not to have ontologies, and the need for unmanned design.

The benefits of ontologies include defining handoffs with stakeholders throughout the lifecycle, including technical specifications, software diagnostics, design elements or consideration for sustainment, and to prove value through metrics. Some of the metrics include minimizing risk early in the lifecycle, looking across domains for solutions, operation availability, and mean time between failures (MTBF). Using ontologies, historical perspectives could be gained by communicating the changes through time. Pilots are important to prove value to a decision maker that this is an important expenditure and effort for the organization.

FEATURED TALK DAY 2, TALK 1: EXPERIENCES USING FORMAL ONTOLOGIES IN SYSTEMS ENGINEERING PRACTICE

(The speaker began by clarifying the remarks and opinions stemmed from his personal experience over years of work in this and related fields and did not represent the position of the Jet Propulsion Laboratory, California Institute of Technology, the National Aeronautics and Space Administration, or any other person or organization.)

The talk began with a discussion of early efforts and lessons learned from the speaker's background in system engineering ontologies at the Jet Propulsion Laboratory (JPL), which included experimenting with formal Model-Based Systems Engineering tools like CORE (Vitech, Inc.) around 2000 and learning about the Semantic Web and W3C Web Ontology Language (OWL). This work found that it was fairly straightforward to represent fundamental concepts of systems engineering (e.g., Component, Interface, Junction, Function, Requirement, etc.), their data properties (e.g., name, identifier, description, etc.), and their relations (e.g., contains, presents, performs, joins, characterizes, etc.) in OWL. The team at the time demonstrated the ability to express a broad range of mission and system information in a standards-based, tool-independent form, to retrieve it using standard protocols, and construct multiple mutually-consistent viewpoints on the same underlying knowledge. That is, of course, the essence of systems engineering.

This work continued as a loose series of prototypes and demonstrations to develop improved and more sophisticated understanding, eventually helping lead to a multi-year investment program called Integrated Model-Centric Engineering (IMCE) in 2008. Under the auspices of IMCE, the team began a multi-pronged effort to develop formal vocabularies in three categories:

- Foundation: core concepts of Systems Engineering such as Component, Function, Interface, Requirement, Quantity, etc.
- Discipline: discipline-specific specializations and extensions, such as electrical, mechanical, telecommunications, etc.
- Application: relating to the mission or specific application area.

The work also sought to configure commercial off-the-shelf MBSE SysML tools to assist (or enforce) building models that conformed to the constraints in the vocabularies, and also develop analysis and reporting tools. These latter elements sought to create standard and *ad hoc* engineering products by extracting the relevant data from these models.

The integrating vision was a recognition that the primary product of systems engineering is *knowledge* and that the team's methodology and processes should be defined and evaluated according to their production of trustworthy knowledge. The packaging of that knowledge into specific artifacts (e.g., reports, presentations, documents, etc.) is important, but can be largely automated if the underlying knowledge is sufficiently rich.

Applying these principles, the team discovered that it was challenging for several reasons, not merely technical, to build semantically-rigorous models with SysML. SysML version 1 and commercial tools that implement it are, by design, permissively expressive. This feature is attractive to people and institutions beginning their journey to model-based engineering, but it was posited that using a modeling tool to make semantically-ambiguous or vague *ad hoc* descriptions of complex things misses the point of modeling entirely.

CAESAR and OML

Over the years, the idea of developing modeling tools tied more closely to the methodology the team was employing, and vocabularies became gradually more acceptable. Then, the electrical engineering team on Europa Clipper began a joint development effort with IMCE. One of the key products needed were the specifications for the spacecraft wiring harness, a complex piece of hardware that provides electrical power and signal connections among thousands of interfaces. To accomplish this, the electrical engineering team describes what the harness must do, including end-to-end characteristics (e.g., acceptable voltage drop), and a team from the Mechanical Systems Division designs and fabricates the device itself. Until that point, the electrical engineering team had done most of its work with a spreadsheet that had been copied and modified from one project to the next. Like most spreadsheets, it provided little input validation and was easily contaminated by undetected errors. Consequently, the team had to devote considerable time to manual validation of results.

In its partnership with the electrical engineering team, the IMCE team built an integrated modeling environment that provided a familiar spreadsheet-like user interface, but with an underlying data model based on OWL ontologies and featuring verification of consistency and automated extraction of entailments. While it may seem extravagant to develop new tools when a number of commercial offerings existed, the development consisted primarily of a relatively thin layer of adaptation atop open-source software including Eclipse Sirius, Apache Jena, Docker, Kubernetes, and others. Much of this adaptation would have also been required had the team adopted commercial modeling tools. There is a wide gap between a general-purpose tool and any organization's specific engineering practices; it is often overlooked by people who think they can procure a COTS solution to *their particular* problems.

The team called the tooling CAESAR. Although its first application was for harness design, CAESAR is in no sense a harness design tool but rather a general systems engineering environment adapted to specific applications and viewpoints by augmenting the core engines with viewpoint specific vocabularies, user interface widgets, and analysis and reporting code.

As the team gained more experience with modeling in OWL, they developed a set of modeling patterns to best serve their purposes. For example, they employed two distinct uses of the notion of *class* in OWL: one for things like Component and Requirement for which there is an intended disjointness (a thing cannot be both a Component and a Requirement), another for things like NamedElement and PerformingElement, for which it is perfectly reasonable for a thing to be both. A software library was developed that implemented a higher-level language that encapsulated these patterns; the team called it *Ontological Modeling Language* (OML). OML has no distinct semantics; it is simply a convenience layer that maps onto a restricted sublanguage of OWL 2 DL.

Around 2019 the team enhanced CAESAR to support steady-state power analysis for Europa Clipper. This analysis largely consisted of adding power mode characterizations to hardware assemblies already present and used for harness design in the system model. Other information in the model (e.g., the compositional structure of assemblies, grouping into subsystems, etc.) was simply reused. This reuse was a core design principle of CAESAR: the world as we know it is represented in the model more or less independently of the uses of that knowledge. That, in turn, makes it easy to use the same knowledge (e.g., the system breakdown structure) to support analysis in multiple distinct viewpoints.

The practitioners in both harness design and power modeling were enthusiastic adopters of the new approach. Also, however, success in these two applications has not led to any direction from management to coalesce around and expand the effort. Tactical success is acknowledged but at present the vision for future success remains tactical. The grand vision of IMCE – a rigorous, knowledge-based engineering methodology backed by powerful modern tooling – remains unachieved. The knowledge-based tooling is there and delivers what it promised. The next step, if it is to be attained, will require a different way of thinking among decision-makers.

On the basis of these experiences, the speaker drew several observations relevant to the workshop focus. Firstly, that ontologies are not a new solution to a new problem. They are simply a somewhat more rigorous, somewhat more expert approach to the fundamental problem of systems engineering, i.e., to bring together multiple disparate viewpoints in order to see a system as a unified whole in such a way that we can be confident it will achieve its mission. The speaker explained that this problem exists whether we acknowledge it or not, and so the only question is whether or not we will we solve it well.

Secondly, ontologies are not about defining terms. Definitions explain words in terms of other words, leading to infinite regress. The speaker posited that a better way to think about ontologies is that they establish the grammar for a language understood by a community. Systems engineers don't necessarily need to know what a Requirement *is* in order to agree on meaningful things they can say about a Requirement, e.g., it has a number, it has a title, it constrains some quantities of a system under certain scenarios, etc. Rather, it is much more productive to agree on the meaning of such sentences than on the terms they employ. For example, the sentence "I took my car to the shop" is clear enough without precise definitions of what exactly can and cannot be a *car* or a *shop*. The sentence is a signifier for a familiar situation; that's what makes it meaningful.

Relatedly, theory and rigor are fundamental to all engineering, including systems engineering, and are often the only path to practical solutions of hard problems. The speaker observed that people who disparage theory and rigor as somehow impractical are chasing illusions. If something is impractical, it is impractical. Theory is not what made it impractical.

Moreover, any number of notorious engineering failures (e.g., Apollo 13) can be directly traced to *knowledge failures*, in which some important facts are known to some parties but not to others who require them. Formal ontologies cannot prevent knowledge failures, but the lack of formal ontologies practically guarantees them.

Finally, the word *modeling* is used commonly in systems engineering, but without any firm notion of what it entails. The speaker suggested that modeling simply means (1) the use of precise language with consensus meaning, (2) the use of mathematical abstractions (e.g., differential equations) to empower analysis, and (3) the use of automation to perform the sorts of tasks (and there are many) at which computers outperform humans. One key virtue of this characterization is that it suggests an immediate challenge to those who proclaim skepticism about modeling, namely, to make a case *against* precise language, mathematical abstractions, and automation.

FEATURED TALK DAY 2, TALK 2: ARMAMENTS DIGITAL ENGINEERING STRATEGY – AN ONTOLOGY BASED APPROACH

The Combat Capabilities Development Command (DEVCOM) Armaments Center (AC) seeks to ensure cutting edge technology is available for the US Army and that US Army Soldiers always have a competitive advantage. Accordingly, the DEVCOM AC also seeks to lead the field of systems engineering (SE) and has invested in Model Based SE (MBSE) and Digital Engineering (DE) for many years. The results have provided value to projects and programs and resulted in numerous lessons learned.

This presentation discussed those lessons learned and how they inform the refined strategy the Armament Center has developed and is implementing in partnership with the Joint Program Executive Office for Armaments and Ammunition (JPEO A&A). The strategy begins with a Digital Engineering Maturity Model (DEMM) developed by DEVCOM AC to clearly measure the current state and to describe our goal state. Originally developed for internal use, the DEMM has proliferated in use. To date, it has been used to assess organizations across the three major commands in the Army material development space, Army Futures Command (AFC), Army Materiel Command (AMC) and the Assistant Secretary of the Army for Acquisition Logistics and Technology (ASA-ALT).

Further, the strategy includes development of detailed use cases to derive the functional and integration requirements for our future DE Ecosystem (DEE). This effort utilizes MBSE best practice, starting with stakeholder use cases, creation of associated activity diagrams to model the data flow through that use case, and then derivation of requirements. The process is repeated for each use case until a complete requirement set is obtained with traceability to the stakeholder use cases. This enables a more robust evaluation of alternatives, as we are then able to describe our integration and interoperability requirements to the data element level. Third, the strategy includes a robust Analysis of Alternatives (AoA) and evaluation of commercial and open-source candidate solutions. DEVCOM AC has developed a set of evaluation criteria, derived from the afore mentioned requirements, also included are various non-functional criteria such at information assurance constraints and life-cycle costs. These are used to evaluated options that look promising.

After the evaluation, we have selected four solutions to further explore. To do so, we instantiate that solution in a development environment, train our own team on the use of that tool set and execute three specific use cases with a representative set of system data. This hands-on exposure to the tools being considered has proved much more informative than any information provided by the vendors.

Finally, and foundationally, the strategy involves the development of a formal computationally-enabled ontology, being adopted by Army and OSD. DEVCOM AC has undergone an effort to build a formal Military Systems Ontology to capture the relevant concepts, properties, and relationships relevant to our systems, including their development, manufacture, and use. This has required development of a mid-level ontology and various domain level ontologies that are extensible across the DoD and the defense industry. As a result, leaders at Department of Defense and Army level have adopted this framework and championed efforts to extend the work. For DEVCOM AC, the ontology is expected to enable a Semantic Web Technology (SWT) based DEE, which is what we've identified as our objective solution.

BREAKOUT WORKING GROUPS SUMMARIES (DAY 2)

Group 1

The discussion was highly activity driven as participants wrote ideas on each topic for discussion to identify specific concepts they felt would help or hinder ontology development and use across a set of related guiding questions:

- **What is needed for workforce development, whether for ontology development and implementation or ontology-based engineering?**
- **What will it take to transition ontology use to applied practice at scale? What are the important technical, cultural, and leadership dimensions necessary to operationalize ontologies?**
- **What is needed in terms of ontology governance and policy? How can policies enable or disable operationalizing ontologies?**

<u>Workforce</u>

When people talk about understanding ontology, one type of necessary understanding is operational. There is a difference between formal and operational understanding. The group felt that engineers do not need to understand ontological formalization, but they do need to know what is meaningful to communicate in the scope of interactions relevant to a domain in question and the mental models – shared or not – that this implicitly reinforces.

The group also identified context as an important factor that would influence the level and type of workforce development necessary for operationalizing ontologies for Digital Engineering. Pressures of cost versus available funding and time are important aspects to balance all while striving to build in layers of abstraction to represent knowledge. Suggestions for managing these conflicts included having specific, task-focused work that would breakdown efforts into levels of manageable complexity. To support the workforce, it will be necessary to define additional levels of understanding based on a specific task, how an engineer will know when it is done well, and when it can be handed-off to the next level. The value of labor must be easier to access for non-practitioners; comprehensive stakeholder engagement is needed. The group also pointed out the need for avenues for the workforce to ask questions and receive feedback, such as through a community of interest or educational commons established at an organization such as the Defense Acquisition University (DAU).

<u>Transition</u>

Multiple participants pointed out the difficulty in sustained, effective development needed to mature ontology development and use when Service officers and decision makers rotate to a new position every 18-24 months. This does not incentivize risk and change; they are graded on what is accomplished on the funding available in their time period on station and not how well their efforts set the program or office up for success in five years' time. This is compounded when these decision makers do not have the workforce already in place to make operationalizing digital ontologies practical and effective.

The group felt that DoD leadership needs to be open to how well we incentivize ontology development, governance, evolution, and bringing tool vendors to the space to support Digital Engineering. Relatedly, it will be vital to articulate the value of operationalizing a digital ontology and tangibly define value-based results that could not be achieved without the ontological basis.

Governance and Policy

The group discussed that ontologies are a means to an end and so we need to be mindful of this when considering policy. However, the group felt that policy should provide the basis for communication agreements and how different organizations could "liberate" knowledge from artifacts to facilitate sharing and re-use.

Crafting policy also frequently leads to conflict between those who want freedom with respect to how to approach these aspects and implementation and those who want specificity defined. Different Services, organizations with a Service, and communities within the workforce all have very different preferences, which challenges how specific a common policy can be. The group felt it was most likely that OSD would not be overly prescriptive and restrictive, but rather each Service would develop their own policy guidelines and constraints. Ideally, these policies would define the what and not the how with respect to ontology development, governance, and use.

An important consideration is that Digital Engineering is goal-based; Services will need to decide if they want to be interoperable both syntactically and semantically. A digital innovation group exists at the Federal level, for example, but adoption of digital transformation and its nature is very Service-specific. High-level policy talks about data, but there is no enforcement to deliver that information. Even without a mandate, what levels of community cross-communication are needed for the DoD to achieve its goals? What are the incentive mechanisms that can move interests? Addressing these questions will lead to more effective policy and its maturation.

The group also felt that we, as a community, are too focused on the "now," and that decision makers have impoverished ideas of what can be done with computers. The participants said that complexity frequently prompts decision makers to turn to a simple means instead and renders them skeptical to new approaches, perhaps due to the strongly risk-adverse culture prevalent across the DoD.

The group additionally identified multiple hinderances to ontological development and effective implementation: encouraging data silos, encouraging one-size-fits-all, not appropriately supporting policies with resources (including staff, working groups, program funding, training, etc.), and over-specifying policy when it is still an open question.

Group 2

This working group focused on enterprise level scaling and adoption and was guided by the following questions:

- **What working model and policies will equitably support applied computation ontology development and evolution?**

- **How do we approach policies for governance – what is needed, how to develop, where do they live?**
- **What workforce development needs are seen and what development paths are needed or exist?**
- **What are the critical dimensions for transition?**

In discussing policies, the group discussed when in the evolution of ontologies for Digital Engineering policies are appropriate. In general, the conclusion was that a larger community of practice is needed to create demand for policy. Policy is intended to change behavior, and more behavior is needed before the policy solution is appropriate.

A suggestion for generating this community of practice is by putting ontologies and their use in Digital Engineering into a charter of a working group to be explored. This could help build a community of practice, or "cast of the willing," to create a demand signal that eventually could result in official policy changes. This community of practice needs to be properly resourced, and the example of the JFAC from the Assurance community was given as a potential model for providing resources and a helpline for those needing support. Further investigation into how to standup a JFAC type group for ontologies is needed. Additional resources could include an ontology handbook, repository of POCs within and separate from the organization, comparisons of top-level ontologies and guidance for how to pick one, and repository of available ontologies.

Additional notes on moving from practice to policy include:

- Incubation. Create a guide from lessons learned in the incubation period. Form a community from those that contribute to and use the guide. Form a policy when the community is large enough (and best practices are firm enough) to warrant it.
- Formalize the concept and the language in guidance. Get the suggestion of ontologies into a general document (e.g., a Defense Acquisition Guidebook (DAG)). Having the language be a part of an official document will help create openness to using ontologies.
- Include the concept in roles and responsibilities. Get "ontology" into someone's charter or job description. Eventually, this needs to be somebody's responsibility.

It was noted that the organizations we are interested in influencing are not ontologically minded. We must work to persuade them of why ontologies are beneficial to their missions.

Data interoperability is the felt need. Discussion of ontologies needs to map their use to data interoperability; we must convince people that data interoperability is achieved by development and use of ontologies. This connects the use of ontologies to fulfilling existing strategy. An additional selling point is the enabling of richer automation.

What level of organizations ought to be targeted with this message was discussed. The product line level (e.g., PEO, etc.) was offered as a possible target audience, but it was also noted the Chief Information Officer (CIO) and Chief Data Officer (CDO) will be critical players eventually.

There seems to be an acknowledged need for defining what good design looks like, often using reference architectures. While it was pointed out that "reference architecture" may refer to many different things depending on the organization using the term, the acknowledged need was discussed as a starting point for discussing ontologies. This discussion led to the question of whether people are doing work in ontologies and calling it something different (e.g., reference architectures), which was met with some pushback that some people call what they do ontology while producing something incompatible with our conception and use of ontologies.

Governance beyond the policy discussion above was discussed briefly. It was noted that ontologies can be local to a team or project, with the idea that official governance may not have a place in demanding how they are used in local settings. However, governance can be used as a guide to make ontologies developed for local use interoperable with other ontologies. It was also noted that governance of a taxonomy of terms is a simpler problem than governance related to axioms establishing relationships between things in ontologies.

While there was general agreement that building a community was necessary to affect real change (i.e., a bottom-up perspective), use of top-down requirements was noted as being potentially effective in certain instances. For example, inclusion of language like "Your reference architecture shall be tied to…" may connect efforts people are already doing to the use of ontologies that could enable more interoperability in the future. Another example is including the use or development of ontologies into a PM's evaluation formula. An example was given of how Digital Engineering was now included in some evaluation criteria, and it was noted that this inclusion had changed behavior of the PMs.

The U.S. Customs and Border Patrol (CBP) had an interesting working model for transitioning new ideas into practice that encourages PMs to adopt new strategies without being strapped with the entire cost/risk of adoption. This model could be further reviewed and may provide guidance for how we could introduce ontologies in project settings.

Group 3

To scale Digital Engineering ontologies to the enterprise-level, the group considered the critical technical and cultural dimensions, including but not limited to workforce development. Incentives are important to support applied computational ontology development and evolution. Some of the incentives are shared cost, complexity reduction, improved communication, and identifying existing project to try on (versus the area of Defense as a whole). The group discussed value of ontologies and developing a compelling business case for the need for ontologies. The amount of data available is unfathomable by human brains, and ontologies can help humans understand vast data analyzed via digital means to identify what provides value.

Ontologies are a way of implementing MBSE and have subject matter experts working on them across the U.S Intelligence Communities. A suggestion was proposed to develop a guidebook or engineering policy. There are policies that will be coming out soon that will require the use of ontologies to implement the DoD Data Strategy. From a tactical point of view, it is vital to have early wins and prove a one-time solution, e.g., conducting a pilot from end-to-end. Creating more case study examples and use cases can help the community to understand the operationalization of the system. The roles of vendors were discussed including getting them to agree to a comprehensive exchange, have an open format that is readable to other programs, internal coding that belongs to the government (i.e., not just the vendors), and consequently no vendor lock. There is a need to change vendor practices and incentivize them to work together. Having government agencies and partnerships with prime contractors together could get these entities to agree on the needs.

In terms of the policies for governance, there is a need for a governance structure for ontologies in the DoD. Working groups exists in this area and one of the key properties is interoperability. In workforce development, there is a need for training as there are no comprehensive guidebooks for ontologies. Looking at the larger Digital Engineering or MBSE community, there is a cultural shift. To use ontologies, the culture needs to change so everyone looks at ontologies the same way, developing a common literacy. To get industry to understand, there is a need to learn how to describe this in a simpler manner and make it applicable to real world. For workforce development, there are competencies at different levels for ontology-based training. A suggestion was made to tie this to strategic plans for workforce development to include Digital Engineering ontology. NIST has a framework on cyber security roles and skills associated with the roles which may have some utility here. There is a joint INCOSE, ISO, and NDIA working group that is developing a standard for Digital Engineering information exchange to be presented to ISO by the end of the year.

SERC is working with the DAU on a research task (WRT-1043: DAU Digital Engineering Simulation) to develop a Digital Engineering credential where students are expected to have some basic digital literacy knowledge and understanding of Digital Engineering and basic understanding of DoD acquisition processes and policies. The Digital Engineering credentialling will allow technical or engineering acquisition professionals to transition from legacy systems engineering and development methods into a model-based environment. The courses in the DAU Digital Engineering credential include Digital Engineering using SysML, Model-Based Systems Engineering, Digital Enterprise Environment, Digital Engineering Acquisition Management Processes, Digital Engineering Technical Processes, and a Digital Engineering Intermediate Credential Capstone.

CONCLUSION

Overall, the workshop exposed that we, as a systems engineering community engaged in supporting digital transformation across the DoD, need but do not yet have a broad understanding of applied computational ontology development and use that will ensure critical aspects of digitally-based decision making:

- Seamless integration of data analytics,
- Harmonized syntactic and semantic information sharing across systems, and
- Knowledge-based decision making relevant to the convergence of Digital Engineering and system engineering practice.

In systems engineering, knowledge derives from the outcome of processes that can be designed, assessed, and improved. As an underlying data model, an applied computational ontology is not simply a reflection, but an active part of shaping data and the conclusions drawn from that data – or, more specifically, how that data is represented and related. Applied computational ontologies will shape semantic data practices.

The workshop revealed gaps in our understanding and practice across three major areas:

1. Foundations for applied computational ontology development, evolution, and use across real DoD programs,
2. Defining and describing the value of applied computational ontologies to Digital Engineering practice, and
3. Critical dimensions for operationalizing applied computational ontologies and transitioning them to scale.

Foundations for applied computational ontology development, evolution, and use across real DoD programs

Different ontological development choices define how entities are separate from or relate to one another and how these relationships are represented. This results in different ontological structures with variation across them in terms of how the world is mapped and hence interpreted. Consequently, this impacts how one can compute on data represented and organized within an ontology as well as how readily different ontologies can be merged or not. Relatedly, since the ontology design choices – the commitment decisions made – define how an ontology views and represents the world, there is often some type of inherent bias. Bias in an applied computational ontology can lead to omission, incorrect attribution, and poor deduction. We need to better understand how even the unique language and concepts prevalent across the DoD will, as with all language, be malleable.

- **How does malleability of language and the transient nature of the technology and business (i.e., acquisitions) worlds we are striving to describe impact the design, evolution, use, and governance of applied computational ontologies?**

As a community, we do not yet have a well-established foundation for understanding choices and use across different ontology models. For example, we do not know how to consistently and efficiently determine (i) which ontology model types are best suited for which purposes or domains, and (ii) which ontology model types are combinable for computational use (i.e., to support Digital Engineering activities) with other ontological model types for further interoperability. (This latter concern is related to but not limited to unified versus stratified ontology design choices.) Further, we do not yet have the understanding and methods necessary to evaluate applied computational ontologies with respect to how “good” they are, i.e., their ability to represent the domain(s) in question well and support the types of data and information synthesis, analysis, and query necessary in the practice of Digital Engineering. This includes how to evaluate them and the metrics to do so, and how to stress test them, which will require not just use cases but defined classes of use cases. It will also include better understanding of best use of the different languages used to represent and query ontologies given they can (a) constrain which concepts in the ontology itself are readily expressed, and (b) these languages have different levels of compatibility or lack thereof with UML-based approaches such as those using UAF or SysML (e.g., OWL compared to SHACL, which is why many users end up employing both in the same solution space).

- **What are the approaches and methods necessary to evaluate and test applied computational ontologies with respect to how "good" they are, i.e., their ability to represent the domain(s) in question well and support the types of data and information synthesis, analysis, and query necessary in the practice of Digital Engineering?**

Defining and describing the value of applied computational ontologies to Digital Engineering practice

It is well-understood within the systems engineering community that advances in technology and our abilities to digitally manifest artifacts do not automatically bring interoperability. We still need a common vocabulary for representing and organizing data and information to define the meaning of concepts and relationships, especially as unique to a given problem area or domain. Applied computational ontologies can enable different systems and applications to share and understand data consistently and unambiguously. Even so, it seems that ontology discussion for these purposes has to date been more point-solution oriented, significantly limited in application and scope. We are still lacking not just consensus in what ontologies are and how we need to apply them but also consistency with respect to how we talk about ontologies and use them. Do we even have the tools to build these consistencies at present?

Pointedly, we talk a lot about applied computational ontologies and how they enable reuse of domain knowledge, make domain assumptions explicit, can support analysis via bringing a common basis and synergy between syntactic and semantic interoperability, etc. In general, as the practice of systems engineering becomes increasingly digital, we continue to frequently conflate system modeling and its orientation toward integration with knowledge management and its emphasis on reasoning. We are not concrete with respect to what specific problems we need to solve that we cannot solve without using an applied computational ontology. We need to better identify these problem spaces in order to best evaluate and communicate the value these approaches will bring, especially in comparison to advances in expansive, flat, graphical data models also becoming more prevalent.

- **What are the problem spaces that must use an applied computational ontology to move forward and achieve the desired outcomes?**

- **Which digital threads are not achievable via syntactic interoperability alone but will also require semantic harmonization or normalization?**

Critical dimensions for operationalizing applied computational ontologies and transitioning them to scale

One of the greatest challenges the workshop participants identified comes at the intersection of policy, governance, and culture. Many leaders across the DoD are time-limited in their positions, under great pressure to be risk adverse and achieve demonstratable results in a comparatively short time frame on budget. They are not evaluated for their ability to set up what for now seems to be a vaguely defined return on investment with respect to a program's Digital Engineering capabilities five years out. At our present level of understanding and maturing, selection, development, and implementation of applied computational ontologies takes time and funding that program level leaders often do not have.

We need models and policies to support DoD-specific development and evolution. Namely, we need to find ways to equitably incentivize development and use in critical problem areas that need ontologies. We do not want to passively create an environment where first movers do the most work and bear the cost while other organizations sit back and benefit from these advances later with minimal time or cost investment of their own. Similarly, we do not want use stifled as organizations put in great investment and then do not share – as there are no mechanisms or encouragement to do so – both the advances and the lessons learned.

- **What governance models and policies will promote and incentivize applied computational ontology development, evolution, use, selection and sharing of use cases, test queries, trade results, and best practices for implementation across the DoD?**

Moreover, the practice of systems engineering is increasingly data driven and moving to increasingly digital artifacts and processes. If we can identify and begin to pilot the problem spaces for which applied computational ontology foundations are critical to achieve the desired capabilities and outcomes, we can begin to gain the insight to define associated workforce development needs. The participants were in agreement that we will need to expand our education and training to include those foundations of data architecting, data modeling, and data science specifically relevant for tomorrow's systems engineers. We do not yet understand, however, how front-line engineers will interact with ontologies and consequently what other skills may be required. Returning to the notion that the value of ontologies lies in what they allow us to communicate, participants recognized that systems engineering, while increasingly digital in nature, is also increasingly socio-technical. It is difficult to achieve consensus in any area with complexity, and perhaps ontologies can help us achieve improved socio-technical practice.

- **What can we learn through targeted pilot problems to identify the workforce development needs from the line engineering to the decision makers for effective and successful use of applied computational ontologies across digital and systems engineering practice?**

Summary

In summary, operationalizing and scaling development, evolution, and use of applied computational ontologies in Digital Engineering across the DoD will require (i) advances in our fundamental understanding of these dimensions and how to assess them, (ii) identifying and piloting their application in problem spaces where we cannot achieve the desired outcomes without them, (iii) creation of a policy and governance structure to promote and incentivize sharing of development and lessons learned, (iv) and, likely through these pilots, improved understanding of the workforce development needs to support effective and efficient use of applied computational ontologies in Digital Engineering.

ACKNOWLEDGEMENTS

The organizers would like to offer special thanks to the presenters in this workshop who shared their knowledge, expertise, and experience. Thank you to MITRE and SERC for planning and facilitating, and to all the participants for the engaging discussion, innovative ideas, and information exchange. It was an enriching gathering and opportunity to bring the community of interest to learn from one another on this topic.

APPENDIX A: WORKSHOP ORGANIZERS

Executive Hosts:

- **Dr. Dinesh Verma**, *SERC Executive Director, Stevens Institute of Technology*
- **Dr. Judith Dahmann,** *Technical Fellow, MITRE*

Workshop Technical Leads:

- **Dr. Mark Blackburn,** *SERC Senior Research Scientist, Stevens Institute of Technology*
- **Dr. Val Sitterle**, *Principal Research Engineer and Chief Scientist, Systems Engineering Research Division, Georgia Tech Research Institute*

Moderators:

- **Dr. Alejandro Salado,** *Associate Professor of Systems Engineering, University of Arizona*
- **Dr. Phil Anton,** *Acquisition Innovation Research Center (AIRC) Chief Scientist, Stevens Institute of Technology*
- **Mr. Tom McDermott,** *SERC Chief Technology Officer, Stevens Institute of Technology*

Assistant Moderators:

- **Mr. Daniel Dunbar,** *Ph.D. Candidate and Graduate Research Assistant, Stevens Institute of Technology*
- **Ms. Molly Nadolski,** *SERC Research Scientist, Stevens Institute of Technology*
- **Dr. Hoong Yan See Tao,** *SERC Research Project Manager, Stevens Institute of Technology*

APPENDIX B: ACRONYM LIST

AoA – Analysis of Alternatives
AAF – Adaptive Acquisition Framework
AFC – Army Futures Command
AFD – Assessment Flow Diagram
AIRC – Acquisition Innovation Research Center
AI/ML – Artificial Intelligence/Machine Learning
AMC – Army Materiel Command (AMC)
ASA-ALT – Assistant Secretary of the Army for Acquisition Logistics and Technology
CBP – Customs and Border Patrol
CIO – Chief Information Officer
CDO – Chief Data Officer
DoD – Department of Defense
DAG – Defense Acquisition Guidebook
DAU – Defense Acquisition University
DE – Digital Engineering
DEE – Digital Engineering Ecosystem
DEMM – Digital Engineering Maturity Model
DEVCOM AC – Combat Capabilities Development Command Armaments Center
EW – Electronic Warfare
FFRDC – Federally Funded Research Development Center
JPEO A&A – Joint Program Executive Office for Armaments and Ammunition
JPL – NASA Jet Propulsion Lab
IAPR – Integration Acquisition Portfolio Reviews
IoIF – Interoperability and Integration Framework
MBSE – Model-Based Systems Engineering
NLP – Natural Language Processing
OSD – Office of the Secretary of Defense
PEO – Program Executive Offices
PM – Program Management
SE – Systems Engineering
SERC – Systems Engineering Research Center
SWT – Semantic Web Technology
SysML – Systems Modeling Language
TAMMS – Tactical Ammunition Management Micro Services
T&E – Testing and Evaluation
TTP – Tactics, Techniques and Procedures
UARC – University Affiliated Research Center