Final Report (2019-09-01 to 2020-12-31)
Authoritative Source of Truth Study (ASoT Study)
W911W6-17-D-0003/W911W6-19-F-703D

Contractor: Adventium Enterprises, LLC
D.B.A. Adventium Labs
Contract No.: W911W6-17-D-0003/W911W6-19-F-703D
Technical Contact: Mr. Tyler Smith
Email: tyler.smith@adventiumlabs.com
USMail: 111 3rd Ave S
Suite 100
Minneapolis MN 55401
Phone: 612.414.8046

DISCLAIMER:
This material is based upon work supported by the U.S. Army Combat Capabilities Development Command Aviation & Missile Center under contract no. W911W6-17-D-0003/W911W6-19-F-703D. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of U.S. Army Combat Capabilities Development Command Aviation & Missile Center.

Contents

1 Executive Summary

This is the final report for the Authoritative Source of Truth (ASoT) study. From September, 2019 to December 2020 Adventium Labs conducted a study to elicit, refine, and exercise requirements for an ASoT.[1] This study was part of the Army-Funded Joint Multi-Role (JMR) Mission Systems Architecture Demonstration (MSAD) Capstone demonstration.

The objective of this study was to define requirements for an ASoT at a sufficient level of detail as to enable acquisition or use of ASoT capabilities in support future Army efforts. The next objective was to provide proof-of-concept demonstrations that the identified requirements can be satisfied. During the first six months of the effort we elicited requirements by conducting interviews with Army stakeholders and surveying the existing body of ASoT research. In the following nine months we assembled and presented three demonstrations, each highlighting a different aspect of the ASoT requirements as prioritized by Army stakeholders (in order: requirements management, analysis and collaboration, traceability and digital thread).

The concept of an ASoT differs significantly from its realization. Conceptually, an ASoT is a single repository that contains all of the digital artifacts, analysis evidence, managed relationships, and artifact dependencies, along with links to the applied development tools, analysis tools, and compliance tools, that are necessary for continuous building of the operationally approved system. In practice, an ASoT is a program-specific collection of component-specific repositories under the control of multiple stakeholders that use a mix of standardized, custom, and manual interfaces along with stakeholder-specific knowledge and processes, interoperating under manual control and oversight, to manage time-dependent builds of the system. Conceptually, an ASoT contains a set of digital artifacts that derive from discrete, standalone build processes and that each artifact includes by default all details necessary for the Government to recompete that artifact. In reality, an ASoT contains digital artifacts that derive from time-sensitive slices of larger, stakeholder-proprietary product lines, and the Government must specify in advance the details that it will require to recompete any particular digital artifact, and those details will vary from artifact to artifact.

Each vendor's interaction with the Government-owned ASoT will vary with that vendor's internal development processes. For example, a vendor following an agile software development process will deliver code artifacts frequently. The ASoT must enable the Government to accept, store, review, and use these code releases, as well as track the tools required for code compilation and analysis. A vendor developing FACE conformant software will deliver code, models, and conformance testing results. The ASoT must enable the Government to accept, store, review, and use these assets while observing software rights and licenses for primary and third-party developers. A vendor following a model-based development process will deliver models, model analysis results, and model-generated code. The ASoT must enable the government to accept, store, review, and use these assets in various modeling languages, and maintain the required model translators and model analysis tools. Our recommendations address these and related use cases.

[1] This study was originally called Single Source of Truth (SSoT), but has been renamed to ASoT for consistency with the Department of Defense (DoD) Digital Engineering Strategy.

An Authoritative Source of Truth necessarily touches on the concerns of many stakeholders, and the breadth and depth of prior work in this domain is large. To mitigate the risk of scope growth on this limited study, the team structured the program schedule with an initial period of broad conceptual exploration via collection of user stories, followed by a midterm review with our Army customer, during which we finalized the scope of the remainder of the study using an Army-driven prioritization of the user stories.

In concert with the user story collection task we created an ontology for an ASoT. An ontology is a collection of the concepts and their relationships. We constructed this ontology to facilitate *internal* consistency within our work products related to ASoT. To validate the ontology we created a database that captured the structure of the ontology and allowed us to create and track examples that followed the lexicon and cardinality rules established by the ontology. The ontology is provided in Appendix B.

From the user stories we derived use cases, using the ASoT ontology to provide consistent terminology. From those use cases we further derived individual ASoT requirements. The user stories, use cases, and requirements are all provided in Appendix A. We recommend transitioning these requirements to an Army-owned Dynamic Object-Oriented Requirements System (DOORS) system for ongoing maintenance and to streamline use on future Army programs (we can provide them in Comma Separated Value (CSV) format for streamlined importing).

We performed a survey of available Commercial Off The Shelf (COTS) and open source tools that provide functions called out in our user stories. We used this survey as a starting point to select tools to incorporate into our demonstrations.

We conducted three demonstrations, which are described in detail in Section 3. In demonstration one we used International Business Machines (IBM) DOORS Next Generation (NG), Intercax Syndeia, and NoMagic MagicDraw to demonstrate capabilities for requirements propagation and synchronization between different tools in an ASoT. In demonstration two we used analysis and automation tools from the Curated Access to Model-based Engineering Tools (CAMET) library to demonstrate propagation of requirements changes into design changes and analysis results. We showed an approach for digital artifact labeling to support user-specific dashboards of design information. In demonstration three we used Neo4J and Open Services for Lifecycle Collaboration (OSLC) to capture, explore, and present traceability metadata across multiple data repositories. We used the Aras Innovator Product Lifecycle Management (PLM) tool to track a digital twin and trace an error report from a deployed asset all the way to requirements, analysis, and design space exploration. The results of each demonstration were fed back into the requirements provided in Appendix A.

Conclusions. The requirements of an ASoT meeting Government objectives are technically feasible. However, some of the enabling technologies are immature or insufficiently adopted, and others are well supported by existing commercial tools, but that the support is either locked to a single vendor or requires a high degree of customization, which may lead to higher cost and fragility over time. Several challenges that make an ASoT particularly difficult in a Government procurement setting (as opposed to in industry in a commercial or industrial setting). These challenges, such as the need to manage artifacts with varying data

rights and the need to maintain a credible "threat of recompete" should be explicitly addressed for any Government agency pursuing creation or use of an ASoT (these are described in detail in Section 5).

Organization of this Report This report first describes our technical approach (Section 2) and results (Section 3), approximately in the order in which we completed tasks. Next, this report provides deep-dives into specific topics relevant to procuring and establishing an ASoT for Army programs. They are the following: relevant to procuring and establishing an ASoT for Army programs. They are the following:

- Object identity and equivalence: Section 4
- ASoT procurement guidance: Section 5
- OSLC and its potential role supporting an ASoT: Section 6
- Needs and challenges associated with provenance in an ASoT: Section 7
- Summary of ASoT related issues and observations from the overall JMR MSAD Cap-stone exercise: Section 8

2 Technical Approach

We structured the technical approach for progressive refinement as the study progressed, as shown in Figure 1. Throughout the study we engaged with Army stakeholders so our technical focus remained inline with Army needs.

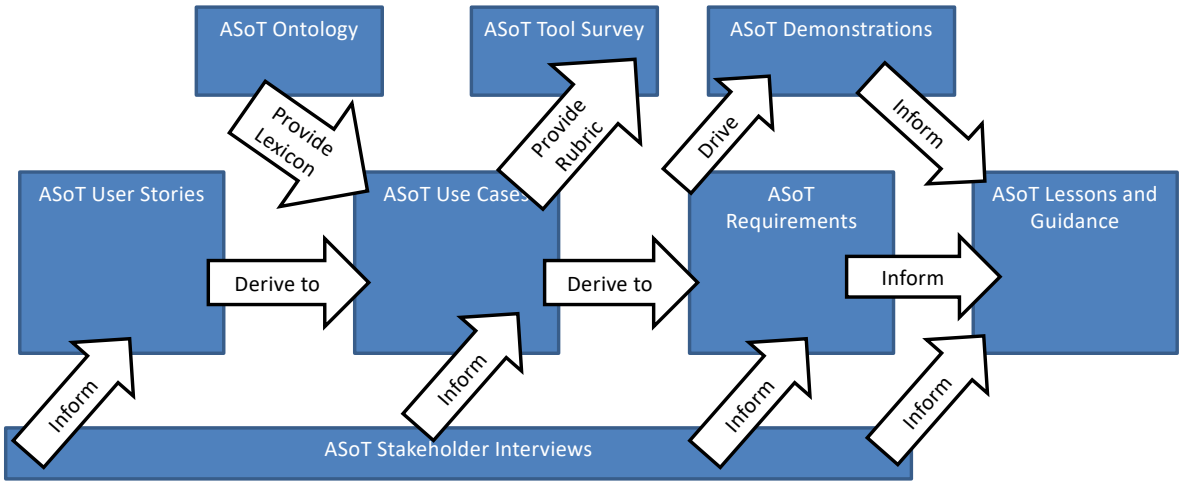

Figure 1: *Authoritative Source of Truth Activity Flow*

This section follows the organization of Figure 1 from left to right.

2.1 Ontology

Early in the project and based on unclear terms and concepts used in industry, we determined that a formalized ASoT terminology would help clarify intent and communicate ASoT results. So we created an ontology to track our selected definitions and support internal consistency. Existing definitions address some aspects of ASoT (prominent offerings include those of International Council on Systems Engineering (INCOSE) and Object Management Group (OMG)), but we determined that it was necessary to develop an ontology specific to this study so that we could refine and add terminology as needed [14] [27]. We decided to create an ontology rather than a glossary because an ontology formalizes relationship between terms. The full ontology is provided in Appendix B.

2.1.1 Key Terms

This section of the report highlights key definitions and summarizes referenced prior research.

Authoritative Source of Truth Definition Authoritative Source of Truth: Current use of this term includes a widely varying notion of scope, and some usages imply particular implementations. To reduce ambiguity and avoid bias toward any particular implementation, we opted for a functional definition:

An Authoritative Source of Truth is a capability that gives definitive answers to queries about a collection of systems.

Where:

- **Queries** are questions about the past, present, or future state of the design or implementation of a collection of systems.
- **Definitive answers** are answers governed by organizational processes.
- **Categories** are functional groupings of queries based on typical tool and process organization.

This definition leverages and refines the DoD definitions of "authoritative data" and "authoritative data source" [19] [22].

Artifacts Central to the data-management responsibilities of an ASoT is the need to manage so-called *artifacts*. To avoid ambiguity, we opted to use the term "digital artifact" as a more precise term than "file," "model," "artifact," etc. We used a definition of *digital artifact* very similar to that used by the OMG but with additional clarifications to enable unambiguous description of version management and traceability capabilities [28]. A "digital artifact" is analogous to an "object" as defined by the National Information Assurance Glossary, but with the added constraint of immutability [29].

A digital artifact is a specific, unique, and immutable piece of information. A digital artifact has a fixed length and fixed internal structure. In practice a digital artifact is analogous to a file tracked by a version control system, except that each version of the file is a distinct digital artifact. A digital artifact can be raw data, or it can be a collection of references to other digital artifacts.

Models We distinguish between the logical concept of a model and the instantiation of that model as one or more digital artifacts, using the term "model" exclusively for the former. This distinction is important because the means of storing, transmitting, or displaying a model may not have matching cardinality - for example, an Architecture Analysis and Design Language (AADL) model may be stored as multiple files or as a single file. DoDI 5000.61 provides an appropriate definition, which we adopted into the ASoT ontology:

A physical, mathematical, or otherwise logical representation of a system, entity, phenomenon, or process [21].

2.1.2 Related Standards

- Modeling and Simulation Coordination Office Glossary [19].
- National Information Assurance (IA) Glossary [30].
- Sharing Data, Information, and Information Technology (IT) Services in the Department of Defense 8320.02 [23].

2.2 Requirements Elicitation

2.2.1 User Stories

To define the scope of the ASoT study we collected user stories describing potential uses of a notional ASoT. Each user story captured one or more stakeholders, stakeholder concerns, ASoT capabilities, and expected outcomes.

An example of one such user story is included below:

2404: The Government will provide software as GFI from a previous contract to a new contract with a different performer. The awarded contractor of the new program wants to know the licensing of the the GFI and also wants to understand any associated open source licensing rights in open source components of the software. The contractor is tasked with adapting the software to a new platform. Why: The contractor wants to know how to mark the newly developed software and what, if any, licensing wording or issues may come up with open source components. How: ASoT will contain previous marking information and licensing information on the GFI.

We used these user stories to feed the requirements definition process, as shown in Figure 2.

2.2.2 Stakeholder Interviews

We engaged with a variety of stakeholders in an effort to understand both Government and industry perspectives and to reduce risk that our final requirements set would not include high priority capabilities or would include requirements that are "deal breakers" for one stakeholder or another. We put special emphasis on engagement with Army stakeholders because the primary goal of this study was to create requirements of use to the Army. The stakeholders we met included:

- **U.S. Army Combat Capabilities Development Command (DEVCOM) Aviation**

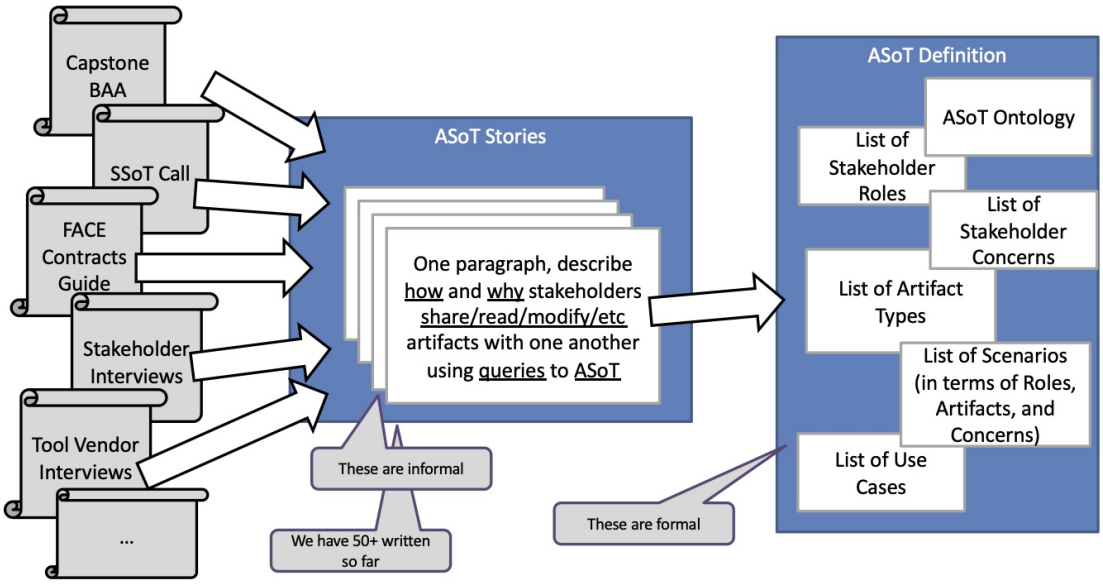

Figure 2: Flow of information from informal user stories to use cases.

& **Missile Center (AvMC)**: As the funding office, DEVCOM AvMC was our most significant stakeholder and held final authority in defining the scope of the study.

- **Program Executive Office (PEO) Aviation**: PEO Aviation is a potential end user of an ASoT solution.
- **JMR MSAD Capstone Performers**: We invited each of the performers to meet with us to share their concerns and desires for an ASoT.
- **Tool Vendors**: We met with a variety of tool performers to gather data for our tool survey and to understand user stories supported by their tools.

2.2.3 Additional User Story Sources

In addition to interviewing stakeholders, we surveyed JMR MSAD Capstone Government Furnished Information (GFI), prior research on ASoT and related topics, including prior research by Adventium Labs. As we solicited user stories and developed our ontology, we defered as much as possible to prior research, placing highest priority onDoD funded research and standards.

- **Capstone Broad Area Announcement (BAA)**: The JMR MSAD Captone BAA provided the general scenario for the Capstone demonstration overall. The Capstone BAA provided an overview of the types of stakeholder interactions an ASoT needed to support.
- **Capstone SSoT Call**: The Capstone SSoT call was the inpetus for this study. The SSoT call includes a variety of scenarios, all of which we incorporated into our user stories.
- **Future Airborne Capability Environment (FACE) Contract Guide**: The

FACE Consortium published a guide for writing contracts for acquisitions involving FACE conformant and aligned components. This guide provided example contract language and guidance that informed our user stories and our ASoT procurement guidance (see Section 5) [10].

- **Comprehensive Architecture Strategy (CAS) Paper**: The CAS paper describes an approach for managing business drivers and functional requirements [8].
- **General Electric (GE) SSoT Paper**: During the JMR MSAD Architecture Implementation Process Demonstrations (AIPD) exercise, GE delivered a paper on the SSoT subject that provided an industry perspective on information management and tool usage.
- **DoD Digital Engineering Strategy**: The DoD Digital Engineering Strategy explicitly calls for use of an ASoT. This document is what prompted us to change the name of this project from SSoT study to ASoT study [25].[2]
- **Capstone Integration Pilot**: In 2018 Adventium conducted a pilot study of ASoT concerns described in the paper *Lessons Learned in Inter-Organization Virtual Integration* [35]. In the pilot study we evaluate the application of software engineering practices (e.g., continuous integration) to model-based systems engineering. We determined that practices adopted from software engineering provided value to model-based engineering efforts.
- **Contracting Considerations for Agile Solutions**: This study provides a summary of *agile* project management concepts and recommendations for creating contracts for agile projects [31].

2.2.4 User Story Prioritization

We worked with our Army customers to prioritize the user stories we collected. The prioritization shown in Figure 3 shows the *research* priority of each category of user story. For example, **infrastructure** received a medium research priority because the systems to support infrastructure user stories (such as user story 2599, which describes how an ASoT should handle a power outage, described in Appendix A) are well understood and supported by existing capabilities.

2.2.5 Use Cases

After collecting user stories and defining the initial ASoT ontology, we developed *Use Cases*. The flow from user stories to use cases is shown in Figure 1. We used *links* in DOORS NG to track the derivation relationships between user stories and use cases (a single user story can derive to multiple use cases, and vice versa). In our approach, a use case differs from a user story in several ways:

- A use case describes a single user, action, and objective, whereas a user story can describe multiple users, actions, or objectives.

[2] Contractually the name of the program is unchanged, but for reports and documents we use ASoT.

Irrelevant	Low Priority	Medium Priority	High Priority	Critical Priority
	• Process Management • Data and Artifacts	• Infrastructure • Safety • Collaboration • Views and User Experience	• Legal and Contracts • Tradeoff Analysis • Metrics • Analysis and Modeling • Airworthiness	• Security • Traceability

Figure 3: *Authoritative Source of Truth User Story Research Prioritization based on Stakeholder Feedback*

- A use case uses terminology from the ASoT ontology, whereas a user story can be informal.

An example use case derived from the user story presented in Section 2.2.1 is:

3401: The ASoT supports the program manager by providing contract based guidance

2.2.6 Requirements

After deriving use cases from the user stories, we repeated the process to define *requirements*, as shown in Figure 1. We again used DOORS NG links to trace the relationship between use cases and requirements. Requirements differ from use cases in several ways:

- Requirements are phrased as *shall statements* that describe something the ASoT implementation must do.
- Requirements do not include rationale (instead they refer back to the motivating use cases).

An example requirement derived from the use presented in Section 2.2.5 is:

3549: The ASoT shall associate all marking and licensing information with an artifact, even if derived from prior contracts.

The requirements for a given ASoT depend on the categories of queries for which it is required to give definitive answers (for example, a program office may decide that its ASoT does not need to address queries about financial information). We based our ASoT requirements on categories of queries that we derived from our use cases. At the direction of our Technical Point of Contact (TPOC), we left out lower priority topic areas such as workflow management

2.2.7 Validation

In concert with the development of the ongology we created and evolved an example database to exercise and validate the terminology. We created database tables for terms in the ongology and populated the database with example data, using the JMR MSAD Capstone projects as an example scenario.

2.3 Tool Survey

In preparation for the ASoT demonstrations, we conducted a survey of commercial and open source tools supporting ASoT capabilities. The intent of the survey was to evaluate the capabilities of each tool, not to evaluate the tools reliability, value, etc. We purchased and tested *some* of the tools in the survey and updated the survey based on the results of our testing.

2.4 Demonstrations

Demonstration Approach We progressively defined each demonstration as the study evolved, generally determining a specific set of tools and objectives for each demonsrtration approximately four months prior to the demonstration date. Our requirements elicitation efforts continually refined our understanding of the Army's needs and priorities, so we tailored each demonstration to address stakeholder concerns.

Our overall demonstration methodology was to start with an example scenario (the FireSat) and expand on that scenario throughout the three demonstrations. This approach allowed us to build each demonstration on the previous demonstration, avoiding duplicative work and letting us focus our attention on new capabilities.

A high priority objective of our demonstrations was to evaluate tools at the bleeding edge of ASoT capabilities. Some of the tools we evaluated were not as robust as others, and we occasionally had to work around technical limitations. The goal of the demonstrations was to *exercise* the tools, not to achieve any specific technical objective (we were not actually building a FireSat satellite). With this in mind, even tool "failures" were valuable lessons and outputs of this study. Demonstration-specific technical problem areas are described in Section 3.

Alignment with NASA CAFFMAD The ASoT study aligned closely with the National Aeronautics and Space Administration (NASA)-funded CAFFMAD project, a Civilian Commercialization Readiness Pilot Program (CCRPP) project also conducted by Adventium. CAFFMAD provides a web-based interface for specifying design subspaces, tooling for analyzing multiple design points in a given design subspace with multiple tools, and a web interface for viewing analysis results. CAFFMAD was funded with NASA funds provided as a match to Army investment.

Demonstration Summaries We conducted three demonstrations of increasing complexity, each focused on a different area of technical challenge and interest. Demonstration one

Technology	Purpose	Requirements Elicitation	Demonstration One	Demonstration Two	Demonstration Three
DOORS NG	Requirements Management				
Django	Web Framework				
MagicDraw	SysML Modeling				
GitLab	Version Control				
Teamwork Cloud	SysML Version Control				
Syndeia	Artifact Synchronization				
OSLC	Model Interoperability				
OSATE	AADL Modeling and Analysis				
CAMET	Model Analysis				
Jenkins	Continuous Virtual Integration				
OpenMBEE	Model Management and Reporting				
ARAS	Product Lifecycle Management				
Neo4J	Graph Database				

Figure 4: *Technologies used at various stages of the ASoT Study*

focused on requirements management between tools. Demonstration two focused on analysis and collaboration, and demonstration three focused on traceability. We used a variety of tools in the demonstrations. Figure 4 provides an overview of which tools we used in each phase of this project. Section 3 provides more detail on each demonstration.

3 Results

3.1 ASoT Requirements

We collected 170 user stories across 25 stakeholder domains, including contracting, airworthiness qualification, process management, multi-party product development, and government collaboration and review. We derived 123 common use cases from the user stories. We initially interviewed Raytheon/GE, Boeing, Skayl, Collins, and Honeywell. Raytheon/GE, Boeing, and Honeywell provided additional feedback that we include in the review guide. We identified 107 ASoT requirements from the common use cases. We used DOORS NG to manage traceability from user stories to requirements. All of the requirements are provided in Appendix A.

3.2 ASoT Ontology Lessons Learned

To validate the ASoT Ontology we created a database to store examples of the concepts in the ontology and web server in order to allow us to exercise the terms in the ontology by creating and relating examples. We implemented the validation database using Django, a Python-based web framework. The full ontology is provided in Appendix B.

We successfully created the validation database and used it to refine the ontology. We also used the validation database in parts of our demonstrations (particularly demonstration two). For example, our ontology contains terms and relations describing execution of tests. Figure 5 shows some of the ASoT ontology terms related to testing.

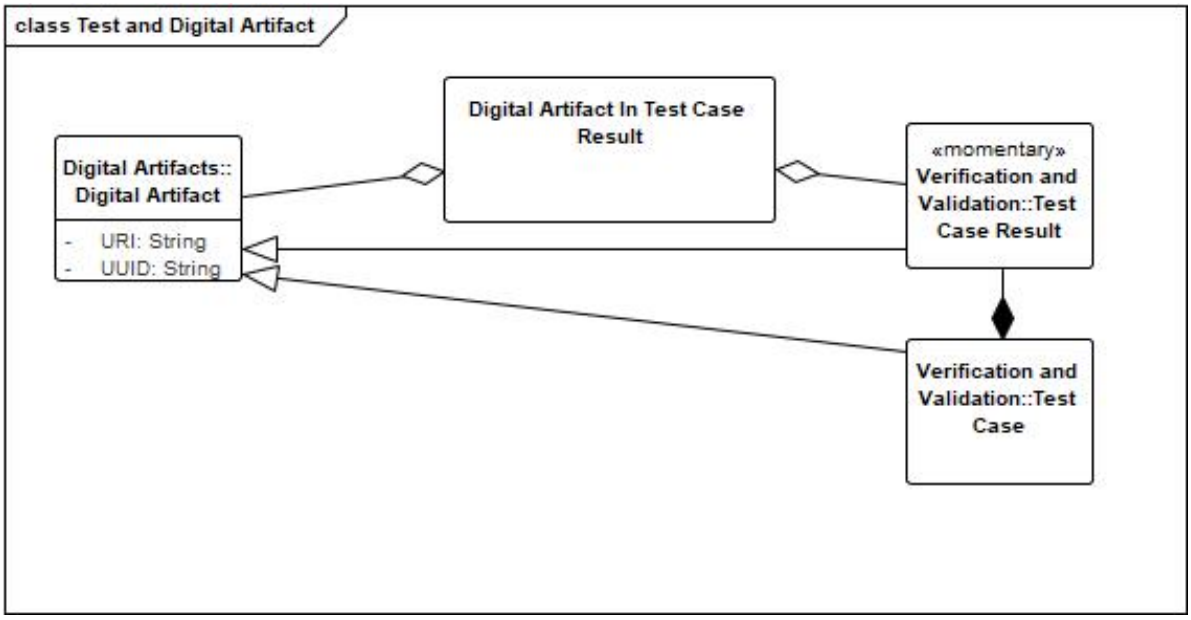

Figure 5: *Digital Artifact and Test Cases in the ASoT Ontology. Relationships between terms are shown using UML notation. For example, Test Case is a* specialization *of Digital Artifact, and Test Case Result has an* aggregation *of Digital Artifact in Test Case Result.*

The validation database contained tables for each of the terms shown in Figure 5. During our demonstrations we populated this database with examples that validated the ontology and supported the demonstration. For example, in demonstration two (see Section 3.5) we ran automated utilization analysis tests on an input model. Using the validation database as part of our ASoT, we could query the database for test results associated with a given model or chunk of a model, as shown in Figure 6. The validation database included links that allowed us to explore the tests and results. For example, to query the database for details on a test result we could click on a test result Universal Unique IDentifer (UUID), presenting the details shown in Figure 7.

Note that we chose to use UUIDs as identifiers for elements of the ASoT validation database. More discussion on the topic of identifiers is provided in Section 4.

In the process of implementing the ASoT validation database, we learned a variety of lessons that will be useful in future ASoT efforts:

Date	Name	UUID
Aug. 24, 2020, 3:04 p.m.	Analyze Utilization firesat.aadl	15e82b15-0082-4485-8cd1-23e48726be6c
Aug. 24, 2020, 3:06 p.m.	Instantiation Construction Project 26	24084121-095f-4d3c-9cae-269181f24ed0
Aug. 24, 2020, 3:07 p.m.	Instantiation Construction Project 26	340f60d4-6647-42f4-9d8d-bdaa331d4fb5
Aug. 24, 2020, 3:13 p.m.	Instantiation Construction Project 26	2821708e-fa0b-40be-be42-09732d7d9a65
Aug. 24, 2020, 3:07 p.m.	Analyze Utilization Construction Project 26	c0d2ea73-7f05-4288-87ad-498b65d26bb7
Aug. 24, 2020, 3:13 p.m.	Analyze Utilization Construction Project 26	f03299d1-191c-4e0f-95c4-2e10d1befa86
Aug. 24, 2020, 4:41 p.m.	Instantiation Construction Project 60	a890cdaa-1c81-4d51-a29f-98a34a3eb4fa

Figure 6: *Example validation database query (the URL) for information associated with SysML model chunk with UUID starting* `7277`. *This screenshot shows the related test results for SysML model chunk with UUID starting* `7277`

Analyze Utilization firesat.aadl

Errors:

1. backup.proc-Resource overloaded, MIPS margin < 1 relative to allowed utilization limit.
2. backup.proc-Resource overloaded, MIPS margin < 1 relative to allowed utilization limit.
3. backup.proc-Resource overloaded, MIPS margin < 1 relative to allowed utilization limit.
4. primary.proc-Resource overloaded, MIPS margin < 1 relative to allowed utilization limit.
5. primary.proc-Resource overloaded, MIPS margin < 1 relative to allowed utilization limit.
6. primary.proc-Resource overloaded, MIPS margin < 1 relative to allowed utilization limit.

Resource: backup.proc

UUID: 7277afbe-0593-4e52-96f9-ebe89aa8c458

Demand	UUID	MIPS Supply	MIPS Demand	MIPS Util
Sum	N/A	250.000	200.000	0.800
backup.virt1	c9000fa6-6f99-4520-abdd-7899682e3c98	150.000	100.000	0.400
backup.virt2	413e9d7b-64bf-4ec6-822d-dc379c4c24bc	150.000	100.000	0.400

Figure 7: *Details from the execution of a single utilization test with UUID starting* `15e8`, *as provided to users of the validation database. This test result shows that the analysis conducted by CVIT, a CAMET Library tool for automated analysis, found multiple processor utilization errors.*

- Multiple levels of inheritance make the ontology difficult to follow - everything ties into Digital Artifact, but it's often not easy to see all of the relevant connections at once. Future work on the ASoT ontology should include consideration of flattening the inheritance structure to make the ontology easier to read.
- The alignment between different viewpoints is not easy to see. For example, the concept of a "shared asset" is not clearly linked to role-based access control, making it difficult to understand requirements for implementation. Future work on the ASoT ontology should include effort to bring concepts from different domains into closer alignment.
- For a more comprehensive and informative ASoT, it may be necessary to expand the ontology for more fine-grained distinctions. An example being `SySMLChunk` which could be split into the various different SysML elements.[3] This would create a richer feed and allow for more insight for the user.
- The ASoT ontology contained primarily *classes* that had textual definitions but few UML attributes. A potential direction for future research is to define different types of attributes and then integrate attribute definitions into the ontology. Also this would pave the way for more work on the operational semantics of the ASoT representational view.

3.3 Demonstration Infrastructure

We used a set of virtual machines to conduct our demonstrations (see Table 1). Many of the machines were used in multiple demonstrations. We need multiple machines because many of the tools we used in the demonstrations had conflicting ports or required different operating systems.

[3]We used the term "chunk" to avoid collisions with existing terms like "part" or "piece" which already have common meanings in modeling and design.

Table 1: Machines used for the ASoT Demonstrations

Machine	OS	Hardware	Software
Virtual Machine 1	Microsoft Windows 10	4 VCPU, 8GB RAM, 40GB HD	Microsoft Office 2016, NoMagic MagicDraw 19, Syndeia plugin to MagicDraw, OpenMBEE Model Development Kit (MDK), Adventium AADL Bridge plugin to MagicDraw
Virtual Machine 2	Cloud Hosted by IBM	Cloud Hosted by IBM	IBM Rational DOORS NG
Virtual Machine 3	CentOS 7;	2 VCPU, 6GB RAM, 40GB HD	Intercax Syndeia Cloud
Virtual Machine 4	Microsoft Windows 10	1 VCPU, 2GB RAM, 20GB HD	Reprise License Manager (RLM), (for Syndeia plugin).
Virtual Machine 5	CentOS 7	1 VCPU, 2GB RAM, 20GB HD	RLM, (for Syndeia plugin).
Virtual Machine 6	Microsoft Windows Server 2016	2 VCPU, 6GB RAM, 40GB HD	Aras Innovator 12 SP9
Virtual Machine 7	Ubuntu 18	2 VCPU, 6GB RAM, 40GB HD	Django
Virtual Machine 8	Ubuntu 18	2 VCPU, 4GB RAM, 40GB HD	Neo4j
Physical Machine 1	CentOS 7	2 x Intel(R) Xeon(R) CPU E5-1650, 100GB RAM, 200GB HD	OpenMBEE Model Management System (MMS) and View Editor (VE)
Physical Machine 2	CentOS 7	4 x Intel(R) Xeon(R) CPU E5-2620, 16GB RAM, 100 GB HD	No Magic Teamwork Cloud

Notes

- **Virtual Machine 1:** Microsoft Office does not need to be installed on the same machine as MagicDraw or the Syndeia Plugin, but having it on the same machine makes it easier to demonstrate the propogation of changes from other tools into Excel.
- **Virtual Machine 1:** The "Connection Viewer" in the Syndeia Dashboard will incorrectly claim it cannot find the connections from model elements to DOORS elements if the user does not first click the DOORS repository in the "Repository Browser" window (which is also in the same Dashboard).
- **Virtual Machine 3:** This is an optional VM used to hold metadata for Syndeia plugin to MagicDraw. Metadata can be held directly in the MagicDraw Model but this would preclude use of the Representational State Transfer (REST) Application Programming Interface (API) offered by Syndeia Cloud.
- **Virtual Machine 4:** The Single seat license for Syndeia requires a server host the license key as it can be checked out by as multiple machines.

- **Virtual Machine 6:** The Aras instance was very slow despite exceeding requirements in the installation guide.
- **Virtual Machine 6:** Aras claims to support Internet Explorer, Firefox (Windows and OSX), and Chrome (Windows and OSX). We found that only Chrome on Windows worked reliably.
- **Virtual Machine 6:** Aras support confirmed upgrade is not possible on the free version of Aras. At one point we needed features in newest version and had to do a full reinstall, losing all previously entered data.
- **Physical Machine 1:** Installation of OpenMBEE was quite difficult. The recommended approach was to use docker containers with all the services pre-configured. We found the containers to be misconfigured in many ways causing them to be unusable. We asked on the mailing list and were told they were unaware anyone was using them and they had not been keeping them in a usable state. We opted for a manual installation which was successful but we hit a handful of small errors in the instructions.
- **Physical Machine 2:** We initially tried to install Teamwork Cloud on **Physical Machine 1** with OpenMBEE MMS and View Editor (VE), but could not resolve port conflicts.

Figure 8 lists all of the tools we used in these demonstrations and shows the traceability and data propagation links between them.

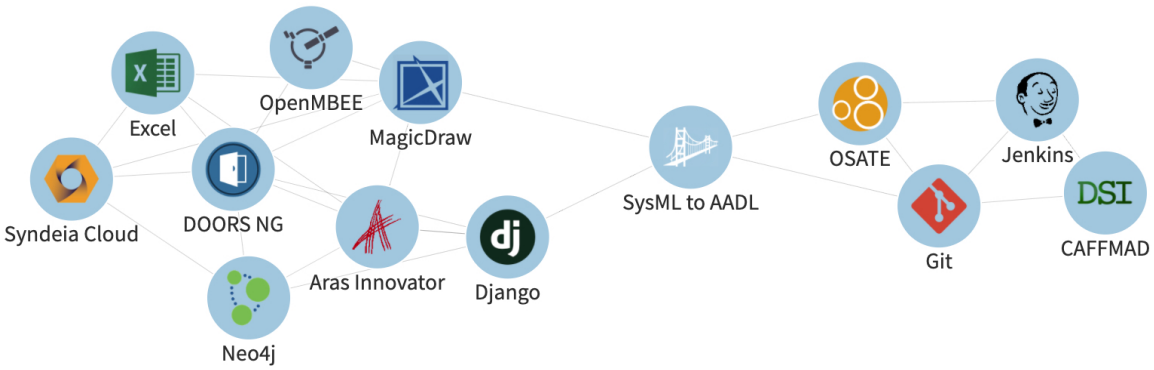

Figure 8: *Tools used in the demonstrations. This image is a screenshot of the ASoT landing page we used in demonstration 3, each icon is a link to a tool.*

3.4 Demonstration One

Demonstration One focused on requirements management and propagation of requirements between multiple tools in an ASoT. The core tools used in demonstration one are shown in Figure 9.

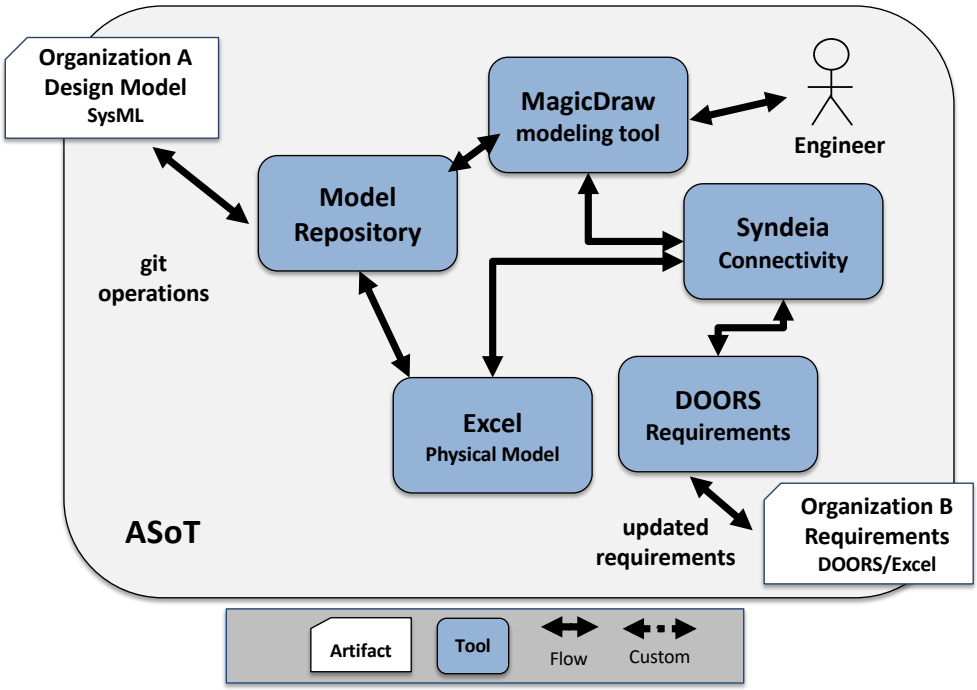

Figure 9: *Demonstration One Tools and Relations*

3.4.1 Demonstration One Overview

Demonstration One Scenario: Organization A is constructing a cluster of satellites for environmental observation. Organization A is using MagicDraw SysML to track requirements for these satellites. Organization A contracts Organization B to construct a satellite for forest fire detection (we are basing our model on the publicly available FireSat project). Organization B uses DOORS to track requirements, SysML to model their satellite design, and a git repository to track resources provided by Organization A. Using Intercax Syndeia, Organization B maintains traceability between the Organization A-provided requirements in SysML to the derived requirements in DOORS. Organization B uses Syndeia to export design information from SysML models for analysis, such as a total mass evaluation using Excel. Partway through the project, Organization A changes launch vehicles, which lowers the total mass requirement for the satellite. Organization A delivers an updated SysML requirements model. Organization B uses Syndeia to propagate the updated requirements into DOORS and into their design, then uses Syndeia to update their mass analysis in Excel based on the new requirements.

Demonstration One Research Objectives: Our top-level objective in Demonstration One was to determine whether the communication illustrated in Figure 9 is feasible. The demonstration provided insight into the "connectivity platform" approach for ASoT tool integration. Connectivity platform is a term we use to describe a tool that defines, stores, and maintains relationship connections (or connection metadata) between artifacts within the ASoT environment (this is described further in Section **??**). The most effective examples of connectivity platforms allow for in-depth connections embedded within larger design assets, such as files, databases, or models. For example, a connectivity platform may provide the means to identify the linkage between a particular requirement in DOORS and a specific

requirements block in a SysML model. Once defined, the collection of artifact relationships maintained in a central repository provides scaleable traceability throughout the ASoT environment that facilitates system analysis and automation. Demonstration One included the connectivity platform tool Syndeia from Intercax.

3.4.2 Demonstration One Results

Limited Automation in MagicDraw. An important initial finding is that the MagicDraw interfaces designed for tool integration and automation do not provide straightforward API-based access to MagicDraw plugins, including the Syndeia plugin for MagicDraw that connects MagicDraw SysML models to the Syndeia database.

Syndeia Works as Desired. The three tool integration connections included in the ASoT Demonstration One, specifically the Syndeia to DOORS NG connection, the Syndeia to MagicDraw connection, and the Syndeia to Excel connection, have worked mostly out-of-the-box, except for minor formatting details within the SysML models and the Excel spreadsheets.

3.4.3 Demonstration One Technical Notes

- Machines used: Virtual 1, 2, 3, 4, 7
- Virtual Machine 1 connects to both DOORS and Microsoft Excel via the Syndeia plugin.
- Virtual Machine 7 (Django) not connected to other machines. All data entered manually.

3.5 Demonstration Two

Demonstration two focused on interactions between stakeholders with difference levels of access to the ASoT and different needs with respect to design and analysis information.

3.5.1 Demonstration Two Overview

Demonstration two explored the relationships between multiple stakeholders in a large-scale procurement effort. As a motivating scenario, we constructed a hypothetical procurement of FireSat satellite manufacturing, maintenance, and operations between several Government and industry organizations. The core tools and data flows in demonstration two are shown in Figure 10. The organizations involved are shown in Figure 11.

NASA provides a GFI SysML model to its developers contracted to design the next generation fire detection satellite. Within the Government ASoT the requirements for the satellite are captured in DOORS and linked to requirements blocks in the GFI SysML model. Contractor A, responsible for the satellite design, uses the GFI SysML model to develop an architecture of the FireSat avionics subsystem. Certain attributes on the avionics subsystem are linked to specific requirements in the GFI SysML, such as maximum utilization thresholds assigned to hardware components. Contractor A delivers a snapshot of the avionics subsystem to NASA in SysML (via .mdzip file). NASA includes the extended model in the ASoT, where

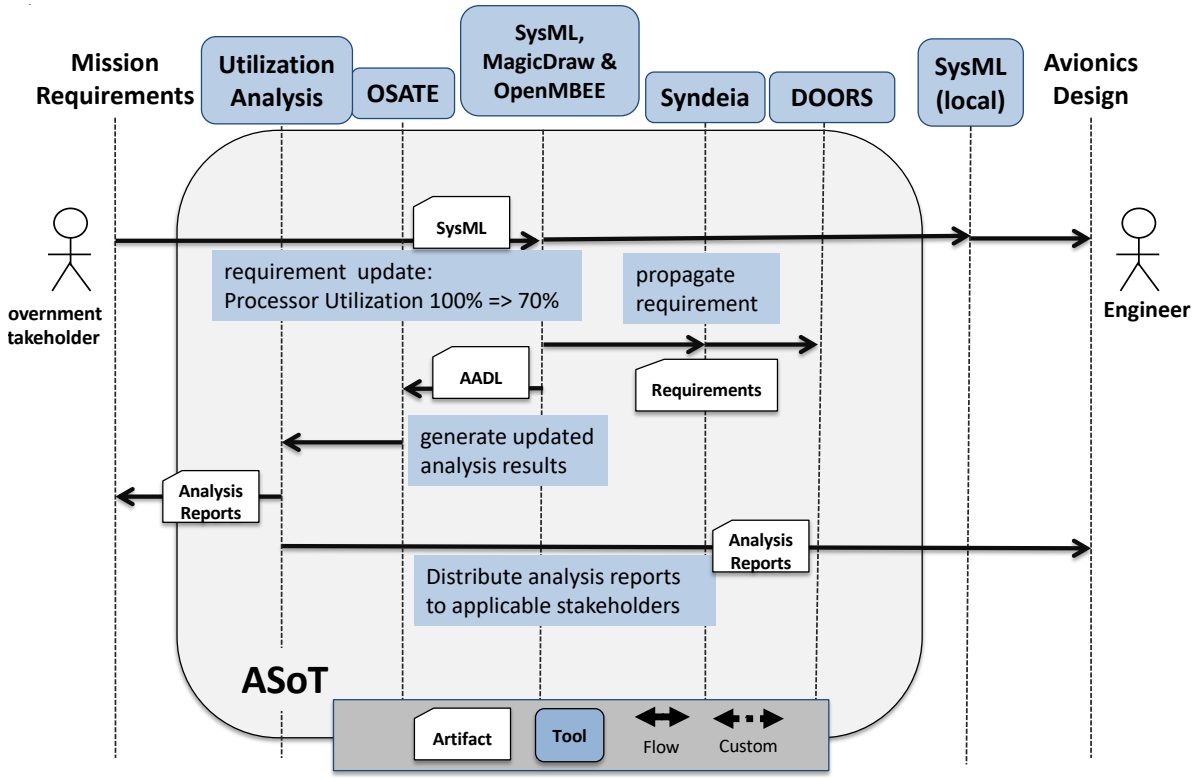

Figure 10: *Demonstration Two Tools and Data Flow*

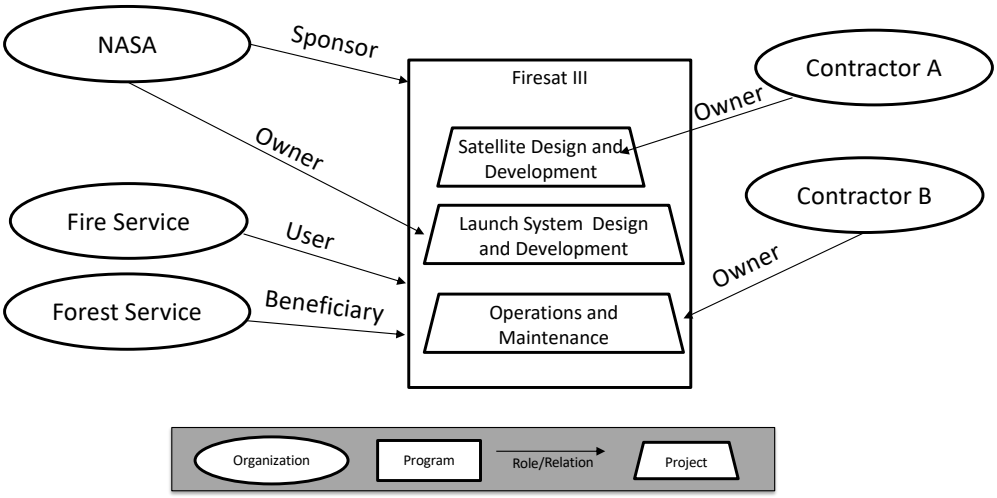

Figure 11: *Demonstration Two Notional Stakeholders and Roles*

automated analysis may be executed on the SysML model, such as system resource utilization analysis. Via OpenMBEE, NASA updates the utilization threshold on the avionics onboard computer from 100% maximum to 70% maximum. The ASoT automatically propagates this update to the SysML model and DOORS and invokes the utilization analysis on the updated model. Updated results of the utilization analysis are collated into a report that is made available via the ASoT dashboard to only the impacted stakeholders.

3.5.2 Demonstration Two Results

The current commercial tools used in the second demonstration are not easy to integrate within an ASoT infrastructure out-of-the-box. In some cases they may be more mature than certain open source tools, but still need further maturation. For this demonstration, we developed custom placeholder technology where we identified significant feature gaps in the existing tools (e.g., the ASoT dashboard).

Successful OSLC Experiments. We have had modest success interacting with DOORS NG using its OSLC API. We are able to create, read, and update requirements in DOORS NG from its OSLC interface, using an OSLC client application that IBM provides. The Syndeia link to DOORS NG predates the OSLC standard, and does not use OSLC for that particular tool synchronization. A detailed discussion of our OSLC exploration is provided in Section 6.

SysML Interoperability. A program must enforce specific SysML modeling conventions for interoperability and re-use of model to be effective. Model interoperability includes model sharing/extending across organizations and across other tools, such as the SysML-AADL bridge. Organization of modeling artifacts and their dependencies within the SysML model is important. Support for OSLC is growing in the cloud based services extended from PLM and modeling environments. Full support of the OSLC standard by tool vendors is not there yet. For example, Cameo Teamwork Cloud does not support service catalogs; Descriptions of modeling elements are available, but operations on the model elements are not available via OSLC (discussed further in Section 6). Security authentication is included in the Teamwork Cloud OSLC implementation.

OpenMBEE. Open source support is usable, but not yet at commercial level. OpenMBEE has a strong user base and active development, but there are considerable start-up costs and minimal installation and user documentation. We pushed the edge of current capabilities for this demo, especially its tight integration with Cameo Teamwork Cloud. Model merging relies on a separate Cameo Teamwork Cloud plugin. OpenMBEE has a plugin for direct interoperability with DOORS, but it is not publically available.

Traceability Objective. Maintaining traceability across tools with different version history management is difficult. There are no industry standards for version control of digital assets. OSLC has a framework in place, but it is incomplete and not currently supported by tool vendors. Often tools have different paradigms in place for version control, reflecting different approaches to workflow management. Bridging the gaps between different paradigms and workflow requires thorough understanding of each approach. Interfacing versioning systems with similar paradigms and APIs is non-trivial, such as connecting git repository version

control with OpenMBEE.

3.5.3 Demonstration Two Technical Notes

- Machines used: Virtual 1, 2, 3, 5, 7, 8 Physical 1
- Connections from demo 1 remain in place.
- VM 4 replaced with VM 5. Windows service created by RLM would not run consistently requiring it to be restarted several times per week. After replacing with a Linux Systemd service the problem disappeared. Moving to Linux required that the license be regenerated by the Intercax help disk.
- OpenMBEE MDK was installed on Virtual Machine (VM) 1 for this demo. MDK connected to the VE via the MMS which ran on Physical Machine 1. Data propagation to or from the SysML Model and the VE was all handled by the MMS.

3.6 Demonstration Three

Demonstration three explored the integration of a "digital thread," connecting the operational and maintenance portion of a product lifecycle with the design tools utilized within an ASoT implementation. Although one could consider a digital twin deployment as part of an ASoT, we have placed the digital twin outside the ASoT boundary. Where the boundary line is drawn impacts who is responsibile for maintaining the ASoT versus maintaining the digital twin. Integration challenges are essentially the same, no matter where the line is drawn.

3.6.1 Demonstration Three Overview

For the third demonstration, we again used the FireSat SysML model as the driving example. In the updated scenario, the digital twin determines that a payload thermal sensor on one of the active FireSat deployments is not meeting mission requirements. In response the ASoT identifies the other subsystems and components impacted by a possible sensor upgrade. The change propagates throughout the design and necessitates a re-evaluation within multiple design domains.

Specifically in this scenario, an upgrade to the thermal sensors necessitates a change to the power requirements of the payload subsystem. As a result, the changes to the power requirements impact the required surface area of the solar panels. The calculations of the solar panels physical dimensions is performed by a separate cost impact analysis team, who for purposes of this demonstration are represented by an Excel spreadsheet. In practice, this cost impact analysis would involve a collection of commercial and/or internally developed tools.

Next in the demonstration scenario, the upgraded sensors also increase the size of the payload data generated, which in turn impacts processor utilization loads, bus utilizations, communication latencies, and potentially the prioritization of onboard resources. The user selects system configuration *options* to consider, forming a *design space*. A series of utilization and

security analyses are re-run over a set of new candidate architectures taken from that design space. The results of the analyses are provided to the user to inform the design decision.

Figure 12 is a sequence diagram that shows the interactions of the demonstration's steps:

1. The ASoT tracks digital twin information and part information in Aras (the PLM tool. The requirements for the part are linked from Aras to DOORS, where they are maintained.
2. The user reports poor signal quality from an individual satellite by creating an error report against the digital twin in Aras.
3. The user uses Neo4j to explore the relationships from that digital twin to find related models and requirements in SysML.
4. The user updates a requirement in the SysML model. The user uses Syndeia to propagae updates to DOORS and to the cost impact analysis (an Excel spreadsheet).
5. The user executes analysis on the updated model and requirements using tools from the CAMET Library.
6. Traceability and analysis results are aggregated from the various tools into a single custom dashboard.
7. The user uses Design Space Investigator (DSI) to evaluate design trades.

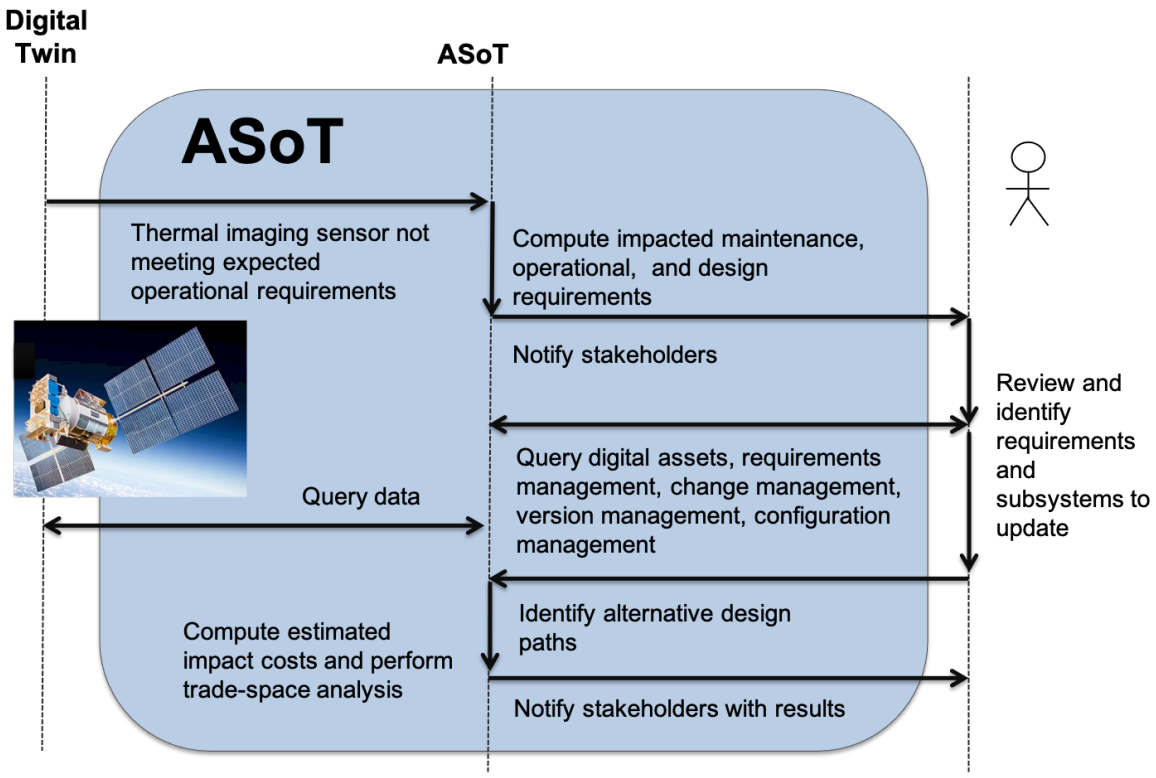

Figure 12: *Sequence Diagram Illustrating the Interactions of Demonstration Three*

For the third demonstration, we utilized the Aras Innovator PLM tool to represent the digital twin. In practice, a digital twin implementation will consist of a large infrastructure including

analysis tools, simualtion tools, and possibily target hardware-in-the-loop. Building such an infrastructure for this study is out-of-scope, so instead we captured a representation of the product data and operational data contained within a digital twin, and used that to exercise the ASoT interfaces and functions. A PLM tool such as Aras Innovator is often used to manage a digital twin.[4]

The requirements updated from the digital twin are maintained in a DOORS database, and translated to the FireSat SysML (MagicDraw) model via the Intercax Syndeia tool. The cost impact analysis represented by an Excel spreadsheet interfaces with the updated requirements also via the FireSat SysML model and Syndeia. The Adventium SysML-to-AADL bridge is employed to convert certain portions of the FireSat design into AADL. The DSI tool allows the user to designate a *design space* in which multiple system configurations can be considered. The DSI analyzer automatically runs analysis tools on each selected configuration in the design space. Once the analyses are performed, the resulting collection of new candiate designs provide a basis for determination of which candiate designs best satsify current requirements. See Figure 13.

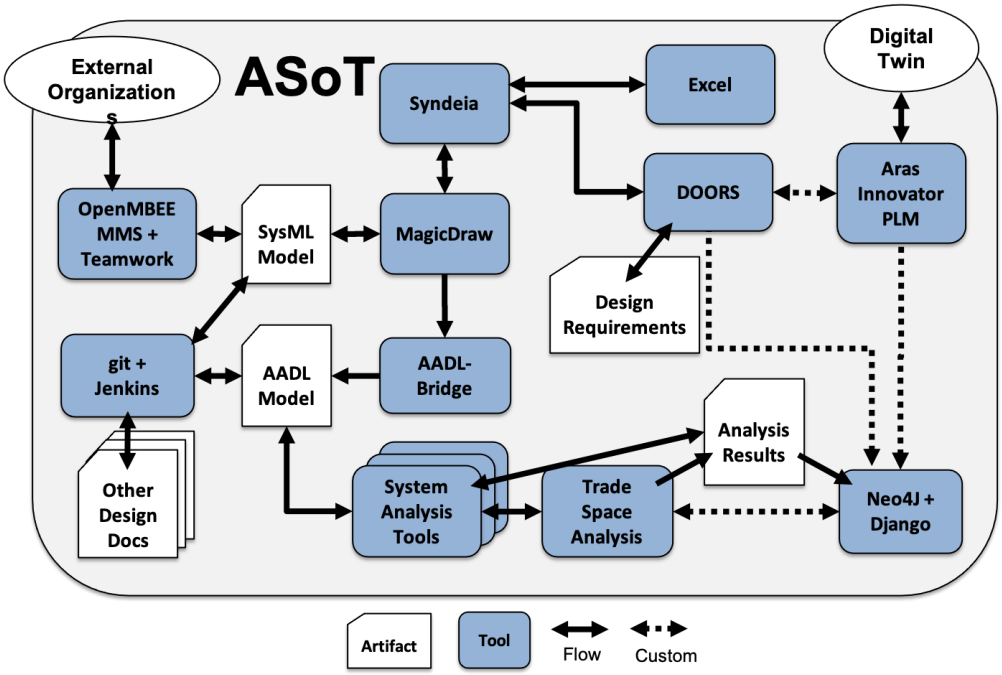

Figure 13: *Demonstration Three Tools and Data Flow*

3.6.2 Demonstration Three Results

The lessons learned from the third demonstration focus on integration challenges.

Model Interoperability. A semantic gap exists for data passed between Digital Twin and ASoT implementations. Consistent rules and format conventions are required to bridge the gap. Maintainers of both the ASoT and the Digital Twin must enforce the rules.

[4]See https://community.aras.com/b/english/posts/a-how-to-guide-to-building-digital-twins.

Model Synchronization. Timely model synchronization is a challenge, especially between different languages and tool vendors. Fully automated updates currently require custom infrastructure. Cloud-enabled or server-based tools complicates the integration further. Manual synchronization is different for each connection. OSLC is designed primarily for "pull" queries. Synchronization requires both push and pull data operations, which is not how OSLC is currently employed.

Data Aggregation. Design and product information in practice is distributed across multiple tools, vendors, repositories, and legacy development infrastructures. Many PLM vendors are extending their internal tools capabilities instead of providing mature and useful APIs to integrate with third-party tools and repositories. Standards like OSLC improve integration, but are not fully supported throughout commercial offerings. A tool may support OSLC, but it may not support the entire breadth of functionality of OSLC. For the tools that support cross-vendor integration, such as DOORS and Syndeia, the interfaces often require special maintenance and model formats.

Data Visualization. Several options are available to visualize data once it is aggregated. We used the graph database Neo4j, but found that we had to do some customization to collect and correlate data from all of our tools for display in Neo4j. Everything in Aras is stored as an item, including relationships.[5] We parsed the Aras items through the Aras REST API and we uploaded information about those items to to Neo4j. We also parsed the DOORS items and uploaded them to Neo4j. We augmented the requirements in Aras to include the URIs of linked requirements in DOORS. If a link existed in an Aras requirement to a DOORS requirement, we created a link in Neo4j as well. Figure 14 shows part of the graph tracked by Neo4j.

3.6.3 Demonstration Three Technical Notes

- Machines used: Virtual 1, 2, 3, 5, 6, 7, 8 Physical 1, 2
- Connections from demo 1 and 2 remain in place.
- Link between ARAS and DOORS facilitated by OSLC connection to Django on VM 7.

3.7 Overall Lessons Learned

We learned that Army stakeholders need to discover information distributed available across tools and repositories to make informed decisions and reduce duplicate work. We learned that Army stakeholders, together with industry, need to share information in a comprehensive and controlled manner. We learned that Army stakeholders need to maintain information and information traceability over time and across tool vendors. We concluded that to address these needs, the army should encourage standards adoption by tool providers, emphasize data rights management in both technical and administrative fields, and should use and invest in open standards. We found that Model Based Systems Engineering (MBSE) and ASoT

[5] In DOORS NG, everything is an "artifact." In Aras everything is an "item." Using the ASoT ontology helps us keep these tool-specific terms from biasing the language used in our requirements.

Figure 14: *Part of the traceability graph generated to show links to the Payload Sensor explored in demonstration 3. The green Payload Sensor node is an Aras digital twin, the pink Payload Sensor node is an Aras part, the yellow nodes are Aras requirements, and the blue nodes are DOORS requirements.*

are complementary, and that connectivity tools provide a scalable and flexible approach to providing ASoT capabilities.

Throughout the requirements elicitation and demonstration process we identified topics for which we determined that the available tools, standards, and processes were insufficient or in need of additional research. For each of these we conducted targeted research thrusts in parallel with the demonstrations:

- **Object Equivalence Tracking:** We learned that tracking the *identity* of an object across tools, repositories, and workflows is challenging. Note that (as discussed in the ASoT ontology in Appendix B) an object is not the same thing as a model or as a digital artifact. We found that tracking object metadata in a separate database was the best approach; our report on this topic is provided in Section 4.
- **Procurement:** We learned from discussions with stakeholders that existing DoD procurement guidance does not adequately address the data rights needs of a project using MBSE and an ASoT. We concluded that increased up-front attention is needed to ASoT procurement concerns, particularly involving data rights strategy. Our report and guidance on this topic is provided in Section 5.
- **Available Tools:** We learned that a varity of commercial and open source tools are available that meet the ASoT capabilities described by the ASoT requirements, but that the capabilities covered and maturity vary significantly from tool to tool.
- **OSLC Survey:** We found that OSLC meets its advertized potential to address tool

interoperability issues. OSLC is a service-oriented standard for exchanging data between tools. An introduction to OSLC and a review of the OSLC services we used in this study is provided in Section 6.

- **Provenance:** We learned that design and logistical effort is required to track origin and evoluation of information in an ASoT; no tools were able to provide provenance information across all of our repositories out-of-the-box. Even our relatively small demonstrations involved design items that were represented in four or more tools or languages. Section 7 explores this topic in further depth.
- **Capstone Lessons Performer Learned:** From our discussions with JMR MSAD Capstone performers we learned that many of them encountered problems related to ASoT capabilities. In particularly, many performs had difficulty *discovering* what data they had and how it was related to other data. Section 8 describes these issues.

4 Object Equivalence Tracking

Equivalence tracking in the context of an ASoT deployment is the process of defining and storing relationships between digital artifacts or pieces of digital artifacts that represent the same described objects. As a simple example, suppose a SysML model includes a block representing a waypoint manager subsystem in the system architecture, while a separate AADL model includes a system component that represents the same subsystem in the design architecture. In this case the SysML block and the AADL system component have an equivalency relationship. A described object, for purposes on this discussion, is any entity in a product design that has well defined and fixed, architectural, logical, functional, or physical boundaries.[6] Figure 15 places the terms *described object* and *equivalency relationship* within a larger taxonomy for representing the elements of an ASoT. Modeling artifacts that have equivalency are always separate and unique digital artifacts. Table 2 gives examples of described objects.

The basic concept behind equivalency relationships is straightforward. In practice, however, the definition of equivalency relationships against real-world models is much less clear-cut [1, 16]. The two main overriding issues will be described here, and explored further in subsequent sections.

First, one of the described objects in the equivalency relationship may be defined as an authoritative source-of-true, which naturally leads to further questions. If the authoritative side of the relationship is modified, is the other side of the relationship automatically modified as well? What if the non-authoritative side is modified first? Often the propagation of a change depends on the context surrounding the modification itself, such as who has made the change or the type of change.

Second, in practice the two sides of an equivalency relationship are almost always imprecise, especially when the equivalency exists across different modeling domains or levels of abstraction. Several conditions exist that create imprecise relationships:

[6] This definition is informed by Pollari and Shaw, with terminology adjusted to address both model artifacts and traditional code and documentation artifacts. See [32].

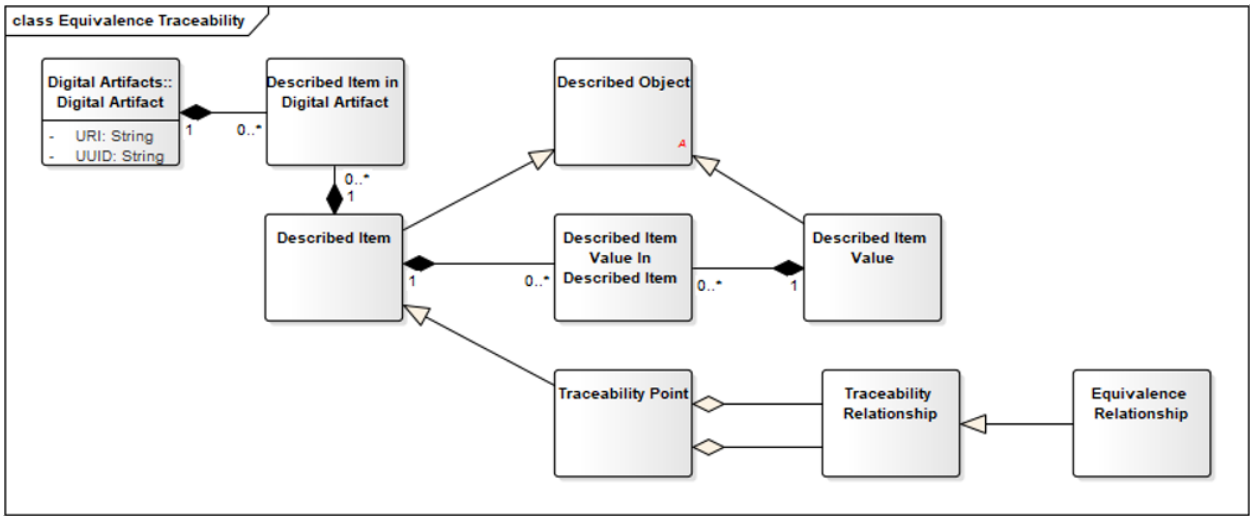

Figure 15: **From the ASoT taxonomy, a digital artifact describes objects that are part of a design.** An equivalency relationship defines a specific type of association between two different described objects.

Tool / Domain	Name	Description
SysML, System Modeling	Model Element	The basic entity in the SysML modeling language. Example Model Element types include packages, blocks, and activities.
FACE Modeling Standard	Segment	Basic components of the FACE architecture, such as the PCS, TSS, PSSS, and the IOSS.
Simulink, Behavior Modeling	Blocks	The basic component of a Simulink model. Types of blocks include Subsystems, Models, and Variant Subsystems.
Creo, Physical Modeling (CAD)	Objects	Smallest named component in a Creo physical model stored in its own file.

Table 2: Examples of Described Objects

- Not all equivalency relationships have one-to-one cardinality. A "bus" element in a system model, for example, could represent a set of multiple wires in a physical model.
- Not all equivalency relationships are obvious. A subject matter expert (SME) may not have the visibility to see how a design architectural components is equivalent to a particular set of sub-functions in a behavioral model.
- Many equivalency relationships are actually partial-equivalency relationships, where the two sides of the relationship both represent similar and/or overlapping described

objects or concepts that are not strictly the same. For example, a component block in a SysML model representing the logical boundaries of a subsystem may have a partial equivalency relationship with a Simulink model that represents the behavior behind the logical block.

- Large programs with many stakeholders, performers, and organizations always risk inserting descrepensies in source material that is passed around and duplicated in multiple repositories. This creates confusion when establishing and maintaining equivalency relationships across different versions of distributed source material.

For the design and maintenance of an ASoT, it is important to establish a consistent approach towards the definition of equivalency relationships within a Model Based Engineering Environment (MBEE). Specifically, a rigorous process must be in place to establish equivalency relationships, and to modify or remove equivalency relationships when its associated described objects have been changed, versioned, or removed. Without a consistent process, equaivalency relationships can become confused, corrupted, or lost, leading to unreliable traceability throughout the ASoT implementation.

Equivalency relationships are just one of many types of relationships defined and stored within an ASoT deployment. For example, a high-level requirement may be broken down into multiple lower-level requirements, which then are satisfied by specific functional and architectural components in a design model. The refinement, compositional, and satisfies relationships, while also necessary for traceability, are not equivalency relationships.

4.1 Technical Approach

The problem of how to represent equivalency relationships can be framed in the context of database design.[7] Most, if not all, digital artifacts in an ASoT utilize unique identifiers, a string of a fixed length that is not duplicated as an identifier anywhere else in its local tool environment. Often additional steps are required to translate or store identifiers across different tools within the ASoT implementation. An equivalency relationship then may be represented as one of the following basic formats, dependent on how the unique identifiers are matched between modeling artifacts (see Figure 16).

The topmost example in the Figure 16 shows an equivalency relationship defined via a single shared identifier maintained as a property of the two modeling artifacts. The example second from the top shows equivalency defined by storing the unique identifier of one component as a property on the other component. In this scenario, the traceability available to this identifier matching is traversed in only one direction, because the "destination" component does not include a reciprocal property linking to the identifier of the "source" component. This scenario occurs when the "destination" component resides within a design tool that does not readily facilitate interoperability, such as a domain-specific analysis tool that is accessed via a command-line interface.

In the third example in Figure 16, both described objects in the equivalency have their own identifiers to define their uniqueness, while a separate unique identifier is used to define the

[7] A large body of literature already exists on database schema mining, mapping, matching, alignment, merging, etc. The details of these topics are outside the scope of this whitepaper.

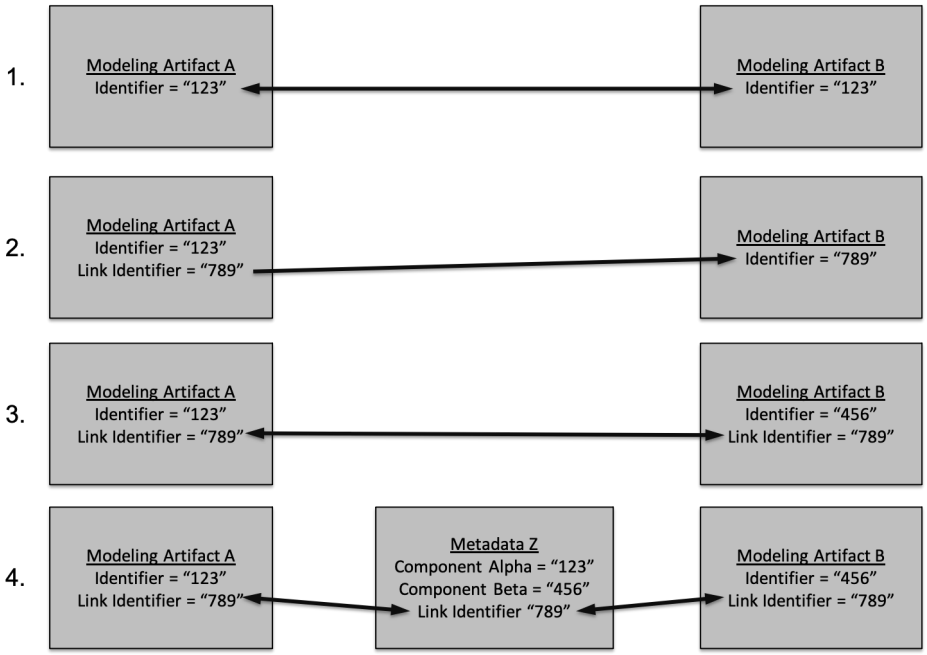

Figure 16: The different approaches for defining equivalency relationships between modeling artifacts.

equivalency itself, linking the two components. The final example takes that concept to the next step, and stores the identifier establishing the equivalency relationship within its own separate metadata entity. The metadata entity representation has a number of advantages for an ASoT implementation:

- The metadata entity can store additional attributes that further describe the nature of the relationship, such as which side is the authoritative side, change history of the relationship, and other domain specific details (e.g., unit conversions or other types of transforms).
- The metadata entity is a more scalable format to represent more complex relationships, such as one-to-many, many-to-many, hierarchical, or inherited relationships.
- Identifiers are unique within their own tools and repositories, but are not typically guaranteed to be unique across the entire ASoT. Unique identifiers generated in MagicDraw, for example, are not checked for uniqueness throughout the rest of the ASoT in other design tools. Consequently, different described objects in an ASoT can in theory share the same identifier, while not sharing an equivalency relationship, thereby creating potential confusion. A metadata representation alleviates this problem by establishing the equivalency relationship without relying on the global uniqueness of the identifiers of each individual described object. Uniqueness, in this scenario, need only be established between metadata objects.
- A database of metadata entities representing equivalency relationships is a powerful tool in an ASoT deployment. From such a database, equivalency relationships may be evaluated, visualized, and used to automate or semi-automate the propagation of

changes when particular design components are modified.

4.2 Examples

For concrete examples of the equivalency tracking in operation, first consider linking equivalent described objects across different modeling languages. The commercial SysML modeling environments MagicDraw and Enterprise Architect both support auto-generated unique identifiers for each of its modeling artifacts. The SysML to AADL Profile and Bridge Tool (Adventium Labs) translates particular modeling artifacts into the AADL modeling language, according to modeling conventions and translation rules captured in the profile [37]. For both the MagicDraw and Enterprise Architect implementations of the profile, the unique identifiers established by the SysML environments are translated verbatim to their equivalent described objects in the auto-generated AADL model, captured in the AADL via a comment line associated with the equivalent object in the AADL source code. While matching and capturing the unique identifiers within a comment line satisfies the basic need for traceability, ideally the unique identifier is captured in a property applied to the various AADL modeling artifacts, so that the unique identifier can be more readily reapplied to other ASoT operations that propagate from an AADL model, such as architectural analysis. Extensions to AADL have been recommended that include such a property on AADL modeling artifacts to represent unique identifiers.

The commercial tool Syndeia by Intercax[8] is designed to provide traceability between heterogeneous tools within a Model Based Engineering (MBE) environment. Syndeia can, for example, define an equivalency relationship between a described object in a SysML model with a component defined in the requirement management tool DOORS.[9] The equivalency relationship is then stored in database managed by Syndeia, along with the other relationship connections defined across the different tools in an ASoT. This database is an implementation of the metadata approach for equivalency tracking described above, and provides within an ASoT the means to easily track and visualize the propagation of equivalency relationships throughout a system design. Syndeia supports similar tool integration with various CAD tools and Microsoft Excel.

4.3 Tool Heterogeneity

The design tools employed in an ASoT largely drive the decision as to what format is used to represent equivalency and how the relationships are maintained. In practice, an ASoT with any complexity will have multiple heterogeneous tools, i.e., tools from different vendors that are not immediately interoperable. As a result, the ASoT developers will have to tailor the approach around their environment's tools.

[8] http://intercax.com/products/syndeia

[9] https://www.ibm.com/support/knowledgecenter/en/SSYQBZ_9.7.0/com.ibm.doors.requirements.doc/topics/c_welcome.html

Favored Tools For traceability purposes, the most advantageous approach is to abstract equivalency relationships into metadata entities that are stored in a separate database.[10] Consequently, ASoT developers should favor design tools that:

- Accommodate the creation of user-defined unique identifiers for the purposes of representing a specific traceability link, and/or
- Directly support maintenance of component relationship metadata.

Most modern modeling languages, along with state-of-the-art product design tools provide the means to create special unique identifier properties on modeling components used to trace an equivalency relationship to another modeling component, within its own direct tool environment or with another design tool. Emerging tools such as Intercax Syndeia are specifically designed to maintain and visualize equivalency relationships as metadata between different tools.[11]

4.4 Maintenance Process

Creation and maintenance of equivalency tracking data in an ASoT will vary from one implementation to the next. A formal process must be in place to review and manage equivalency relationships.

Many of the obvious equivalency relationships in an ASoT may require little to no attention, and may be fully automated, so that a change to one side automatically updates the other side. As equivalency relationships become more imprecise, however, the maintenance requires a human in the loop. This could be as simple as an ASoT administrator following an established checklist to determine the correct action (determined by the formal process), or it could involve a committee of system engineers and SME to determine the best course. We recommend setting up an ASoT with a system of alerts, to automatically notify certain personnel that an equivalency relationship requires attention. Nevertheless, due to the complex nature of cross-domain modeling and heterogeneous tooling, it is likely that the relationships representing partial-equivalency will require hands-on review.

Timing is a factor as well. The boundaries of a described object may change as a design matures, and an equivalency (or partial equivalency) that spans design domains may alter significantly, or become inapplicable over time. The formal review process should include the opportunity to review these specific, complex relationships at designated design milestones.

4.5 Recommendations

In summary, we list the following recommendations:

- Employ unique identifiers for every digital artifact in the ASoT, including modeling artifacts.

[10] These databases are also referred to as relationship management databases.

[11] Commercial tools typically follow their own custom format for representing metadata. Standardization of metadata formats across modeling languages and tools is a related topic of study for ASoT design and deployment.

- Whenever possible, select tools for the ASoT that accommodate a metadata approach to storing equivalency relationships. If equivalency relationships are not implemented using a metadata approach, then additional steps may be required to ensure that an identifier is globally unqiue across the entire ASoT implementation.
- Use and consistently apply a standard format for unique identifier strings, such as RFC 4122, which describes the format and generation of UUIDs.
- When a unique identifier is defined on a modeling artifact, do not use the identifier for any other purpose than unique traceability. E.g., do not use a component name label also as a unique identifier.
- A unique identifier cannot be modified once the described object is created. Stability of the identifier uniqueness is relied upon from both within in the host tool domain as well as through the ASoT deployment. Equivalency tracking relies on unique identifier stability.[12]
- Once an equivalency relationship is established, it is acceptable to redeploy the identifier of a described object to create new equivalency relationships with other described objects. For example, a unique identifier property on a SysML block that defines its equivalency with a system component in an AADL model may be re-used to establish an additional equivalency with a component definition in a DOORS project. Reapplying unique identifiers in this way simplifies the consistency across the tools, and improves scalability. ASoT maintainers do not need to generate additional identifiers for each tool added to ASoT as a result.
- Establish a formal process to review and update equivalency relationships throughout the ASoT, especially complicated relationships that cross design domains or change over time as the design matures.

Furthermore, we recommend a new standard, (optional) property set within AADL, proposed as an erratum on the core language property set, in order to facilitate integration of AADL models with a larger ASoT:

```
UUID : aadlstring applies to (
           thread, thread group, process, data,
           subprogram, processor, memory, device, bus, system, virtual
           processor, virtual bus, abstract, port, access, parameter,
           end to end flow, flow sink, flow source, flow path, call sequence,
           mode, feature, feature group);
```

5 Procurement Guidance

This section provides procurement recommendations for the Government to consider when acquiring systems from suppliers who will interact with a government-owned ASoT.

We draw our recommendations from the same source material used to generate the ASoT requirements (see Appendix A), that is user stories, interviews with industry and U.S. gov-

[12] Refer to discussion of URL stability in [12] within the larger context of the OSLC standard.

ernment subject matter experts, surveys on existing practice, and prior research. We assume a government acquisition involving multiple stakeholders (e.g., multiple government program managers, system integrators, component suppliers, etc.) that will apply the principles of MBSE and the Architecture-Centric Virtual Integration Process (ACVIP). We examined those sources for situations where, in order for the Government to achieve its objectives, the Government must acquire something from those suppliers for its ASoT. For example, in order for the Government to enforce authorized access to the ASoT's digital artifacts, the Government must acquire from its suppliers the data rights associated with those digital artifacts.

We organized this section as follows. In Section 5.1, we summarize our study of acquisition planning documents relevant to procuring the ASoT. In Section 5.2, we list recommendations for the adaptations to these documents when procuring the ASoT. In Section 5.3, we list related acquisition guidance that will also inform the Government's use of the ASoT.

5.1 Acquisition Planning

At the acquisition planning stage, the Government should identify documents that may need to be tailored to include procuring the ASoT for a program. Internal government documents, such as the System Engineering Plan (SEP), should address engineering tools and data delivery methods including products and licenses required for the ASoT. The technical review section of the SEP should address how information such as models in the ASoT will be utilized for review and document generation.

The Government should include the requirements for the use of an ASoT in a program during the Request for Proposal (RFP) planning stages. Specifically, ASoT requirements are relevant to the following sections of the RFP.

- Section L: Instructions, Conditions and Notices to Offerors: Require the Offerer to address the use of an ASoT in a section of the technical volume. Include requirements for tools or licenses required for the ASoT in this section. Define the use of an ASoT for delivery of technical data, iterations of models and analysis, or collaboration required. Address Data Rights requirements for artifacts made as deliveries with the ASoT per Contract Data Requirements List (CDRL) requirements. The response by the Offerer in this section of the proposal will describe the Offeror's understanding of the use of an ASoT.
- Section M: Evaluation Factors for Award: The Government will evaluate the proposal, in part, based on ability of the Offeror to demonstrate anticipated use of an ASoT with consideration the ASoT interface, data rights of digital artifacts delivered or maintained within the ASoT which allow for the Government to meet re-competition goals of subsystems or components.
- CDRL Definition and Data Item Description (DID): The CDRL and DID should be tailored to include how the ASoT would be used or if it is to be used as a method for providing deliveries, defined by the CDRL, to the Government. CDRL answer to whom, how and when data is to be delivered. The DID define the format or contents. Both should be tailored to incorporate the use of the ASoT along with any other

requirements.

- Statement of Work (SOW): Add language regarding requirements of the ASoT in the SOW or Performance Work Statement (PWS)
- Require offerors to address use of ASoT in the System Engineering Management Plan (SEMP) or other relevant management plan deliverable.

5.2 Recommendations

We list below our recommendations for the Government's procurement authority. The recommendations are of two types: things the Government should acquire and store in the Government's ASoT and things the Government should communicate to other stakeholders who will access that ASoT. For guidance about acquiring the ASoT itself, see the ASoT requirements.

5.2.1 What to Acquire

1. Acquire the digital artifacts the Government needs to approve and recompete the fielded system.

   Digital artifacts that the Government requires and recompete the system should exist within the ASoT that is under government control. Mark the digital artifacts approved for integration, and associate with each digital artifact the evidence that justifies that approval. Track the system throughout its lifecycle to identify the as-approved, as-built, as-maintained, and as-destroyed versions of the system. Acquire models to represent legacy components.

2. Acquire the data rights for each digital artifact that the Government stores in the ASoT.

   Consider Technical Data (TD), Computer Software (CS), and Computer Software Documentation (CSD) data rights and communicate the government desired rights in the solicitation for each procurement based on the TD and CS strategy according to Defense Federal Acquisition Regulations Supplement (DFARS) 207.106 in the Acquisition Planning Phase of the procurement. The Statement of Work and CDRL should identify negotiated data rights for each digital artifact to be delivered in the ASoT. The Government should require digital artifacts be marked with appropriate data rights legends and Defense Technical Information Center (DTIC) Distribution Statements to ensure proper control of digital artifacts within the ASoT. Data rights markings on the digital artifacts should be consistent with negotiated data rights listed in the Statement of Work and DTIC Distribution Statements identified in the CDRL. If contractor delivers TD, CS, and CSD to the Government with less than Unlimited Rights or Government Purpose Rights (GPR), the ASoT requirements include enforcing access controls according to those data rights, such as restricting access to digital artifacts according to contractual agreement and national security guidance. Contractual agreements should clarify delivery access controls. Contracts should specify the national security guidance to apply according to the digital artifacts acquired.

3. Acquire required metadata[13] for each digital artifact needed in order to support access control, search, approval, and recompete functions.

   Develop and adhere to a standard for the metadata collected. The ASoT requirements call out collecting artifact expiration dates, country-of-origin, country-of-delivery, information criticality, non-functional requirements such as manufacturing constraints, and cost and scheduling metrics. The ASoT requirements also call out evidence to demonstrate the provenance of digital artifacts, such as the tools used to build or generate the artifact, the contract guidance (e.g., FACE guidance) used to produce the artifact, marking and licensing information (even from previous contracts), template models used to produce the artifact, and analysis results and certification results associated with specific versions of the artifact.

   Metadata acquisition should be a flow-down requirement for the procurement. For example, if the Government acquires the artifact from a system integrator, who acquired the artifact from a component supplier, acquire the metadata during the Government's acquisition from the integrator, and the integrator, in turn, should acquire the metadata during its acquisition from the component supplier.

5.2.2 What to Communicate

1. Communicate the types of digital artifacts that the government ASoT will manage.

   The types are the digital artifacts required for recommendation 1 in Section 5.2.1, i.e., those artifacts required to approve and recompete the system. The ASoT requirements provide examples such as models associated with legacy components, government template models, models developed under the performance of this contract, and government-furnished information.

2. Communicate the approved representations for each type of digital artifact.

   Adopt and adhere to a set of approved representations (e.g., languages, formats) to facilitate interoperability between different ASoTs and to simplify the recompete of any digital artifact.

3. Communicate the security policies that will enforce authorized access to digital artifacts stored in the ASoT.

   Stakeholders contributing digital artifacts to the ASoT should understand how the ASoT will protect those artifacts. The ASoT requirements call for security policies addressing, for example, information sensitivity, contractual rights, and organizational role.

4. Communicate the change management system within the ASoT that will manage digital artifacts.

   The ASoT requirements call for a change management system, including each stakholder's role therein, that includes issue tracking and resolution, comparing and merging different versions of an artifact, and staging proposed changes for approval before submission.

[13] See examples in Appendix A.

5. Communicate the planned execution of Government-selected operations over digital artifacts.

   The ASoT requirements include calls for the ASoT to query and visualize artifact associations, to schedule automatic execution of user-defined analysis over artifacts, to notify the artifact owner of changes, and to facilitate translation of artifacts to alternate representations or languages.

6. Communicate the Government approved tools that the Government will require stakeholders to use.

   Communicate these selections in the solicitation and/or Statement of Work. The ASoT requirements call out the need for a registry of approved modeling and analysis tools and the need to store the model analysis results in a systemmatic way that supports examination by subject matter experts.

7. Communicate the interfaces approved for access to other stakeholder ASoTs, such as OSLC.

   Our tool survey revealed that of the two common approaches for tool integration (either build a custom interface or build to a common standard), building to a common standard scales and better supports future capabilities.

   If possible, stipulate the protocol for using those interfaces.[14] If the other stakeholder is also a Government entity, then both government entities should agree on their shared interfaces. Note that any interface may require data conversion between different representations of the digital artifact, and the Government should require that conversion as necessary.

5.3 Related Guidance

Related acquisition studies and acquisition guidance will also inform the Government's procurement and use of the ASoT.

- **Enterprise System-of-Systems Model for Digital-Thread Enabled Acquisition** [18]

  This report documents the results of a research project to understand how a Digital Engineering strategy "might evolve and change the way the DoD conducts acquisition of new systems and supports existing systems." The report identifies recommendations that could impact acquisition processes, including:

  - Identify and develop metrics appropriate to Digital Engineering. The Government should acquire with each digital artifact any context necessary to calculate the metrics.
  - Develop a rigorous approach to verify, validate, and accredit the models to incorporate into the ASoT. Acquisition processes should acknowledge this approach.

[14] For example, should the Government both pull from and push to other stakeholder ASoTs? What do these actions imply contractually? For example, should a contractor with component updates first obtain approval to push those updates to the Government ASoT?

  - Standardize metadata to incorporate into the ASoT.

- **DoD Open Systems Architecture (OSA) Contract Guidebook for Program Managers (DoD OSA Guidebook)** [24]

  This guidebook is designed for Program Managers and acquisition professionals to use when preparing contractual language for a solicitation or request for proposal to incorporate OSA technical and business principles. The purpose of Open System Architecture is to provide the Government with systems that have severable modules that the Government can re-compete throughout the life cycle of the system allowing for independence from vendor lock. The OSA approach incorporates an open business model, which provides transparency of design elements, modular interoperable systems with the intent to support component modification or replacement by different vendors throughout the life cycle of the system. Reuse and portability are key tenets in OSA.

- **Future Airborne Capability Environment (FACE) Contract Guide** [13]

  This is a reference guide assisting Program Managers and acquisition professionals to apply FACE Standards and requirements into a solicitation or request for proposal. It serves as a supplemental guide to the DoD OSA Guidebook where FACE standards are a requirement of a program. The FACE approach incorporates technical practices and software standards to promote development of Reusable Software Components (RSC) and portable software components. The FACE Technical Standard is publicly available. Like OSA, the intent of these standards is to promote replacement, reusability, and sustainability, avoiding vendor lock throughout the lifecycle of the system. The FACE Contract guide, although specific to FACE Standards, provides significant detail to assist the PM in the development of a Data Rights strategy for Technical Data and Computer Software. It also provides sample wording and commentary for sections of the solicitation or request for proposal.

- **Architecture-Centric Virtual Integration Process (ACVIP) Acquisition Management (AAM) Handbook with the Architecture Analysis & Design Language (AADL)** [15]

  This handbook provides program managers and acquisition professionals with guidance to add ACVIP support for aircraft systems acquisitions. ACVIP supports models and analysis throughout the development of the system with a collaborative and iterative approach. Major acquisitions require adherence to DoDI Number 5000.02, which is a structure of a program which require review points (System Requirements Review – SRR, Critical Design Review -CDR, Preliminary Design Review – PDR, etc.) These milestone meetings tend to map into a Waterfall type process described in software development. This handbook supplies the acquisition team and program managers with suggested structure of what to ask for from contractors and contractual items to consider in order to implement ACVIP into a traditional acquisition lifecycle process. The handbook makes suggestions for how models are to be reviewed at each of these traditional milestone reviews.

- **DoD Contracting Considerations for AGILE Solutions v1** [26]

  This document is specifically designed for procurements of AGILE software systems

and is designed to help programs transition mindset and contract documentation from a waterfall development to an Agile, iterative, rapid development approach. Like the ACVIP Handbook, the document describes how to incorporate the AGILE process when the program is bound to the (DoDI) Number 5000.02 requirements.

- **Army Military Airworthiness Certification Criteria (AMACC)** [7]

  This criteria defines the safety and airworthiness qualification requirements that will inform the system's approval to operate. In particular, the AMACC calls out digital and other artifacts, in support of those requirements, that the Government's ASoT will need to collect and store, such as design criteria, technical drawings, methods, processes, tools, manuals, checklists and so on requiring configuration management and review.

5.4 Conclusions

The primary challenges facing the government procurement authority for using an ASoT during a system acquisition are to identify, acquire, and manage the required digital artifacts, their approved representations, the approved tools to analyze those artifacts, and the necessary metadata to demonstrate provenance and make a convincing argument for system approval. Additional challenges are to acquire and manage the necessary context and metadata to support recompete for any given digital artifact. The last set of challenges are to design the ASoT and communicate its visible behavior to other stakeholders while satisfying both the government requirements and the stakeholders' expectations.

6 OSLC Survey

6.1 Abstract

OSLC is a standard (technically a set of standards) for defining APIs for tools in a variety of domains. OSLC is based on REST and Resource Description Framework (RDF). In the ASoT study we examined the OSLC core standard and OSLC standards for requirements and architecture. We evaluated the OSLC implementations of DOORS NG and Cameo Teamwork Cloud. We found that although both tools "implement" OSLC, the DOORS implementation of OSLC is extensive, whereas the Cameo Teamwork Cloud implementation is limited. OSLC has potential to improve tool interoperability in an ASoT but will require additional research, maturation, and adoption to be viable for use in large-scale, heterogeneous tool environments. We found that claims of OSLC conformance are insufficient for understanding the extent to which OSLC is implemented, and we propose a method for evaluating OSLC conformance.

6.2 Introduction

The expected audience for this paper is system engineers and Information Technology (IT) personnel who are considering which tools to incorporate into an ASoT and how to integrate those tools. We assume readers are aware of the role of OSLC as a standard for tool interoperability and integration, but are not familiar with the details of its specification,

implementation, and use. This paper provides an introduction to the OSLC specification in the context of its potential use in an ASoT.

Recommended Reading We encourage readers who are unfamiliar with OSLC to review the OSLC primer [36]. The OSLC Primer recommends familiarity with basic web technologies such as HTTP, RDF and Linked Data. The OSLC Primer cites W3C primers on RDF and Turtle (Turtle is one of several standard text formats for RDF documents) and a tutorial on linked data concepts, all of which will aid in understanding our findings.

Goals of OSLC To meet engineering objectives an organization may need to use multiple tool environments. For example, an organization might use MagicDraw for a modeling environment, IBM DOORS for requirements management, and IBM Rational Change Management for change management. In any significant engineering effort there is a vast amount of data, some of it overlapping in representation and storage (for example, a requirement may appear in DOORS and in MagicDraw). The problem is to integrate this data so that operations can be distributed across data relationships. For example, an organization may wish to integrate MagicDraw and DOORS by enabling access to DOORS information from the MagicDraw tool environment. OSLC is one approach to enable this integration. In demonstration two (see Section 3.5) we demonstrated use of Intercax Syndeia to transfer requirements between DOORS and MagicDraw. The approach shown via DOORS-MagicDraw connectivity could be extended to manage relationships between any number of tools providing OSLC servers.

Many tools provide APIs for external access to their data, but these APIs are often proprietary or highly tailored to a particular tool. Standards like Extensible Markup Language (XML) or JavaScript Object Notation (JSON) help address this problem by simplifying the process of parsing or encoding data, but XML and JSON only specify the format of data, not the *semantic* information; just because a tool has a JSON-based API does not mean the process of integrating it into a workflow is straightforward. OSLC addresses this challenge by applying *semantic* structure (in the form of OSLC *domains*).

Tools that implement OSLC provide an API that can respond to queries to create, read, update, or delete information. OSLC standards define a simplified query mechanism.[15] The structure of data provided by an OSLC server is defined by *shapes*, which can also be considered as constraints on a graph of linked data. OSLC standards define a simplified shape specification and document validation mechanism. Shapes can require content and constrain content of known types, and providing a shape specification from an OSLC server to a client can provide the client some meta-data about the content of RDF documents to which those shapes apply.

OSLC Standards OSLC standards are published in natural language form and have accompanying Turtle files that define standard OSLC vocabularies (standard classes and

[15]The OSLC requires vendors to provide a basic query capability without requiring the burden of a full SPARQL Protocol and RDF Query Language (SPARQL) query engine and server [11].

properties) as Web Ontology Language (OWL) Full *ontologies*.[16] An ontology can provide meta-data about documents and allow *reasoner* software to infer additional information that is not explicitly declared in a document [38].[17] The extent to which reasoning is possible depends on the extent to which a document uses any standard vocabularies and the extent to which a reasoner can infer additional meta-data and content from it. Although OWL Full is more expressive than OWL DL (a subset of OWL Full), the semantics for OWL Full are much more complicated (non-standard, undecidable) and lack standard reasoning/entailment regimes. An initial examination of the OSLC Core ontology did not identify any reason why this could not be rewritten as an OWL DL ontology to enable standard OWL DL reasoning.

6.3 Key OSLC Terminology

Resource Description Framework (RDF) RDF is a central underlying technology in OSLC. At its heart, RDF is a document format that records information as a graph. Unlike XML or HyperText Markup Language (HTML), there is no single standard textual representation for an RDF document; there are several. The two that seem used by the OSLC community are RDF/XML (which is the W3C preferred and most widely-used format to exchange RDF documents between tools) and Turtle (widely-supported and preferred by many for human readability, with exceptions when we get to OWL). RDF documents as a whole borrow some concepts from XML. Every document has a globally unique namespace (uniqueness relying on Internationalized Resource Identifiers (IRI)/Uniform Resource Identifier (URI) standards, see below). Entities (resources) named in a document have a globally unique name that begins with the namespace of the document that declares or defines that entity/resource. A document can name entities/resources that are defined in other documents by using the namespaces of those other documents when naming those entities/resources. For readability purposes, a document can include a list of abbreviations for namespaces that it uses, which are called prefixes. RDF literals (values from mathematical domains) use XML literal syntax.

IRI An RDF document consists of a set of triples of the form "subject predicate object." These are also called statements. A predicate is also often called a property. All three fields in a triple are generally represented as IRIs (IRIs, a generalized syntax for Uniform Resource Identifiers, URIs), which provide a syntax and conventions that when followed result in globally unique identifiers. In RDF, the distinction between an IRI/URI and a Uniform Resource Locator (URL, a subset of the more general Internationalized Resource Locator, IRL) is important.

URL and URI A URL is a string that can be entered into a browser to fetch a web page. A URI can identify any kind of resource, physical or conceptual as well as a web page (URLs are a subset of IRLs, a web page is one kind of resource). For example, a URI could be constructed to identify a specific physical air vehicle (specific tail number), and that URI could be used in RDF documents that have information about that specific physical vehicle.

[16] An OWL ontology is more semantically rigorous than the ASoT ontology provided in Appendix B.
[17] A list of OWL reasoners is provided here: https://www.w3.org/2001/sw/wiki/OWL/Implementations.

Two exceptions are that an object (third element in a triple) can also be a literal, and any of the three can be a blank node (a way of referring locally within a document to anonymous resources, resources that have no known IRI).

An RDF document specifies a graph structure because a "subject predicate object" triple is interpreted as node-edge-node. The subject and object fields uniquely identify nodes. Each triple specifies an edge between that pair of nodes. The predicate of a triple does not identify a specific edge, it is a sort of annotation on an edge that is conventionally interpreted as a type or kind of edge. An RDF document is globally uniquely identified by its namespace IRI. Note this is an IRI/URI and not necessarily a URL/IRL. It is common to see copies of a document stored in multiple locations that are fetched using different URLs. There do not seem to be any standards or conventions for change management of RDF documents. Different copies of an RDF document may identify themselves with the same IRI but have different content. This is a significant consideration for MBE.

Shape An RDF shape declares a set of constraints that can be checked against an RDF file. For example, a shape may require that a resource having a certain type/class must have certain properties (must appear as a subject in a triple with a certain predicate). A shape may require that the property value (the object of the triple with the subject and predicate) be constrained in some way. A validation tool will check a given RDF document against a shape and raise errors if any shape constraints are violated.

The distinction between shapes and ontologies is that shape constraints are "enforced" in the sense a verification tool will complain if any individual constraint is violated. Ontology definitions are accepted as axioms that can be used to infer additional information, and a reasoner will only complain if a set of axioms is inconsistent (leads to a logical contradiction). For example, if a shape declares that a particular property is constrained to have an object resource of a particular class, then any instance of a property having a different class of object causes an error. If an ontology defines a particular property as having a certain class as its range, then any triple having that property and an object resource will allow the reasoner to infer that object is also a member of the defined property range. That may add to the set of classes of which that object resource is a member (resources may be members of many classes, as long as that does not result in a contradiction in the overall set of ontology definitions and RDF document statements). A shape may take a given document and remove information from it in the sense it may find some information in that document to be invalid. An ontology may take a given document and add information to it, information that can be inferred using data in the document and definitions in the ontology. Our impression is that the OSLC is more focused on the use of shapes and constraint-checking than they are on the use of ontologies and reasoning.

Service Provider In OSLC, a Service Provider is defined as a "product or online service offering that provides an implementation of one or more services" [11]. The ServiceProviderCatalog is composed of ServiceProviders as indicated in the UML diagram of Figure 18. For DOORS, ServiceProviders are defined as the requirements projects located on the DOORS server in use. Each of theses ServiceProvider nodes in the ServiceProviderCatalog contains a link to another RDF document containing the resources of the ServiceProvider.

6.4 Examination of an OSLC Implementation

The ServiceProvider RDF document expresses the core of the OSLC functionality. This is shown in Listing 1. First the services of the OSLC Service Provider are differentiated by domain. This is done through grouping by OSLC Service nodes. Each OSLC Service node has an exactly one relationship with the property oslc:domain [5]. This means that each OSLC Service must implement services for exactly one OSLC domain. In the case of DOORS, these domains are `http://open-services.net/ns/rm#` and `http://jazz.net/ns/rm/dng/reqif#`. These domains URIs correspond to the Requirements domain for OSLC and a ReqIF domain for IBM Jazz.

Listing 1: Service Provider

```
<rdf:RDF
\dots
  <oslc:ServiceProvider rdf:about="https://adventium.clm.ibmcloud.com/rm/
     oslc_rm/_FXT0IPZ6EemT2u6MUxNx3A/services.xml">
  \dots
    <dcterms:title rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Area 51</dcterms:title>
    <jp10:supportOSLCSimpleQuery rdf:datatype="http://www.w3.org/2001/
       XMLSchema#boolean"
    >true</jp10:supportOSLCSimpleQuery>
    <oslc:service>
      <oslc:Service>
        <calm:filter>
          <calm:Filter>
            <calm:filterBase rdf:resource="https://adventium.clm.ibmcloud.com
               /rm/calmFilter/_FXT0IPZ6EemT2u6MUxNx3A/
               requirementsChangedSince"/>
            <oslc:resourceType rdf:resource="http://open-services.net/ns/rm#
               Requirement"/>
            <oslc:usage rdf:resource="http://jazz.net/xmlns/prod/jazz/calm
               /1.0/requirementsChangedSince"/>
            <dcterms:title>Requirements Changed Since Filter</dcterms:title>
          </calm:Filter>
        </calm:filter>
        <oslc:creationFactory>
          <oslc:CreationFactory>
            <oslc:usage rdf:resource="http://open-services.net/ns/core#
               default"/>
            <oslc:resourceType rdf:resource="http://jazz.net/ns/rm/dng/config
               #DeliverySession"/>
            <oslc:creation rdf:resource="https://adventium.clm.ibmcloud.com/
               rm/delivery-sessions"/>
            <dcterms:title rdf:datatype="http://www.w3.org/2001/XMLSchema#
```

```
            string"
          >Delivery Session Factory</dcterms:title>
        </oslc:CreationFactory>
      </oslc:creationFactory>
            \dots
```

The other predicates defined for a Service are

- oslc:creationDialog
- oslc:creationFactory
- oslc:queryCapability
- oslc:selectionDialog
- oslc:usage

Each of these in the specifications for an OSLC Service is defined such that these predicates may occur zero-or-many times. The following are the occurrences of these properties in DOORS (from [5]):

CreationDialog

- Collection Creation
- Requirement Creation

SelectionDialog

- Collection Selection
- Requirement Selection

QueryCapability

- Folder Query Capability
- Query Capability

CreationFactory

- Type System Copy Session Factory
- Delivery Session Factory
- Collection Creation Factory
- Requirement Creation Factory

Filter

- Requirements Changed Since Filter [18]

[18] This Service is of Jazz type and not OSLC

Each of the listed service instances addresses a certain aspect of the functionality to the tool installation which the OSLC server provides access. For example, in DOORS the Requirement QueryCapability provides query capabilities on Requirements. In DOORS for the Query Capability, this means the ability to lookup resources with OSLC query syntax. This means that with the appropriate request to a resource, the RDF documenting the resource should be returned as the result. This document contain an oslc:ResponseInfo and a rdf:Description URI. Contained in the rdf:Description are the resources which have been queried about. Based on the query, there may be various attributes provided, but at minimum there will be the URI in the rdf:about property. This URI is the location in which the various Create Read Update Delete (CRUD) operations may be performed. The update operation has some limitations to changes to the resource that may be effected. Two properties that can not be changed through a PUT are date modified and title. OSLC outlines support for paging in the case that the returned resources are too many to be handled as a single RDF document. DOORS supports this feature.

With the CreationFactory, the process for creating a resource (in DOORS a Requirement or Requirement Collection) is to submit a RDF document to a CreationFactory with the attributes required by the resource(s) defined for that CreationFactory. A sample response of a query is provided in Listing 2.

Listing 2: Query Response RDF

```
<rdf:RDF xmlns:rrmNav="http://com.ibm.rdm/navigation#"
\dots
    xmlns:rt="https://adventium.clm.ibmcloud.com/rm/types/"
>
    <oslc:ResponseInfo rdf:about="https://adventium.clm.ibmcloud.com/rm/
       views?projectURL=https%3A%2F%2Fadventium.clm.ibmcloud.com%2Frm%2
       Fprocess%2Fproject-areas%2F_7ag_0HTpEeqbb_UaJSTruA&amp;oslc.query=
       true&amp;oslc.prefix=dcterms%3D%3Chttp%3A%2F%2Fpurl.org%2Fdc%2Fterms
       %2F%3E&amp;oslc.select=*&amp;oslc.where=dcterms%3Aidentifier%3D3637">
        <dcterms:title>Query Results: 2</dcterms:title>
    </oslc:ResponseInfo>
    <rdf:Description rdf:about="https://adventium.clm.ibmcloud.com/rm/views?
       oslc.query=true&amp;projectURL=https%3A%2F%2Fadventium.clm.ibmcloud.
       com%2Frm%2Fprocess%2Fproject-areas%2F_7ag_0HTpEeqbb_UaJSTruA">
        <rdfs:member>
            <oslc_rm:Requirement rdf:about="https://adventium.clm.ibmcloud.
               com/rm/resources/CA_072480cb54504da18c049fef35233e09">
                <dcterms:creator rdf:resource="https://adventium.clm.ibmcloud
                    .com/jts/users/cdake"/>
                <jazz_rm:primaryText>(Phase 3 will be considered in the
                   development of the requirements for Phase 2, but the Phase
                    3 requirements will be documented separately.)</jazz_rm:
                   primaryText>
                            ...
```

```
            <dcterms:identifier rdf:datatype="http://www.w3.org/2001/
               XMLSchema#integer">3637</dcterms:identifier>
            <nav:parent rdf:resource="https://adventium.clm.ibmcloud.com/
               rm/folders/_7DOSoXUUEeqbb_UaJSTruA"/>
         </oslc_rm:Requirement>
      </rdfs:member>
      <rdfs:member>
            ...
      </rdfs:member>
   </rdf:Description>
<
```

Each of the CreationDialog, SelectionDialog, QueryCapability and CreationFactory comes with its own OSLC specification assigning which properties are associated with the given capability/service. The specification is done in a table (at least in version 2.0 of the OSLC specification, everything else has been 3.0) with Prefixed Name, Occurs, Read-only, Value-type, Representation, Range and Description. The properties effectively in common between the different services are dc:title, oslc:label and oslc:usage. In DOORS only dcterms:title and oslc:usage are encountered. For dcterms:title the correspondence is to a 'title string' while oslc:usage is for the 'domain specified usage' [5].

For each of the CreationDialog, SelectionDialog, Query Capability, and CreationFactory services are associated additional properties important to their usage or reference. In the case of Query Capability, it is required that there be 'exactly-one' oslc:queryBase [5]. This queryBase is the URI to which queries are submitted as to the resources that the QueryCapability is defined for. The definition of these resources may be added through the providing the oslc:resourceShape and oslc:resourceType properties. In DOORS there are two oslc:resourceType properties provided. One of these is for Requirement Collections and the other for Requirements. Similarly for the different dialog services, there is an oslc:dialog item which must occur 'exactly-one' time [5]. Also for a CreationFactory there is a oslc:creation URI to post resources to be created . This property must occur 'exactly-one' time in a CreationFactory [5].

Figure 17 shows the RDF graph associated with one of the statements in the RDF document of the ServiceProviderCatalog shown in Listing 3.

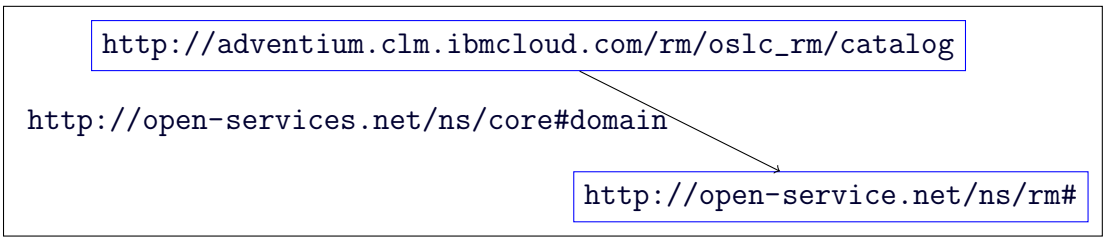

Figure 17: OSLC RDF Graph

Here, the labels for the nodes and the graphs are URLs. This is because each element is a URI reference. The subject is `http://adventium.clm.ibmcloud.com/rm/oslc_rm/`, the predicate is `http://open-service.net/ns/rm#`, and the object is `http://open-services.net/ns/core/#domain`. This says the entities represented by the subject and object URI references have the subject-object relationship specified by the predicate URI reference.

Listing 3: ServiceProviderCatalog

```
<rdf:RDF>
       <oslc:ServiceProviderCatalog rdf:about="https://adventium.clm.
          ibmcloud.com/rm/oslc\_rm/catalog">
       <oslc:domain rdf:resource="http://open-services.net/ns/rm\#" />
       <dcterms:publisher rdf:resource="https://adventium.clm.ibmcloud.com/
          rm/application-about" />
       <dcterms:title rdf:parseType="Literal">RMCatalog</dcterms:title>
       <oslc:serviceProvider>
              <oslc:ServiceProvider rdf:about="https://adventium.clm.
                 ibmcloud.com/rm/oslc\_rm/\_7ag\_0HTpEeqbb\_UaJSTruA/
                 services.xml">
                     <oslc:details rdf:resource="https://adventium.clm.
                        ibmcloud.com/rm/process/project-areas/\_7ag\
                        _0HTpEeqbb\_UaJSTruA" />
                     <jp:consumerRegistry rdf:resource="https://adventium.
                        clm.ibmcloud.com/rm/process/project-areas/\_7ag\
                        _0HTpEeqbb\_UaJSTruA/links" />
                     <dcterms:title rdf:parseType="Literal">Demo Project (
                        Requirements)</dcterms:title>
           </oslc:ServiceProvider>
       </oslc:serviceProvider>
       ...
       </oslc:ServiceProviderCatalog>
</rdf:RDF>
```

An OSLC resource is assigned a unique URI which identifies it in accordance with the principles of linked data. In OSLC, a resource corresponds to a digital artifact which is managed under the tool's OSLC implementation. The OSLC service is required to provide representation of resources as Resource Description Framework Schema (RDFS) [4]. Some parts of these representations must be done through OSLC vocabularies and others through other Semantic Web vocabularies. These resource documents are obtained through Hypertext Transport Protocol (HTTP) requests. In particular, OSLC Core 3.0 necessitates Create, Read, Update, and Delete operations (CRUD) in item 4.6.1.

"4.6.1 OSLC Services use HTTP for create, retrieve, update and delete operations on resources. OSLC Services MUST comply with the HTTP specification" [4].

6.4.1 OSLC Discovery

There are two methods stated for discovery in the OSLC specification: Static Up-Front and Dynamic Incremental. With Static Up-Front discovery, ServiceProviderCatalog, ServiceProvider and Service resources are used to provide access to the OSLC resource of the tool in question. A UML picture representation is shown below in Figure 18. This approach offers up-front access to all the OSLC resources and functionality of the server.

Dynamic Incremental offers only a select amount of information about the server at a given time. Here the discovery is "deferred," just to offer the information that is needed. In the OSLC 3.0 Discovery specification, it is stated that this is more suited to the situation where the services being provided are changing rapidly [5].

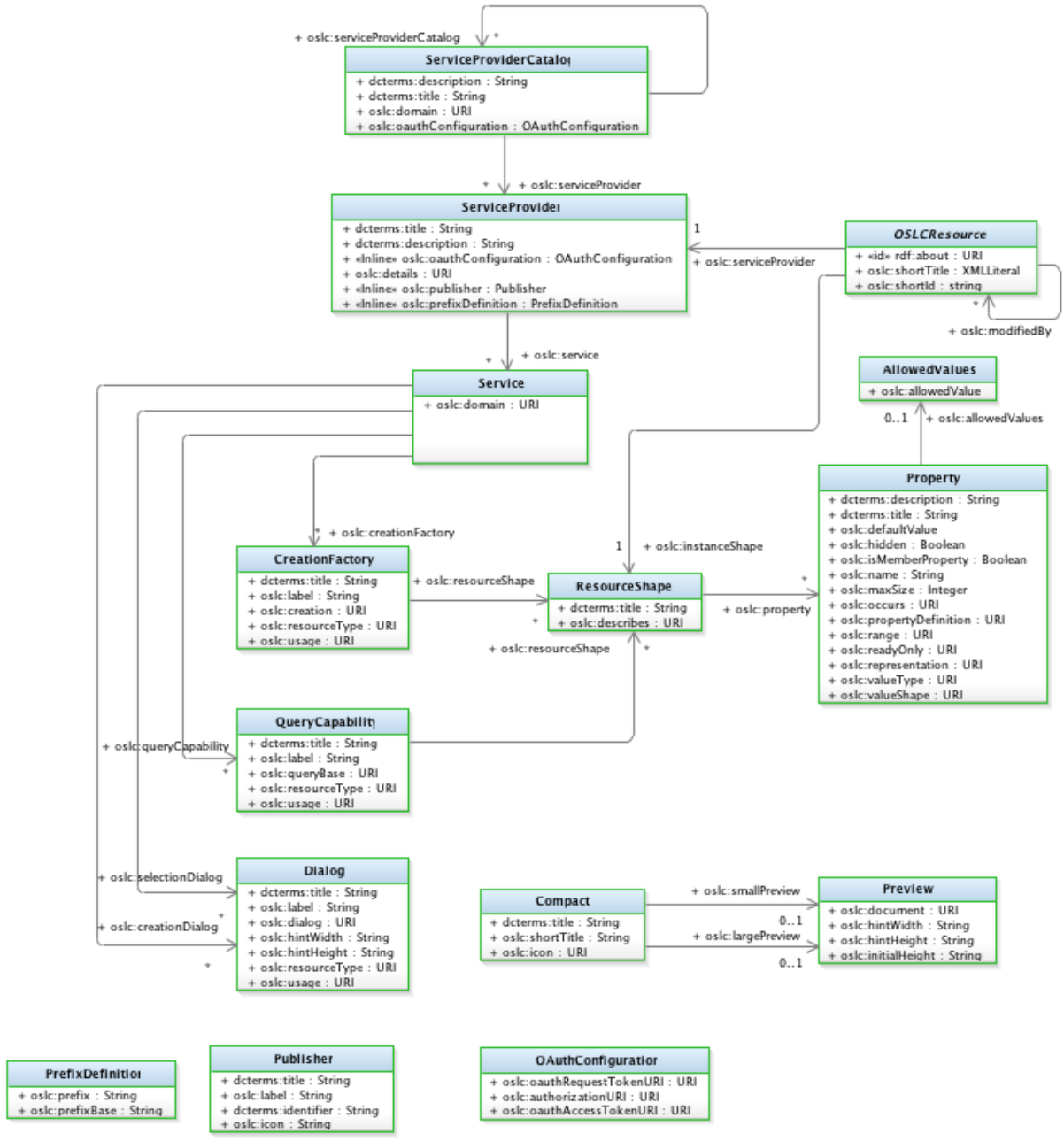

Figure 18: ServiceProviderCatalog [5]

The DOORS OSLC server uses the Static Up-Front in a setup matching that of the diagram above in Figure 18. A ServiceProviderCatalog is used to access Service Providers and through the Service Providers individual domains and services. The significance of these terms will have subsequent explanation. With regard to DOORS, the starting point is not the ServiceProviderCatalog however but a rootservices webpage located at `https://<YOUR_JAZZ_SERVER>/rm/rootservice` which provides a listing of services including the

OSLC server. This is provided through a RDF document. Access to the ServiceProviderCatalog is restricted by requiring proper authentication through OAuth. OAuth is an open standard for authentication. The resources needed to perform the multi-step OAuth 'handshake' are provided in the rootservices RDF. We were able to use Requests OAuth library in Python to obtain the OAuth token for accessing the ServiceProviderCatalog [33]. We automated the OAuth sequence using Selenium for the User Authentication step [20]. According to the OSLC standard, the definitions are MAY for using HTTP Basic Authentication, SHOULD for using SSL in case of HTTP Basic Authentication and SHOULD for using OpenIDConnect (which is "simple identity layer on top of the OAuth 2.0 protocol") [4].

A sufficiently intelligent client can take the ServiceProviderCatalog URL of an arbitrary OSLC server and determine what the OSLC server has to offer automatically. Through the listing of ServiceProviders the client can see what projects are exposed. By accessing the individual ServiceProviders, it may determine what domains the ServiceProviders implement and what OSLC functionality is available on those domains. The client here does not need to know any of the implementation specific details of the OSLC server. Navigation to the OSLC functionality follows directly from the OSLC specification of a Static Up-Front implementation.

6.5 Resources and Resource Shapes

A more challenging aspect of OSLC integrations than service discovery is the interpretation of resources which represent not OSLC services but domain-specific resource types. It is the OSLC domain-specific resource types which are used to represent the internal data of the tool or application. To understand the approach taken in OSLC, it is worth understanding linked data in general as a semantic web concept. Resources with URIs are the principal element in linked data. This is evident in Berners-Lee four principals of linked data:

1. Uses URIs to identify things.
2. Use HTTP URIs so that people can look up names.
3. When someone looks up a URI, provide useful information, using the standards (RDF, SPARQL).
4. Include links to other URI so that they can discover more things [9].

For certain OSLC elements, like those in the core specification, the resources are given direct specification according to OSLC vocabularies. For example, a ServiceProvider or a QueryCapability have definite properties associated with them. Not all of the resources of an OSLC server are defined in this way. In the case of OSLC domain-specific resources, the resources represent, agian, the data of the server. As such, they are variable in form and need a system of integrity constraints to enforce the characteristics of the data. To this end, OSLC uses resource shapes for defining resource integrity constraints.

The resource shape specification outlines the construction of shape formats which are in turn enforced on OSLC RDF resources. This enforcement of the "shape" of RDF resources

is what gives it the name. For a given resource, there may be zero-or-more resource shapes applying to it. Three ways are defined for associating shapes with a resource.

1. oslc:instanceShape – associates a resource with a constraining shape
2. oslc:resourceShape – associates constraining shape with the service
3. oslc:valueShape – "associates a constraining shape with the resource that is the object value of a property of a resource"

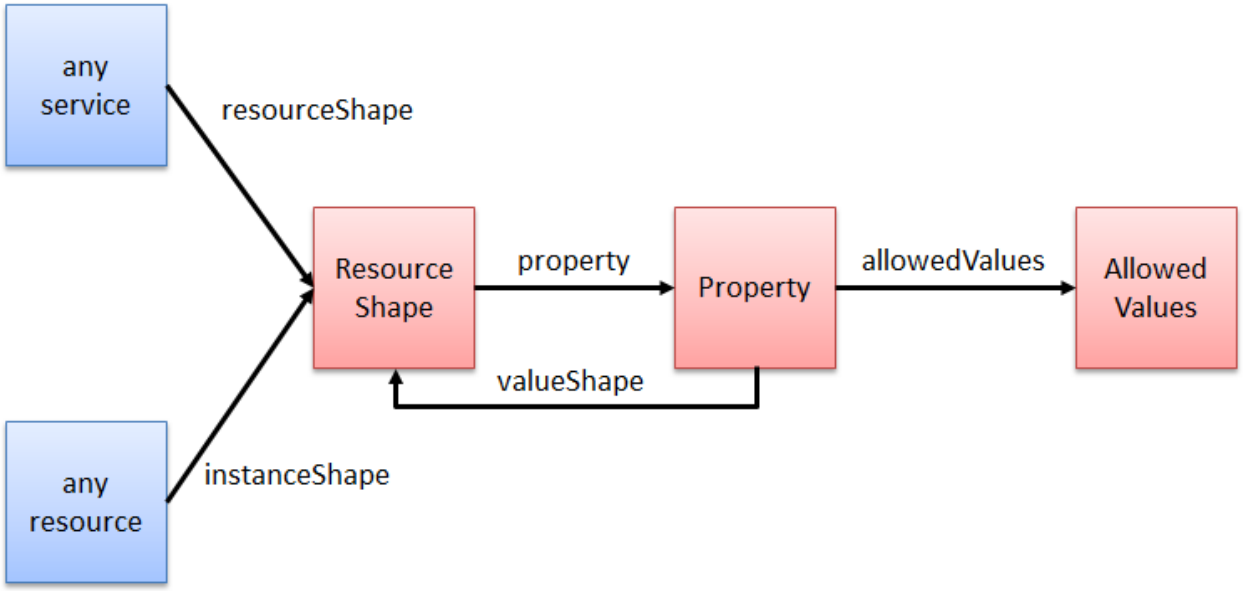

Figure 19: Resource Shape [5], [6]

The usage most relevant at this point of discussion is the second, oslc:resourceShape. The resource shape that deals with the domain and tool specific resources (non-OSLC), the internal data. For example, in DOORS, the CreationFactory has ResourceShapes associated with it specifying the resources which can be created.

The RDF properties of ResourceShapes are defined as the following (from [6]).

- dcterms:description – zero-or-one – description of Property resource
- dcterms:title – zero-or-one – summarized ResourceShape or Property
- oslc:describes – zero-or-many – lists described resources associated with shape
- oslc:hidden – zero-or-one
- oslc:property – zero-or-many – list defined properties expected in resources associated to shape

These different properties together specify the definition of the resource shape and the resources to which it may apply. The different oslc:Property nodes of the resource shape determine the definition of the resource (here nodes is meant as in referece to the RDF

structure). Each of these oslc:Property nodes constitutes a potential value in a resource which is constrained by its definition in the associated resource shape. The definition of this value is given by the RDF properties which are defined for the oslc:Property. An example here is shown in Figure 4.

Listing 4: OSLC Resource Shape Property (DOORS)

```
<oslc:Property>
    <oslc:propertyDefinition rdf:resource="http://purl.org/dc/terms/
       references"/>
    <dcterms:title rdf:parseType="Literal">References</dcterms:title>
    <oslc:valueType rdf:resource="http://open-services.net/ns/core\#Resource
       "/>
    <oslc:range rdf:resource="http://open-services.net/ns/rm\#
       RequirementCollection"/>
    <dcterms:description rdf:parseType="Literal">Captures the relationship
       between a Requirements Management artifact and another Requirements
       Management artifact in a different RM instance or external project
       area.</dcterms:description>
    <oslc:range rdf:resource="http://open-services.net/ns/core\#Resource"/>
    <oslc:name>references</oslc:name>
    <oslc:representation rdf:resource="http://open-services.net/ns/core\#
       Reference"/>
    <oslc:range rdf:resource="http://open-services.net/ns/rm\#Requirement"/>
    <oslc:occurs rdf:resource="http://open-services.net/ns/core\#Zero-or-
       many"/>
  </oslc:Property>
</oslc:property>
```

In example 4 are a few of the Property Constraints defined in OSLC. A full list is the following (from [6]):

- dcterms:description – zero-or-one
- dcterms:title - zero-or-one
- oslc:allowedValue – zero-or-many
- oslc:allowedValues – zero-or-many
- oslc:defaultValue – zero-or-one
- oslc:hidden – zero-or-one
- oslc:isMemeberProperty – zero-or-one
- oslc:maxSize – zero-or-one
- oslc:name – Exactly-one
- oslc:occurs – Exactly-one

- oslc:propertyDefinition – Exactly-one
- oslc:range – one-or-many
- oslc:readOnly – zero-or-one
- oslc:representation – zero-or-one
- oslc:valueShape – zero-or-one
- oslc:valueType – zero-or-many

Necessary predicates for OSLC properties are oslc:name, oslc:occurs, oslc:PropertyDefinition, and oslc:range. As shown in Listing 4 each of these properties are found in the example property. Breaking down the image, first we see that it is defining the constraints of a property named References. This is clear from the propertyDefinition, name and title properties. Then we see that from the Property definition that the oslc:occurs is defined as zero-or-many. This means that there can be an arbitrary number of occurences of this property in a correspondingly contrained resource. When examining the oslc:range property, it is apparent that there are two occurrences for the Property. These are `http://open-services.net/ns/rm#Requirement` and `http://open-services.net/ns/rm#RequirementCollection`. This range makes sense when considering that the snippet is taken from DOORS, a requirements tool, and that in the requirements domain there are two resource types, Requirement and RequirementCollection. The oslc:valueType is `http://open-services.net/ns/core#Reference`. This implies that it will be a URI of an OSLC server and the URI will be defined to be either a requirement or requirement collection. The oslc:description gives a more particular definition of the usage.

The property example gives illustration of how resource shapes provide definition for resources. The oslc:Property nodes in the resource shape define requirements and constraints for the definition of the requirement. A resource shape file also has an oslc:describes property. In the particular resource shape used in the example, there exists the following instances of oslc:describes (from [6]):

- `http://jazz.net/ns/rm#Module`
- `http://jazz.net/ns/rm#Composite`
- `http://open-services.net/ns/rm#RequirementCollection`
- `http://jazz.net/ns/rm#Text`
- `http://jazz.net/ns/rm#UseCaseDiagram`
- `http://jazz.net/ns/rm#Glossary`
- `http://jazz.net/ns/rm#Sketch`
- `http://jazz.net/ns/rm#Document`
- `http://jazz.net/ns/rm#SimpleFlowDiagram`
- `http://jazz.net/ns/rm#Storyboard`
- `http://jazz.net/ns/rm#WrapperResource`

- http://jazz.net/ns/rm#Collection
- http://jazz.net/ns/rm#Part
- http://jazz.net/ns/rm#BusinessProcessDiagram
- http://jazz.net/ns/rm#Term
- http://jazz.net/ns/rm#Diagram
- http://jazz.net/ns/rm#ScreenFlow
- http://open-services.net/ns/rm#Requirement

The resource shape applies to resources with the above rdf:type values. This aspect of ResourceShape definition appeals to the variation in tools. Other shape definitions may also be used in OSLC such as SHACL. In the case of DOORS, the Requirement type does not adequately represent the variation among requirements in DOORS. Resource shapes is the dcterms:title predicate is also important for Resource Shapes. In DOORS, there are several different requirement classification that may be selected when a requirement is added. These include user story, system requirement, functional requirement and so on. The dcterms:title predicate specifies which of these sub-classification the requirement belongs to.

6.6 Requirements Domain

The OSLC Requirement Management Domain is concerned with the definition of a standard for the OSLC servers of Requirements Management tools. As will be seen, the OSLC Requirement Management Domain is implemented in the DOORS OSLC server. The principal focus in the OSLC specification for Requirements Management is on the implementation of resource definition. For Requirements Management there are two types of resources, *Requirement* and *RequirementCollection*. Associated with each of these is a table listing the constraints by which the defined properties are to occur. For Requirements the properties are (from [34]):

- dcterms:title
- dcterms:description
- dcterms:identifier
- oslc:shortTitle
- dcterms:subject
- dcterms:creator
- dcterms:contributor
- dcterms:created
- dcterms:modified
- oslc:serviceProvider
- oslc:instanceShape

- oslc_rm:elaboratedBy
- oslc_rm:elaborates
- oslc_rm:specifiedBy
- oslc:rm:specifies
- oslc_rm:affectedBy
- oslc_rm:trackedBy
- oslc_rm:implementedBy
- oslc_rm:validatedBy
- oslc_rm:satisfiedBy
- oslc_rm:satisfies
- oslc_rm:decomposedBy
- oslc_rm:decomposes
- oslc_rm:constrainedBy
- oslc_rm:constrains

The properties for RequirementCollection are the same except for an additional oslc_rm:uses. The only property whose definition is necessary is dcterms:title. The rest can admit zero occurrences according to the specification.

The resource shape file whose property was used previously belonged to the CreationFactory named "Requirement Creation". This in turn belonged to the service domain which is associated with the OSLC requirements management domain, `http://open-services.net/ns/rm#`. This and the other ResourceShape files associated with the DOORS service domain offer some insight into how the OSLC Requirements Management Specification is implemented. Since it is defined to belong to the requirements domain, it must implement the vocabulary guidelines of the domain according to the domain's specification. The property definition implementation is shown in Listing 5.

Listing 5: Title Property

```
<oslc:Property>
  <oslc:valueType rdf:resource="http://www.w3.org/2001/XMLSchema\#string"/>
  <oslc:propertyDefinition rdf:resource="http://purl.org/dc/terms/title"/>
  <oslc:occurs rdf:resource="http://open-services.net/ns/core\#Exactly-one"/
     >
  <dcterms:description rdf:parseType="Literal"></dcterms:description>
  <dcterms:title rdf:parseType="Literal">Title</dcterms:title>
  <oslc:name>title</oslc:name>
</oslc:Property>
```

In Listing 5 we see that the title property has cardinality exactly-one. The definition of title in Listing 5 implies that in the DOORS implementation having a title is a necessary property not just for the resource shape file but also in the resource itself. This is because while it had been necessary to define the property in the ResourceShape, the implementation above says that the property must occur exactly-one time in the Resource. The additional property that has this occurs value is shown in Listing 6.

Listing 6: instanceShape Property

```
<oslc:Property>
  <oslc:valueType rdf:resource="http://open-services.net/ns/core\#Resource"/
     >
  <oslc:propertyDefinition rdf:resource="http://open-services.net/ns/core\#
     instanceShape"/>
  <oslc:allowedValue rdf:resource="https://adventium.clm.ibmcloud.com/rm/
     types/\_6IEzUnwKEeqbb\_UaJSTruA"/>
  <oslc:occurs rdf:resource="http://open-services.net/ns/core\#Exactly-one"/
     >
  <oslc:name>instanceShape</oslc:name>
</oslc:Property>
```

Creating Requirements Recall that oslc:instanceShape is one of the ways of associating a resourceShape with a resource. Here the direction is from the resource to the resourceShape. In DOORS it is possible to create a requirement resource with only the oslc:instanceShape and the dcterms:title supplied as properties. If a resource is created in this manner with the instanceShape from the resource shape file being used, in DOORS it will appear as a heading. The reason for this is that the dcterms:title of the resourceShape file being used is 'Heading'. A similar result will occur for the other resourceShapes associated with the DOORS creation factories.

6.7 Architecture Domain and TeamWork Cloud

In the Architecture Domain, "the resource formats are intended to define a high-level resource that can be specialized by enterprise architecture, analysis or design artifacts" [3]. There are two resources defined in the Architecture Domain: Resource and LinkType. For a AM Resource, the properties are defined as the following.

1. dcterms:contributor
2. dcterms:created
3. dcterms:creator
4. dcterms:description
5. dcterms:identifier
6. dcterms:modified

7. dcterms:source
8. dcterms:title
9. dcterms:type
10. oslc:instanceShape
11. oslc:serviceProvider
12. oslc:shortTitle
13. rdf:type

For AM LinkType, the properties are shown below.

1. dcterms:contributor
2. dcterms:created
3. dcterm:creator
4. dcterms:identifier
5. dcterms:modified
6. oslc:instanceShape
7. oslc:serviceProvider
8. rdfs:comment
9. rdfs:label

The OSLC server on TeamWork Cloud implements the Architecture Management domain. As will be seen, this is an incomplete implementation which in fact fails to meet many of the specification of the AM OSLC domain. The TeamWork Cloud OSLC server may be accessed through a rootservice page. This page is shown in Figure 7.

Listing 7: Rootservices

```
<?xml version="1.0"?>
<!--
        Copyright (c) 2017 - 2018 No Magic, Inc.
All rights reserved.
 -->

<rdf:Description xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
        xmlns:dc="http://purl.org/dc/terms/"
        xmlns:jfs="http://jazz.net/xmlns/prod/jazz/jfs/1.0/"
        xmlns:oslc="http://open-services.net/ns/core#"
        xmlns:rm="http://www.ibm.com/xmlns/rdm/rdf/"
        xmlns:oslc\_am="http://open-services.net/ns/am#"
        xmlns:oslc_rm="http://open-services.net/xmlns/rm/1.0/"
        rdf:about="https://192.168.222.120:8111/oslc/rootservices">

   <!--
                Root services resource template for applications based on JAF SDK.
```

```
                Contains required contributions both for applications and for the JTS.
                Applications may add additional services, but may only remove services noted as
                    being "JTS only".
                Specification is available at https://jazz.net/wiki/bin/view/Main/
                    RootServicesSpec
    -->

        <!-- Modify to provide a descriptive title for the application -->
        <dc:title xml:lang="en">Teamwork Cloud</dc:title>
        <dc:description>Teamwork Cloud is No Magics next generation repository for
            collaborative development and version model storage.
</dc:description>

        <!-- The following services must be included in both the JTS and applications -->

        <jfs:oauthDomain>https://192.168.222.120:8111/oslc/</jfs:oauthDomain>
        <jfs:oauthRealmName>TWC</jfs:oauthRealmName>
        <jfs:oauthAccessTokenUrl rdf:resource="https://192.168.222.120:8555/authentication/
            oauth/access_token" />
        <jfs:oauthRequestConsumerKeyUrl rdf:resource="https://192.168.222.120:8555/
            authentication/oauth/register" />
        <jfs:oauthRequestTokenUrl rdf:resource="https://192.168.222.120:8555/authentication/
            oauth/request_token" />
        <jfs:oauthUserAuthorizationUrl rdf:resource="https://192.168.222.120:8555/
            authentication/oauth/authorize" />

        <!-- End of services common to JTS and applications -->

        <!-- Applications may add any services they provide here -->
        <oslc_am:amServiceProviders rdf:resource="https://192.168.222.120:8111/oslc/am/
            catalog" />

        <oslc_am:majorVersion>19.0 SP3</oslc_am:majorVersion>
        <oslc_am:version>19.0</oslc_am:version>
        <oslc_am:buildVersion>v20191101-2128</oslc_am:buildVersion>

        <!-- End of application-specific services -->

</rdf:Description>
```

The OSLC ServiceProviderCatalog can be accessed after authenticating via OAuth 1.0 as in the case of DOORS. The ServiceProviderCatalog is shown in listing 8.

Listing 8: ServiceProviderCatalog

```
<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF
    xmlns:dcterms="http://purl.org/dc/terms/"
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
```

```
    xmlns:oslc="http://open-services.net/ns/core#"
    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#">
  <oslc:ServiceProviderCatalog rdf:about="https://192.168.222.120:8111/oslc/am/catalog">
    <dcterms:description rdf:parseType="Literal">Project hosted on this server with \ac{OSLC
        } AM services</dcterms:description>
        <dcterms:publisher rdf:resource="https://192.168.222.120:8111/oslc/twc/about" />
        <oslc:domain rdf:resource="http://open-services.net/ns/am#"/>
    <dcterms:title rdf:parseType="Literal">AMCatalog</dcterms:title>
    <dcterms:title rdf:parseType="Literal">OSLC Service Provider Catalog</dcterms:title>
    <oslc:serviceProvider>
      <oslc:ServiceProvider rdf:about="https://192.168.222.120:8111/oslc/am/12f87862-c7ea-4
          a1a-b305-2b8bf37d91f0/services">
        <dcterms:description rdf:parseType="Literal"></dcterms:description>
        <dcterms:title rdf:parseType="Literal">AADL_Profile.MASTER</dcterms:title>
        <dcterms:created rdf:datatype="http://www.w3.org/2001/XMLSchema#dateTime">
            Aug 12, 2020 1:14:24 PM</dcterms:created>
        <oslc:details rdf:resource="https://192.168.222.120:8111/oslc/am/12f87862-c7ea-4a1a
            -b305-2b8bf37d91f0/detail"/>
      </oslc:ServiceProvider>
    </oslc:serviceProvider>
    <oslc:serviceProvider>
      <oslc:ServiceProvider rdf:about="https://192.168.222.120:8111/oslc/am/11200ef3-10f1
          -45e1-a9f0-ff33ec560e10/services">
        <dcterms:description rdf:parseType="Literal"></dcterms:description>
        <dcterms:title rdf:parseType="Literal">asot-architecting-spacecraft.MASTER</
            dcterms:title>
        <dcterms:created rdf:datatype="http://www.w3.org/2001/XMLSchema#dateTime">
            Aug 12, 2020 1:11:21 PM</dcterms:created>
        <oslc:details rdf:resource="https://192.168.222.120:8111/oslc/am/11200ef3-10f1-45e1-
            a9f0-ff33ec560e10/detail"/>
      </oslc:ServiceProvider>
    </oslc:serviceProvider>
        \dots
  </oslc:ServiceProviderCatalog>
</rdf:RDF>
```

The ServiceProviders in this catalog correspond to the SysML projects on the TeamWork Cloud just as the requirement projects did on the DOORS OSLC server. An examination of these ServiceProviders shows the level of OSLC functionality supported by TeamWork CLoud.

Listing 9: Project Services

```
<rdf:RDF
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:rm="http://www.ibm.com/xmlns/rdm/rdf/"
    xmlns:oslc="http://open-services.net/ns/core#"
    xmlns:oslc_config="http://open-services.net/ns/config#"
    xmlns:dcterms="http://purl.org/dc/terms/">
```

```
  <oslc:ServiceProvider rdf:about="https://192.168.222.120:8111/oslc/am/12f87862-c7ea-4a1a
      -b305-2b8bf37d91f0/services">

    <dcterms:title rdf:datatype="http://www.w3.org/2001/XMLSchema#string">AADL_Profile
        .MASTER</dcterms:title>
    <dcterms:description rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Service
         Descriptor for Project: AADL_Profile.MASTER</dcterms:description>

        <oslc:service>
                <oslc:Service>
                        <oslc:domain rdf:resource="http://open-services.net/ns/am#"/>

                        <!-- TODO add serviceshere -->
                </oslc:Service>
        </oslc:service>

  </oslc:ServiceProvider>
</rdf:RDF>
```

According to the specificiation for the OSLC Architecture Management Domain, "AM Servers MUST provide query capabilities on oslc_am:Resource resources to enable clients to query for resources" [2]. From the above ServiceProvider document, it is apparent that it implements the domain "http://open-services.net/ns/am#." Clearly, however, there is no QueryCapability present in the ServiceProvider. In fact, there are is no OSLC functionality at all being offered. Since there are required functionalities for a valid implementation of an Architecture Management, it is clear that TeamWork Cloud does not technically implement an OSLC Architecture Management domain.

Discovery Approaching the TeamWork Cloud OSLC server from the OSLC discovery perspective, there is nothing that the server has to offer. In fact, the only information that may be obtained about the TeamWork Cloud is in the oslc:details uri for ServiceProviders in the ServiceProviderCatalog. This gives some relevant information, namely the title, description and date modified.

Listing 10: oslc:details

```
<rdf:RDF
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:oslc="http://open-services.net/ns/core#"
    xmlns:dcterms="http://purl.org/dc/terms/"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
    xmlns:acc="http://open-services.net/ns/core/acc#" >
  <rdf:Description rdf:nodeID="192.168.222.120:3579">
    <oslc:domain rdf:resource="http://open-services.net/ns/am#"/>
    <rdf:type rdf:resource="http://open-services.net/ns/core#Service"/>
  </rdf:Description>
```

```
<rdf:Description rdf:about="https://192.168.222.120:8111/oslc/am/a800b7b6-5caa-40de-
    ba24-403f4a2c2889/services">
  <oslc:publisher rdf:resource="https://192.168.222.120:8111/oslc/twc/about"/>
  <oslc:service rdf:nodeID="192.168.222.120:3579"/>
  <oslc:details rdf:resource="https://192.168.222.120:8111/oslc/am/a800b7b6-5caa-40de-
      ba24-403f4a2c2889/detail"/>
  <rdf:type rdf:resource="http://open-services.net/ns/core#ServiceProvider"/>
</rdf:Description>
<rdf:Description rdf:about="https://192.168.222.120:8111/oslc/twc/about">
  <oslc:icon rdf:resource="https://192.168.222.120:8111/oslc/twc/icons/twc.png"/>
  <dcterms:identifier>https://www.nomagic.com/products/teamwork-cloud</dcterms:
      identifier>
  <oslc:label>Teamwork Cloud</oslc:label>
  <dcterms:title rdf:datatype="http://www.w3.org/1999/02/22-rdf-syntax-ns#XMLLiteral
      ">Teamwork Cloud</dcterms:title>
  <rdf:type rdf:resource="http://open-services.net/ns/core#Publisher"/>
</rdf:Description>
<rdf:Description rdf:about="https://192.168.222.120:8111/oslc/am/a800b7b6-5caa-40de-
    ba24-403f4a2c2889/detail">
  <oslc:archived rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean">false</oslc:
      archived>
  <oslc:serviceProvider rdf:resource="https://192.168.222.120:8111/oslc/am/a800b7b6-5caa
      -40de-ba24-403f4a2c2889/services"/>
  <dcterms:description rdf:datatype="http://www.w3.org/1999/02/22-rdf-syntax-ns#
      XMLLiteral"></dcterms:description>
  <dcterms:modified rdf:datatype="http://www.w3.org/2001/XMLSchema#dateTime">Sep
      1, 2020 12:17:06 PM</dcterms:modified>
  <dcterms:title rdf:datatype="http://www.w3.org/1999/02/22-rdf-syntax-ns#XMLLiteral
      ">asot-firesat-gfi.MASTER</dcterms:title>
</rdf:Description>
</rdf:RDF>
```

The TeamWork Cloud OSLC server does have resource descriptions for model elements in the TeamWork Cloud. These resource descriptions are again minimal in content. They provide only the title, the identifier and the date modified. There does not exist an automated discovery mechanism for locating the resources. Teamwork Cloud does provide some means of getting around this by asserting that the resource may be discovered at the URI given by the pattern http(s)://TWC_IP:PORT/oslc/am/projectID/elementID [17]. The projectID is discoverable as the UUID in the ServiceProvider url. For the elementID, the author had to go into MagicDraw and extracting it by hand. This involved copying the UUID I found in the element hyperlink.

The resulting document obtained from the OSLC resource URI provided by Teamwork Cloud itself provides limited information. The semantics of the resource are not included and what is provided is the title, date modified, identifier (the UUID) and the resource type. This is shown in Listing 11. In this case the resource was always http://open-services.net/ns/am#Resource. Another resource type is also defined by the Architecture

Management domain but it was not in use here. This type, http://open-services.net/ns/am#LinkType, defines the links between architecture resource and could be useful for traceability reasons. However looking up SysML links via their UUID only yielded resources of the former type (http://open-services.net/ns/am#Resource) and with empty titles to identify them.

Listing 11: Architecture Resource

```
<rdf:RDF
    xmlns:oslc="http://open-services.net/ns/core#"
        xmlns:oslc_rm="http://open-services.net/xmlns/rm/1.0/"
        xmlns:oslc\_am="http://open-services.net/ns/am#"
        xmlns:dc="http://purl.org/dc/terms/"
    xmlns:dcterms="http://purl.org/dc/terms/"
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#">
  <rdf:Description rdf:about="https://192.168.222.120:8111/oslc/am/11200ef3-10f1-45e1-a9f0
      -ff33ec560e10/6176cbf2-7a33-473f-ba9e-139e4fd15221">
    <rdf:type rdf:resource="http://open-services.net/ns/am#Resource"/>
    <oslc:serviceProvider rdf:resource="https://192.168.222.120:8111/oslc/am/11200ef3-10f1
        -45e1-a9f0-ff33ec560e10/services"/>
    <dcterms:modified rdf:datatype="http://www.w3.org/2001/XMLSchema#dateTime">Aug
        12, 2020 2:39:16 PM</dcterms:modified>
    <dcterms:identifier rdf:datatype="http://www.w3.org/2001/XMLSchema#string">6176cbf2
        -7a33-473f-ba9e-139e4fd15221</dcterms:identifier>
    <dcterms:title rdf:parseType="Literal"></dcterms:title>
  </rdf:Description>
</rdf:RDF>
```

6.8 Conformance Measurement

When using OSLC in applications, the level of OSLC support is critical to integrations. This meaning that the extend of integration allowable will depend on what OSLC functionality is supported. For the purposes of this section, conformance will mean the extent to which the requirements and suggestions of the OSLC standard are followed within the application. The different provisions of the OSLC standard are specified in the language SHOULD, MAY and MUST. This is best illustrated through examples.

"AM Servers MAY provide creation factories for resource formats that it supports" [2]. This spec states that the presence of a CreationFactory may be a part of an OSLC implementation of the Architecture Management domain but it does not have to be. "AM Servers should support resource modifications with standard HTTP PUT and DELETE methods" [2]. In this section, the specification states that resource modifications with the standard HTTP PUT and DELETE are recommended functionalities to include. "AM Servers must provide query capabilities on oslc_am:Resource resources to enable clients to query for resources" [2]. In this example of a MUST statement, it is asserted that all implementations of a OSLC Architexture Management domain must provide query capabilities on oslc_am:Resource resources.

From the above example of OSLC specifications, it is apparent that there are varied levels of support even in valid implementations of a given domain. The difference between having a CreationFactory and not may mean the difference in the support of an intended integration scenario. In the example of a CreationFactory, this could mean being able to import resources from one tool to the another. Recall from the previous section on OSLC Discovery that there are two different discovery scenarios, *Static Up-Front* and *Dynamic Incremental*. While it is possible to take an automated approach to a Dynamic Incremental implementation, these instances are largely on a case by case basis according to what is provided by the tool vendor on their implementation. The general approach may be analyzed by considering the Static Up-Front implementation model and then applying this piecemeal for Dynamic Incremental implementations. Thus the general approach will contain the means for analyzing incremental offerings such as ServiceProviders or resources and these routines can be broken off to analyze whatever isolated part of the OSLC server is exposed.

As stated before, a Static Up-Front implementation allows a sufficiently intelligent OSLC client to determine what possible integrations are available in an OSLC server-client relationship. With such an apraisal of the capabailities in hand, the client is then in position to determine what integretion scenarios are viable.

Because of the fixed structure of a OSLC Static Up-Front Server, it is relatively straighforward to program a client to parse the OSLC server to the level of individual services being offered, even to the level of individual resources. With the server thus processed, it is possible to walk through the OSLC server to check on what level individual specification of the standard are met. The degree of completeness to which the specifications are measured will vary on how rigorously the server is tested. It is assumed here, for example, that if the URI for a CreationFactory exists and its specification in RDF is according the OSLC standard that it indeed functions as is expected of an OSLC CreationFactory. Whether or not it is implemented correctly is out of scope. The point of this OSLC client is to be able address the varying level of OSLC support present in applications and not to be a tool for testing the correctness of the functionality implemented as that is the responsibility of the tool vendors themselves.

The core of this approach is to program checks to determine whether the application follows the OSLC standard and if it does on what level it follows the standard given the variability afforded i.e. MAY, SHOULD and MUST. We illustrate this approach with the following example. Consider the specification in Table 3.

The programmatic check for this specification argument can be defined as two existence arguments, one for a requirement CreationFactory and one for a requirement collection CreationFactory. Listing 12 shows the RDF for an OSLC ServiceProvider of a DOORS project. This is the critical resource document for this conformance check.

Listing 12: Requirements Creation Factory

```
<oslc:CreationFactory>
```

Creation Factories	MUST / MAY	OSLC service providers MUST provide at least one creation factory resource for requirements and MAY provide creation factory resources for requirement collections

Table 3: Conformance cc-9 [34]

```
<oslc:resourceType rdf:resource="http://open-services.net/ns/rm\#
   Requirement"/>
<oslc:creation rdf:resource="https://adventium.clm.ibmcloud.com/
   rm/requirementFactory?projectURL=https%3A%2F%2Fadventium.clm.
   ibmcloud.com%2Frm%2Fprocess%2Fproject-areas%2F\
   _1l0HcC2GEeqsueXZ\_glatQ"/>
<oslc:usage rdf:resource="http://open-services.net/ns/core\#
   default"/>
<oslc:resourceShape rdf:resource="https://adventium.clm.ibmcloud.
   com/rm/types/\_28zIii2GEeqsueXZ\_glatQ"/>
<oslc:resourceShape rdf:resource="https://adventium.clm.ibmcloud.
   com/rm/types/\_28zIgi2GEeqsueXZ\_glatQ"/>
<oslc:resourceShape rdf:resource="https://adventium.clm.ibmcloud.
   com/rm/types/\_28yhgi2GEeqsueXZ\_glatQ"/>
<oslc:resourceShape rdf:resource="https://adventium.clm.ibmcloud.
   com/rm/types/\_28zIjC2GEeqsueXZ\_glatQ"/>
<oslc\_config:component rdf:resource="https://adventium.clm.
   ibmcloud.com/rm/cm/component/\_1smE4C2GEeqsueXZ\_glatQ"/>
<oslc:resourceShape rdf:resource="https://adventium.clm.ibmcloud.
   com/rm/types/\_28zIhy2GEeqsueXZ\_glatQ"/>
<oslc:resourceShape rdf:resource="https://adventium.clm.ibmcloud.
   com/rm/types/\_28zIjS2GEeqsueXZ\_glatQ"/>
<oslc:resourceShape rdf:resource="https://adventium.clm.ibmcloud.
   com/rm/types/\_28zIgS2GEeqsueXZ\_glatQ"/>
<oslc:resourceShape rdf:resource="https://adventium.clm.ibmcloud.
   com/rm/types/\_28zIjy2GEeqsueXZ\_glatQ"/>
<oslc:resourceShape rdf:resource="https://adventium.clm.ibmcloud.
   com/rm/types/\_28zIiy2GEeqsueXZ\_glatQ"/>
<oslc:resourceShape rdf:resource="https://adventium.clm.ibmcloud.
   com/rm/types/\_28zIgy2GEeqsueXZ\_glatQ"/>
```

```
            <dcterms:title rdf:datatype="http://www.w3.org/2001/XMLSchema\#
               string"
            >Requirement Creation Factory</dcterms:title>
            <oslc:resourceShape rdf:resource="https://adventium.clm.ibmcloud.
               com/rm/types/\_28yhgy2GEeqsueXZ\_glatQ"/>
            <oslc:resourceShape rdf:resource="https://adventium.clm.ibmcloud.
               com/rm/types/\_28zIhi2GEeqsueXZ\_glatQ"/>
            <oslc:resourceShape rdf:resource="https://adventium.clm.ibmcloud.
               com/rm/types/\_28zIiS2GEeqsueXZ\_glatQ"/>
            <oslc:resourceShape rdf:resource="https://adventium.clm.ibmcloud.
               com/rm/types/\_28zIhC2GEeqsueXZ\_glatQ"/>
            <oslc:resourceShape rdf:resource="https://adventium.clm.ibmcloud.
               com/rm/types/\_28zIgC2GEeqsueXZ\_glatQ"/>
            <oslc:resourceShape rdf:resource="https://adventium.clm.ibmcloud.
               com/rm/types/\_28yhhC2GEeqsueXZ\_glatQ"/>
            <oslc:resourceShape rdf:resource="https://adventium.clm.ibmcloud.
               com/rm/types/\_28zIji2GEeqsueXZ\_glatQ"/>
            <oslc:resourceShape rdf:resource="https://adventium.clm.ibmcloud.
               com/rm/types/\_28zIhS2GEeqsueXZ\_glatQ"/>
          </oslc:CreationFactory>
```

Note that oslc:resourceType is defined with the value `http://open-services.net/ns/rm#Requirement`. This identifies that the CreationFactory is in fact defined for requirements. From this it is implied that the MUST part of the compliance requirement is met.

Listing 13: Requirement Collection Creation Factory

```
<oslc:CreationFactory>
  <oslc:usage rdf:resource="http://open-services.net/ns/core\#default"/>
  <oslc:resourceShape rdf:resource="https://adventium.clm.ibmcloud.com/rm/
     types/\_28zIiC2GEeqsueXZ\_glatQ"/>
  <oslc\_config:component rdf:resource="https://adventium.clm.ibmcloud.com/
     rm/cm/component/\_1smE4C2GEeqsueXZ\_glatQ"/>
  <oslc:resourceType rdf:resource="http://open-services.net/ns/rm\#
     RequirementCollection"/>
  <oslc:creation rdf:resource="https://adventium.clm.ibmcloud.com/rm/
     requirementFactory?projectURL=https%3A%2F%2Fadventium.clm.ibmcloud.com
     %2Frm%2Fprocess%2Fproject-areas%2F\_1l0HcC2GEeqsueXZ\_glatQ"/>
  <dcterms:title rdf:datatype="http://www.w3.org/2001/XMLSchema\#string"
  >Collection Creation Factory</dcterms:title>
</oslc:CreationFactory>
```

From Listing 13 the MAY requirement is met since the type for CreationFactory is `http://open-services.net/ns/rm#RequirementCollection`. The OSLC standard is composed

of many specification documents each composed of many individual requirements. For this reason, an attempt to outline a programmatic approach for each requirement has not been made at this point. Not of all the requirements of the OSLC standard may be simply forumlated as existence arguments pertaining to the presence of specifically encoded URIs. An existence argument may be sufficient in the case of Table 4 but in the case of Table 5 more sophisticated checks would be requried.

Query Capabilities	MUST	OSLC service providers MUST provide query capabilities to enable clients to query for resources.

Table 4: Conformance cc-10 [34]

Query Syntax	MUST	OSLC query capabilities MUST support the OSLC Core Query Syntax.
Resource Paging	MAY	OSLC service MAY provide paging for resources but only when specifically requested by service consumer

Table 5: Conformance cc-11,cc-5 [34]

The problem with the latter table is that checking whether query capabilities support the OSLC Core Query Syntax is itself non-trivial. An approach might be to see if valid OSLC queryies are accepted and invalid ones are not. However, such an approach will fail to be complete.

The query syntax requirement is an outlier among the requirements of the OSLC standard. The majority of the requirements have to do with RDF formatting, return types and existence or existence contstraint arguments. Checking whether a query capability uses

the OSLC query syntax is ultimately a responsibility of the tool vendor and not that of a client attempting an integration. The existence arguments which are of the most interest to integrations since, assuming they are implemented correctly, they determine the possible integration scenarios. Returning to the CreationFactory example, a pseudocode approach for the check is shown in Algorithm 1.

Algorithm 1 Compliance Check

```
1: procedure CheckCompliance-cc-9
2:     resources ← parse_catalog()
3:     creationFactories ← resources.get_creationfactories()
4:     resourceTypes ← creation_factories.get_resourceTypes()
5:     return resourceTypes.contains(Requirement)
```

The approach proposed is to break the specification into those items which are discoverable by the client and those which are the responsibility of the vendor. From the previous example, it is clear that the presence of query capability as specified by its domain is something which is discoverable by the client. With regard to whether the server's query capability correctly implements the OSLC query syntax, that is a matter of implementation which is invisible to the client. The client will attempt to integrate on the assumption that such implementation details have been performed correctly. The client will therefore use the fixed nature of an OSLC server in order to determine what domains are implemented and within those domains what services are offered. From this analysis, the client may couple what integrations may exist from tool to tool. Also it may determine, by virtue of the specifications, whether the OSLC server meets the requirements of the domains it supports.

6.9 Topics for Further Research

1. To what degree does OSLC conform to/make use of RDF, RDFS, Shapes Constraint Language (SHACL), OWL, and SPARQL standards, in theory and in practice?
2. How well do these standards support tool interoperability? How much latitude do tool developers have to encode information in tool-specific ways? How much collaboration among vendors and/or additional standardization by the OSLC community is needed?
3. Could a standard SPARQL portal be provided in conjunction with an OSLC portal, i.e., could SPARQL be used with OSLC in practice?

6.10 Conclusion

OSLC offers a lot of possibilities provided a featureful implementation of an OSLC Server. This can be seen readily in DOORS. With the simple CRUD operations, many applications of a diverse scope are available. Through these operations, it is apparent how a connectivity to DOORS similar to that of Syndeia can be created. The operations for creating a requirement exist in DOORS by means of the CreationFactory capability. OSLC is less valuable when less-extensively implemented, as we observed with Teamwork Cloud. We showed a potential approach to programmatically evaluate the *degree* of OSLC implementation by a given tool. Further topics for research include application of reasoners to OSLC APIs and prototype implementation of the notional OSLC evaluation tool.

7 Traceability and Provenance

Given an Authority to Operate (ATO), the Program Manager can deploy the system described by the ASoT. To obtain the ATO, the Program Manager must first demonstrate convincingly that the system satisfies its requirements and that the system-as-approved is the system-as-built. The second condition includes an argument about the *provenance* of the system and its components. The Program Manager must be able to answer for each component, who created the component, what tools they used to create the component, who has accessed the component and what changes they made, what analyses support the approval to integrate that component into the system, and what tools performed those analyses. The ASoT, through the system relationships that it maintains, plays an important role in demonstrating provenance to support the ATO.

We will typically demonstrate provenance using traceability relationships maintained in the ASoT. In this section, we consider how the tools we used in our demonstrations (see 13) can help the ASoT answer provenance queries about digital artifacts. For each type of digital artifact, we pose provenance queries, then for each query, we describe how we can use these tools to answer the query. For this discussion, we assume only four digital artifact types (requirements, SysML models, AADL models, and AADL analyses). Including other artifact types should be straightforward. See Section Section 3 for a detailed description of each tool and its role.

We arrange this section as follows.

- In Section Section 7.1, we provide more details about the types of information that the ASoT can collect using the tools named above.
- In Section Section 7.2, we consider how the ASoT could handle provenance queries against system requirements.
- In Section Section 7.3, we consider how the ASoT could handle provenance queries against SysML models of the system and its components.
- In Section Section 7.4, we consider how the ASoT could handle provenance queries against AADL models of the system and its components.
- In Section Section 7.5, we consider how the ASoT could handle provenance queries against ACVIP analyses of the AADL models.
- In Section Section 7.6, we summarize our findings for using the ASoT to demonstrate provenance.

7.1 What We Collect

We summarize here what an ASoT can collect from each tool:

- Using DOORS, we can export, using either OSLC or the custom DOORS report generator, detailed data on the relationships between the elements that DOORS manages (e.g., user stories, use cases, and requirements).
- Using MagicDraw or Cameo Teamwork Cloud, we can export detailed relationships

between elements that this tool manages (e.g., model elements represented as boxes, arrows, properties, etc.). Some of these model elements correspond to requirements in DOORS. After export, we can represent those relationships graphically using Neo4j.

- Syndeia can capture detailed relationships between select elements in MagicDraw or Cameo Teamwork Cloud and elements in DOORS. Neo4j can illustrate those relationships graphically.

We divide the information that the ASoT can collect into four categories: the OSLC data from DOORS, the OSLC data from MagicDraw, the Neo4j graph, and element relationships exported from the Syndeia Cloud REST server.

Using OSLC in DOORS, we can extract all the links between requirements, such as *validates*, *child of*, *satisfies*, etc. We can also track when the requirement was last updated. The OSLC data from Teamwork Cloud can also be used for tracking updates, but not links.

The Neo4j graph captures all internal links in a given MagicDraw/Teamwork Cloud project, and can also capture any metadata exposed by MagicDraw's Java API. We have demonstrated capture of the usual traceability information, such as *satisfies*, *usage*, and so on.

There currently are no requirements linkages in Syndeia Cloud, but it can be used to show linkages between synchronized elements of DOORS and MagicDraw. We plan to integrate this capability into the Neo4j graph along with the DOORS data exposed by OSLC.

We collect OSLC data from DOORS and Cameo Teamwork Cloud using custom OSLC client programs written in Python; Syndeia data, through its REST API using a separate custom Python script; and the Neo4J graph using a MagicDraw plugin utilizing its Java-language Open API.

We collect the DOORS data from an OSLC server hosted by IBM cloud. Teamwork Cloud hosts a similar server for its OSLC data. Likewise, Syndeia Cloud has a REST server where its data is collected. We run the MagicDraw Plugin in a user instance of MagicDraw and trigger it from within an open project. We can host these clients on a continuous integration server where they run as cron jobs.

7.2 Requirements Provenance

The Program Manager needs to demonstrate that system developers built only against authorized requirements, such as requirements provided as Government Furnished Information (GFI) or directly derived from GFI. The ASoT captures requirements traceability evidence from many sources to support this need.

- *What use cases derive from the given user story? What requirements derive from the given use case? What use cases motivated the given requirement? What user stories motivated the given use case?*

  OSLC can distinguish, based on the Resource Description Framework (RDF) properties, what type of requirement a resource is. This distinguishes user stories from use cases and from user requirements, for example. We can create a query that matches all user stories, through use cases, to the given requirement.

- *Who created the requirement/use case/user story?*

  DOORS/OSLC creates an RDF property for every resource automatically that captures the user logged into DOORS or the credentials used to access the OSLC server when the requirement is created.[19]

- *When was the requirement/use case/user story created?*

  This is also an RDF property.

- *Was the requirement/use case/user story modified? If so, when and by whom?*

  DOORS/OSLC typically lists only the most recent modification, but the contributor property should be tied to the modified property.

- *From what standard or regulation does this requirement derive?*

  This is a property of a resource and we can match related requirements using this property.

7.3 SysML Model Element Provenance

The Program Manager needs to demonstrate that every part of the design traces back to an authorized requirement. Here we assume an initial design using SysML to lay out major subsystems and their interactions. The initial design allocates requirements to SysML elements modelled using MagicDraw or Cameo Teamwork Cloud. The initial design may call out data types required by GFI. As the design proceeds, system developers ensure that each new SysML element continues to support the authorized requirements. The ASoT captures this traceability evidence as well as who created each element, who accessed and modified the element, and what legacy components, if any, the element represents.

- *What is the provenance of the allocated requirement?*

  See the questions above in Section 7.2.

  MagicDraw/Cameo Teamwork Cloud tracks traceability relations to other model elements (including requirements), but it does not currently capture user information.

- *What other elements map to this requirement? What other requirements map to this element? What other model elements use this model element? ("use" = "contain," "bind to") What other model elements does this model element use? ("use" = "contain," "bind to")*

  If the modeler documents these relationships in the SysML, then we can display them in the Neo4j graph.

[19] We did not consider Aras for the provenance discussion, but Aras provides an API that allows custom code to be triggered when an object is changed, created or deleted. This code could include recording events about the user.

7.4 AADL Model Element Provenance

The Program Manager will direct analyses to perform against the system to satisfy program goals. Some analyses will be against the SysML model, but other analyses will require translating the model to AADL. For example, ACVIP analyses will detect defects in the system architecture as modelled before any significant investment in system implementation. The ASoT will capture the links between the SysML models and the AADL models translated from them.

Currently the only method to query the provenance of AADL model elements is to extract from the AADL model the associated SysML UUID that appears as a comment for that AADL model element. (A similar comment applies to FACE to AADL model translation.)

- *What is the provenance of the corresponding SysML model element? What other SysML model elements map to this AADL model element?*

  Match the SysML element with the corresponding UUID.

- *What other AADL model elements map to this SysML model element?*

  Examine the AADL instance model for other elements with the same UUID.

- *What other AADL model elements use this AADL model element? ("use" = "contain," "bind to") What other AADL model elements does this AADL model element use? ("use" = "contain," "bind to") What AADL model elements does this AADL implementation include?*

  Examine the AADL instance model, which captures these relationships between model elements.

- *What AADL implementations include this AADL model element?*

  Search across AADL model implementations for model elements with the same path.

- *Who created this AADL model element? Was it modified? If so, when and by whom?*

  AADL, and OSATE, do not track user information. For AADL models created from scratch under version control, the version control system will reveal this information. Assuming a SysML model generates this AADL model, we can find the matching SysML model element with the same UUID to see if MagicDraw or Cameo Teamwork Cloud captures this information. In the latter case, the AADL model should include a generated date as a comment.

- What are the attributes (e.g., data rights) for this AADL element?

  Modelers typically include this information in the model header.

7.5 ACVIP Analysis Provenance

The Program Manager needs to demonstrate provenance not only for the model elements but also for the analysis results that those model elements yield. The ASoT will capture the analysis methods and analysis results, and it will link results to the AADL models that produced them.

- *What is the provenance of this AADL implementation?*

  See the questions above.

- *To what AADL implementation does this ACVIP analysis result correspond? What other ACVIP analysis results correspond to this AADL implementation? What ACVIP analysis tool and version created these ACVIP analysis results? What is the most recent ACVIP implementation against this model implementation?*

  OSATE reports the analysis using the analysis tool name and the model instance name, though it overwrites earlier analysis results for the same instance unless the user has saved those results.

- *When did we create these ACVIP analysis results? Have the AADL implementation(s) changed since producing these ACVIP analysis results?*

  The filesystem tracks creation and modification dates for these results.

7.6 Conclusions

Most tools already track the traceability information necessary for the ASoT to answer provenance queries, but they vary in the degree to which they do so automatically. For example, SysML tools do not capture user information automatically, but the developer can include that information manaully in the models. Other tools track the required information automatically but do not make that information available to other tools. For example, OSATE tracks relationships between AADL model elements in its instance model, but OSATE does not provide external way to examine that model. Universal support for an open standard like OSLC would simplify provenance demonstrations for the ASoT. Fortunately, these challenges are largely technical and easily overcome.

8 Lessons from Capstone Performers

The recently completed Joint Multi-Role - Technology Demonstrator (JMR-TD) MSAD Capstone demonstration provided numerous examples and lessons learned for several ASoT capabilities:

- *discoverability* is the ease with which various stakeholders can find the information they need to perform specific tasks.
- *version and configuration manageability* is the ease with which multiple variations of a component and collections of components can be managed so that they are consistent and work with each other for specific projects and systems.
- *interoperability between organizations* is the ease with which organizations that have different governance structures, processes, assets, tools, and business interests can exchange information to collaboratively execute a specific project.
- *interoperability between modeling environments* is the ease with which information captured in different modeling languages and data formats, operated on by different tools, and stored in different repositories, can be related to or exchanged with other infor-

mation.

8.1 Capstone Demonstration Background

Figure 20 illustrates the model-based acquisition context exercised by the Capstone Demonstration. This is a forward-looking rather than existing context, as MSAD investigated and matured technologies to support the transition to model-based agile acquisition of famlies of systems.

The left column shows a hierarchy of scopes of activities and purposes. Enterprise is the broadest scope considered. An example in this context would be all activities and concerns relating to Army aviation weapon systems. The second column lists examples of organizations that could have decision authority over a scope of activities and concerns, including managing assets common across that scope. The third column lists information assets used in the Capstone demonstration. For example, PEO Aviation could manage the information assets used to procure and sustain all weapon systems used by the Army aviation enterprise.

For the Capstone demonstration, the Reference Architecture contained enterprise-level assets developed as part of MSAD and provided as GFI to Capstone performers. Some examples of Reference Architecture assets (in addition to documents describing this overall CAS) are regulatory and certification requirements; enterprise-level business drivers and architecture quality attributes that flow down as needed to specific acquisitions; a common taxonomy for mission system capabilities; a model-based Functional Reference Architecture (FRA) used to specify required mission system functionality with greater rigor and less ambiguity; a Domain Specific Data Model (DSDM) used to specify data processed by a mission system; and standards, templates, and Government Furnished Equipment (GFE) that flow down as needed to specific acquisitions.

Concepts and assets in the MSAD Reference Architecture have been incorporated into the FVL Architecture Framework (FAF), and we anticipate from there will go into the Army Aviation Architecture Framework (AAAF).

In the Capstone demonstration, two notional GFI system specifications were used: one for assault missions and one for attack missions. The two Objective Architectures (one done by an Architect contractor, the other by two collaborating MSIs) applied to a Family of Systems. These models included variation points to distinguish common from system-specific aspects of the two kinds of mission systems. Each of the three MSIs then developed a System Architecture model for one of the two kinds of mission systems using their assigned Objective Architecture. A System Architecture specifies the architecture of an acquired mission system in a way that minimizes unnecessary constraints on design and implementation. Detailed architectural design information is provided in a System Design Technical Data Package (TDP) that should be sufficiently detailed, and have sufficient data rights, to (for example) enable a qualified third party to integrate new components into a mission system.

The Reference Architecture assets primarily focused on regulatory and business requirements such as certifications, Modular Open Systems Approach (MOSA) objectives, and permited and required standards and guidelines. The capability and performance requirements for a specific mission system were captured in a specification document and Department of

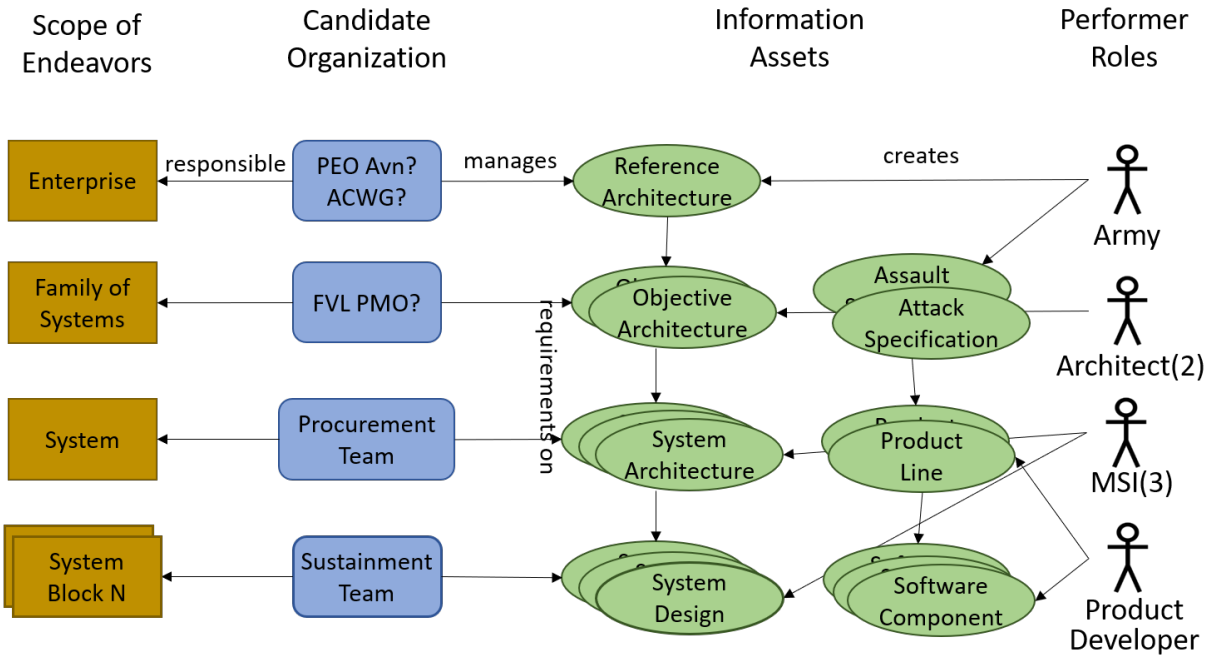

Figure 20: *Capstone Demonstration Governance, Roles, Information Assets, Performers.*

Defense Architecture Framework (DoDAF) capabilities model. The acquired demonstration systems were required to satisfy both architecture and capability/performance specifications. The System Designs had to meet the full set of requirements derived from all sources. The degree to which selected performance and capability requirements could or should appear in the System or Objective architectures was a subject of investigation.

Each information asset shown in Figure 20 contained a mix of formats. In addition to natural language documents in Word® and PDF formats, models written in several modeling languages were produced and consumed by multiple Capstone performers. System Modeling Language (SysML), FACE, and AADL were specifically selected for investigation and maturation.

Figure 20 shows seven Capstone performers. The Architect and Mission System Integrators (MSIs) were either industry teams or had subcontractors so that internal collaboration had to occur as well. The Army role stood up a FedRAMP-approved Sharepoint® site to exchange information between the Government and other performers. Each performer stood up their own development infrastructure to support information exchange with their industry partners.

Capstone execution began with development of the two Objective Architectures and proceeded down through the layers of architectures and specifications. These were used for prototype design, implementation, software and systems integration, and demonstration in MSI System Integration Lab (SIL)s (a distributed SIL in the case of one MSI team). The Software Product Developer role provided four software components built according to GFI Reference Architecture specifications that were then furnished as GFI to be used by the MSIs. Capstone concluded with novel Government-directed upgrades to the three prototype

mission systems to assess open system objectives.

Of the nineteen Government demonstration objectives for Capstone, two of them specifically cited SSoT:

- Investigation/Definition/Demonstration of a Single Source of Truth (SSoT)
- Assessment of maturity and issues associated with implementing a SSoT

At the time this document was prepared, final reports were available from Capstone contractors, and quotations from those are used in our observations. Adventium Labs and Government final technical reports were not yet completed.

"The single most useful aspect of MBSE was the concept of Single Source of Truth (SSoT). Having truth defined in one place in an unambiguous manor was a big challenge but was critical to enabling the team to work together and execute. In every area where we did not have SSoT due to tooling or process short comings, there was wasted time and miscommunication causing misses and rework." (Boeing/Sikorsky MSI 2 final technical report)

"The concept of a Capstone Single Source of Truth (SSOT) for the MS is to generate a single source of MS Architecture, Requirement, Design, and Analysis information. This information did not need to be contained within a single tool for the entirety of the MS. No piece of Mission System information would have more than one source, and all sources must be clearly linked to the model and searchable for SSOT." " While attempting to create a SSOT for MS development, the JMR Capstone team encountered challenges that impeded a clear establishment of SSOT for the program." (Collins MSI 1 final technical report)

8.2 Discoverability

8.2.1 Observations

"To improve readability, efficiency of peer reviews, and overall time spent in the model, the model should be organized in a format that better aligns with what a systems or software engineer would see on a traditional program." "There is a challenge in trying to navigate through the model from the functional requirements up to the SysArch requirements, as multiple diagrams must be referenced, and they do not all link together seamlessly." "As the size of the models grow, they become more and more challenging to navigate through and digest what is being modeled. Though relatively simple in JMR Capstone, this issue will be exacerbated on full-scale programs." (Collins)

"In practice, most software developers will not be MBSE experts and should not be expected to trudge through models for information that should be presented in a more developer-centric and familiar fashion." (Raytheon/GE)

"The RSC deliveries were a mix of SPARX Enterprise Architect (EA) models, Architecture Analysis and Design Language (AADL) models, a Software and Model Users Guide (SMUG), and source code. While good information was contained in all, the overall feel was somewhat disjointed." (Raytheon/GE)

"While we are delighted that Collins Aerospace, one of the MSIs in the Capstone demonstration, made good use of our AADL models in understanding the 'requirements' of their

replacement component as well as the PD supplied components, Honeywell would like to bring to the Army's attention that this was not at all what was expected." (Honeywell)

"The FACE queries and templates in the DSDM were challenging to visualize and utilize." (Raytheon/GE)

"Data model interpretation was challenging without accompanying documentation explaining rationale and assumptions of each data element." (Raytheon/GE)

"FACE data models, as is, are not consumable by humans and require tools to enable search, viewing, and contrasting data entities." "While data elements and relationships are captured in digital format within the GFI data model and can be searched, additional tools are desired to efficiently explore the definitions of elements within a message." (Collins)

"SysML profiles often just add yet another layer of confusing terminology to a model." (hallway remark made by an experienced SysML modeler)

8.2.2 Discussion

All of the Capstone performers reported challenges in data discoverability. Some of the difficulty was due to lack of clear communication, understanding, and precise semantics about how to interpret and use unfamiliar models that they did not develop.

Different layers of the CAS architecture hierarchy were done by different organizations using different interpretations of the GFI and different modeling styles. The SysML language defines six different kinds of relationships that can be used for requirements management. At the more detailed levels, additional profiles including performer-developed profiles were used. For example, requirements imported from DOORS use a DOORS profile. Following requirements traceability through these multiple layers was difficult.

A similar discoverability challenge arose with the materials delivered to the MSIs by Honeywell. Honeywell developed four reusable software components. Each MSI integrated three of these and duplicated a forth using the Honeywell TDP as a specification. Different Honeywell component development teams did different models and TDPs. This resulted in different modeling styles among the four, even though they came from the same supplier. The MSIs said common styles and structures for the TDP assets for the four components would have made integration easier.

Honeywell delivered a combination of software, FACE data models, SysML models, and AADL models. The MSIs reported difficulty determining which artifacts to search for which purposes. The MSIs were often presented with so much information in software component TDPs that they were not sure where to start and what information to focus on. For example, the performers received a set of GFI including resources like the ACVIP Capstone template model. However, the performers did not have clear guidance on how to review the provided information, which resources were highest priority, etc. The result was that few performers were able to make use of the Capstone template.

For SysML there was ambiguity due to lack of rigorously-defined domain-specific semantics. Profiles are the normal way to add domain-specific semantics, but many profiles were poorly documented, and application in practice raised detailed questions. Some contractors used

custom profiles they developed themselves.

Discoverability and traceability problems were challenging for performers using the FACE data models. Several performers described challenges understanding and exporting the GFI DSDM. Skayl described problems managing FACE identifiers when importing and updating FACE data models in enterprise architect.

Tracing relationships between elements found in different SysML, FACE, and AADL models was difficult. The FACE-to-AADL synchronization tool uses a FACE::UUID property to convey traceability, while the SysML-to-AADL synchronization tool conveys this as a comment. Information in the reusable component TDPs was not always found where expected. It was difficult to determine which model (SysML, FACE, or AADL) or document contained what information.

8.2.3 Recommendations

It should be possible to define viewpoints that identify specific kinds of information useful for specific stakeholders and tasks. Ideally a viewpoint definition could be used to automatically generate specific views (specific diagrams) rather than having to manually create and maintain individual views. We prototyped such a viewpoint in demonstration two (see Section 3.5).

Modeling guidelines should identify types of links used to trace between information. These link types should have rigorously defined semantics. Some links can be captured using modeling language elements. Some will require meta-data because they cannot be captured using the modeling language or because they identify relations between elements in different models and other information assets. We explored options for such links in demonstrations two and three (see Section 3.5 and Section 3.6).

Include navigation viewpoints. A model structure view (a.k.a. containment view, part/assembly view) is an example of a navigation viewpoint that is available in many modeling tools. Different processes or stakeholder interests can benefit from other ways of navigating information. A common recommended practice is to provide additional navigation views (diagrams) where they support common user workflows. We used Neo4j for such a navigation viewpoint in demonstration three (see Section 3.6).

Allocate time and resources for stakeholders to become familiar with the links, viewpoints, and modeling conventions and guidelines for the ASoT contents (not just for how the ASoT itself works). Documentation and organized training are needed for things like modeling languages and guidelines, but hands-on experience and ongoing collaboration are also necessary.

8.3 Version and Configuration Manageability

8.3.1 Observations

The Government established a Change Control Board (CCB) for the DSDM. Whenever a Capstone performer wanted a change in the DSDM, they went through the CCB.

"Honeywell suggests that the Army's DSDM CCB be made more efficient." (Honeywell)

"For a final product, a more stringent configuration control can be found in the guidance in the document JCAS Governance Plan v1.0.docx. The recommended entry point for this rigor would be post-initial development of the System Architecture as the overhead associated with the CCB and other configuration control measures, checks, and balances could be at the wrong level for initial development." (Raytheon/GE)

"[Our team] used several different concepts for Configuration Management of the MS TDP model." "JIRA was utilized to maintain configuration control over model elements by documenting changes made to the model under stories and work packages." " Within EA, Model Baselines were taken at program milestones." "Diagram Notes were utilized on an individual diagram level within Enterprise Architect to track future changes that needed to be made to a diagram to bring it up to current modelling practices." "Using the Phase Property within EA as a method of configuration management did not begin until excursion development." "Configuration Management for the Capstone AADL model utilized SVN to track version changes to model components." "Documentation was generated both in Microsoft Word and within the Confluence document editor." (Collins)

"Model configuration control was performed using various git tools, including Bitbucket, Tortoise Git, and Source Tree, and two separate git repositories (one GE and one Raytheon). This configuration was managed using branches from the two "master" branches, or trunks, and various baselines for coordination amongst GE and Raytheon." (Raytheon/GE)

"Maintaining accurate models for a developed component takes significant effort." (Honeywell)

"Documents were not easy to go through and multiple versions of the documents being released led to confusion between the product developer and mission system integration teams." (Honeywell)

"Aside from the lack of an SSoT, the next biggest challenge encountered was model management." (Boeing/Sikorsky)

8.3.2 Discussion

CCB will be important ASoT stakeholders. To enable more agile acquisition, more agile CCBs are needed.

Configuration management must span multiple kinds of assets that are provided and versioned by multiple organizations.

A variety of version control and configuration management approaches, methods, and tools exist. How versions and configurations are specified as well as how they are managed will be different for different kinds of assets as well as for different organizations. There will need to be some non-trivial mapping between different approaches in order to provide a unified (authoritative) way to identify and manage versions and configurations.

To support more agile and collaborative development, more sophisticated features are necessary than are found in office management products. Features such as diff/merge, maintaining branching and merging histories of component versions and configurations of components,

history tracking, and moving related changes between different repositories, are needed. These are features found in PLM and Application Lifecycle Management (ALM) development environments, where a repository is only one among a set of tools and capabilities.

8.3.3 Recommendations

State-of-the-art version and configuration management processes and tools should be used, such as those found in ALM and PLM environments. Basic file share and office management tools are not sufficient.

Identify a common way to specify versions and configurations for the ASoT. For all the sources of information and tools, define a mapping between their versioning and configuration schemes and the common ASoT scheme. This could be as simple as using the source version and configuration identifiers tagged with a unique identifier for that source, e.g., the version number tagged with a unique identifier of the vendor that assigned that version number.

Identifying rules to determine when a collection of versioned assets forms a consistent configuration is more complicated, but that also needs to be addressed.

Identify the set of CCBs that will be using the ASoT. To maintain an authoritative or single source of truth, each specific asset to be versioned should have its changes controlled by a specific CCB. The same versioned asset may appear in multiple configurations, but a configuration itself may be versioned and thus controlled by a specific CCB.

8.4 Interoperability Between Organizations

8.4.1 Observations

"Aside from the lack of an SSoT, the next biggest challenge encountered was model management." (Boeing/Sikorsky)

As part of Capstone investigation and maturation, the Architect performer developed scripts to automate the verification of conformance between the Reference Architecture, Objective Architecture, and System Architecture layers, each of which were done by different organizations.

"Model Analysis scripts were run using the Reference, Objective and eventually the System Architecture models and their relationships. The analysis focused on the CAS elaboration of requirements and the identification of specified mechanisms." "However, the MSI's modeling did not support the inclusion of all the mapped Mechanisms, meaning some of the scripts were not as effective as desired in tracing Mechanisms to the CAS Book of Knowledge and automated generation of Architecture Matrix Math." "in the Objective Architecture, the use of Tailored Communications TSS (a Mechanism specified in CAS) was identified to support enhanced safety, security in the design. The MSI did not use the Tailored Communications approach but used a more general Communications TSS (Transport Service Segment)." "additional work needs to be done to automate the understanding of such areas as flowdown of Mechanisms in a proper manner to insure that decisions made at the Reference and Objective Architecture levels are utilized at the System Architecture and Design levels or a understanding of why they were not followed is documented and discussed at the proper

time to understand why a variation is necessary." (Skayl model conformance report)

Skayl described the role of "architect" as evolving but said that one responsibility of the architect could be to manage the modeling toolchain, in which case the "architect" would be a significant contributor to the ASoT.

"We performed additional efforts during the initial requirements elicitation due to multiple sources of originating requirements... When we started execution, we lacked an understanding of the relationships between these requirements." "Duplication of requirements between different sources to the MSI(Architecture, system specification, system analyses, etc.,) creates extra work for the MSI and creates the potential for confusion as the different sources may define the same requirement in different(and potentially conflicting) ways." (Collins)

"It is highly likely that each MSI's approach to functional decomposition differed..." (Raytheon/GE)

Collins used the ACVIP Template to perform virtual integration with their subcontractor University of Alabama at Huntsville (UAH), providing UAH with a model-based specification of the more detailed component model to be delivered by UAH. The delivered model was virtually integrated before the actual component was implemented and delivered. Collins reported this was successful and provided significant benefit.

"Information from several sources needed to be combined somewhere to compose a 'procurement package' that completely defines the system to be developed..." "Third party model content is critical to cost effective MBSE implementation... This will be greatly enhanced by having the vendor provide models that capture all (or at least some) of the development artifacts (requirements and design) as well as key characteristics necessary for a defined set of virtual analyses." (Collins)

The government used a SharePoint site to exchange models with and between performers. The performers themselves used PLM and ALM tools and repositories to exchange models and other development assets within their own teams.

The Boeing team experienced a variety of challenges associated with managing a geographically distributed team. Boeing described the the initial network setup as straightfoward (1-2 days of effort) but then debugging network issues significantly slowed them down (Tyler Boeing closeout notes p2).

"Model configuration control was performed using various git tools, including Bitbucket, Tortoise Git, and Source Tree, and two separate git repositories (one GE and one Raytheon). This configuration was managed using branches from the two "master" branches, or trunks, and various baselines for coordination amongst GE and Raytheon." (Raytheon/GE)

"Model collaboration within the design group is achieved using Enterprise Architect's Design Master/Replica feature." "The AADL Model is configuration controlled by Boeing through the use of GitLab containing the source repository that makes up the workspace for the OSATE application. Boeing configuration management is the "gatekeeper" of merging and moving content in and out of the GitLab repository, while Sikorsky downloads/accepts content through the use of the SharePoint." (Boeing/Sikorsky OSMP)

"In retrospect, the MSI2 team should have spent the time up front to set up a live Enterprise Architect (EA) model that all MSI2 team participants could view and modify. EA does have

support for live models with CM, but the team did not take this approach due to the Information Technology (IT) challenges of collaboration between the teams." (Boeing/Sikorsky)

"One unique aspect of RTX/GE team's demo system is that the system is comprised of components deployed in two geographically separated system integration labs (SIL)." (Raytheon/GE)

"Integration efforts into the three MSI systems were not smooth and didn't go as expected. Several factors contributed to this problem including documentation... as well as differences in integration approaches by the three teams." "Documents were not easy to go through and multiple versions of the documents being released led to confusion between the product developer and mission system integration teams." (Honeywell)

"IP protection in model environment needs to be solved..." (Collins)

The MSIs drew legacy software components from prior projects or organization product lines, modifying or wrapping them as needed to use them in their Capstone demonstrations and excursions.

8.4.2 Discussion

Each organization will have its own internal ASoT to suite its business interests, product lines, processes, and guidelines. Each will have their own development environment suited for their needs. Different organizations will select different tools. These will inevitably include in-house as well as commercial tooling.

As noted in the *Discoverability* discussion, different layers of the CAS architecture hierarchy were done by different organizations using different interpretations of the GFI and different modeling styles. Many organizations use internal modeling guidelines and profiles. Organizations use different tools, and tools often recognize certain specific modeling idioms or rely on profiles that come with the tools. This complicates the problem of understanding relationships between models from different organizations and integrating information and models.

Most performers said model user guides are needed. A Software and Model User Guide (SMUG) was included in a software component TDP by the provider (Honeywell). However, there was still misunderstanding about aspects of the components that were necessary for integration. Significant technical support from the software component provider to each of the MSIs was needed.

An open issue is the degree to which GFI should be proscriptive versus prescriptive. Automation using tools will drive this towards more proscriptive and prescriptive but less flexible, less tolerant to organizational variability and able to handle legacy or OTS.

8.4.3 Recommendations

In practice, plan for each organization to have its own ASoT. The challenge is to establish coordinated use of multiple organization ASoTs so they all see a consistent ASoT for the projects on which they are all collaborating.

8.5 Interoperability Between Modeling Environments

8.5.1 Observations

"A single model will likely never (or at least not in the near term) contain all information necessary to define/characterize, design, verify and qualify systems and components." (Collins)

"Not all development tasks (e.g. requirements, design, and verification) for all systems, subsystems, and components will be executed in a model environment." (Collins)

"Functional requirements should be represented only by elements in the model, not as text. Having both model elements as well as text describe requirements creates redundancy and potential conflicts in the model." (Raytheon/GE)

"Two CF components, compliant to the same FACE data model and interface, behaved differently." (Raytheon/GE observation about the need for information captured in both FACE and non-FACE models.)

"The MSI2 team had models in a variety of formats and used a number of tools to manage those models. A bulk of the systems engineering work was captured in SysML developed in Enterprise Architect (EA). In addition, the team had FACE and AADL data models." "Future projects should focus on having as much of the systems engineering and system definition within a single coherent model set." (Boeing/Sikorsky)

8.5.2 Discussion

MSIs sometimes referred to a "Single Source of Truth" as a single model (usually SysML) that contained everything or was the source of truth for everything. This caused several problems. First, any limitations or flaws in the tool supporting the Single Source of Truth (in this case, Enterprise Architect) would lead to bottlenecks in design progress. Second, if the single source of truth model was not configuration managed effectively (e.g., was shared via email, did not support merge/branch, etc) additional effort was added in collaborating, prototyping, and sharing the model. Third, use of a single model is problematic when translators between between model representations do not support incremental updates. For example, the FACE import tool for Enterprise Architect cannot *merge* changes to a FACE model in `.face` format into an existing Enterprise Architect model, significantly increasing the effort required to synchronize FACE models between organizations. The architect (Skayl) argued that SSoT was *not* realized on Capstone. They argued that more traceability is needed, and there is a need to get away from "all data in all models." (notes from Capstone close out) SysML, FACE, and AADL models were all important.

Certain information from the GFI SysML model was provided as a FACE DSDM in the GFI. Certain information was flowed by the performers from SysML and FACE models to AADL models. The FACE and AADL models also contained information that was not found in other models. FACE tools and models were used to create Unit of Portability Supplied Models (USMs) from the DSDM, adding information in order to do so. AADL generated from SysML and AADL models was manually extended and refine to add properties and other details needed to perform various analyses.

The DSDM was provided in FACE format in the GFI. A FACE-to-AADL tool was available early in the program. Delayed availability of a mature SysML-to-AADL capability resulted in substantial work to manually create and maintain AADL. All performers reported that automated model synchronization tooling will be essential in a production project.

As noted in the *Interoperability Between Organizations* section, fairly prescriptive and proscriptive modeling guidelines are needed if tools are applied to models to automate certain activities. In the previous example, this was a CAS conformance tool developed by Skayl. Collins initially used an internal AADL profile and a translation script written in Python to generate AADL from suitably stereotyped elements of a SysML model. Their profile aligned sufficiently well with our SysML-to-AADL bridge tool that they said they would likely adopt that because it had more features and would require less maintenance from them. However, Boeing used a number of modeling idioms that did not align with those AADL profiles or did not contain all the information needed for translation. To our knowledge, two tools have been developed to generate FACE from suitably structured and stereotyped SysML; we have not examined those.

One model may be the source of truth for some items while another model is the source of truth for other items. In some cases a property may be used in a model with different intentions. For example, when one model is used to specify a more detailed model that is being procured (as happens with GFI), properties in the first model are to be interpreted as requirements that are to be subsequently verified by analysis of the responding model. Properties in a model could be interpreted as either a specification to be satisfied by an implementation or as a statement of fact about a legacy implementation that is being reused. (The ACVIP Modeling & Analysis Handbook provides guidelines for such cases.)

8.5.3 Recommendations

Select or develop modeling guidelines and other assets, such as standards, profiles, and tools, to automatically synchronize common information found in two models. This must be done in a way that the source-of-truth model for each common information item is clearly identified. For example, where portions of an AADL model are generated from elements found in a SysML model, the SysML model is the source of truth for the generated AADL model elements.

9 Conclusion and Next Steps

ASoT is a complex, difficult-to-constrain subject. This study explored many facets of the ASoT concept. In this report we provide recommendations and guidance on a variety of topics, but the core takeaways are:

- Use and invest in open, flexible standards like OSLC.
- Procure with ASoT in mind.
- Use Connectivity tools and Model-Based Engineering.

To put the findings of this study into practice, the next step is to tailor these findings for

a particular program. For example, The access control ASoT requirements provided by this study (in Appendix A) need to be tailored to the access control policies of a given program. Similarly, a program will need to tailor the procurement guidance we provided in Section 5 to form a ready-to-use procurement process.

In future research we hope to explore technologies and strategies for improving discoverablity of data in an ASoT and for facilitating automated translation and traceability between tools and repositories.

A ASoT Requirements

Authoritative Source of Truth (ASoT) Requirements

Contract: W911W6-19-F-703D

Mr. Charles Payne
Dr. August Schwerdfeger
Mr. Tyler Smith

DISCLAIMER:
This material is based upon work supported by the U.S. Army Combat Capabilities Development Command Aviation & Missile Center under contract no. W911W6-17-D-0003/W911W6-19-F-703D. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of U.S. Army Combat Capabilities Development Command Aviation & Missile Center.

Table of Contents

1. Introduction

This document contains requirements for minimum, recommended, high level capabilities for an Authoritative Source of Truth (ASoT). We derived these capabilities from user stories collected from stakeholders representing the Government and Industry. The capabilities themselves, called Use Case Specifications, derive to requirements that technology and organizations providing the ASoT must satisfy.

As we collected the user stories and derived their capabilities, we began to notice emerging themes that motivated us to divide the capabilities into fourteen topic areas. Section 2 provides a brief overview of each topic area and includes links to their supporting capabilities in Section 3, which lists each capability (again by topic area) and provides a detailed mapping from the capability back to its user stories and forward to its requirements. Finally, Section 4 lists the user stories themselves.

We provided the topic area summaries and their capabilities to the Capstone performers for review and received responses from three performers (Raytheon, Boeing, and Honeywell), which appear in Section 2.

The reviewer should begin in Section 2 and follow the hyperlinks to other sections.

2. Topic Area Summaries

2.1. Overview

This section summarizes the Authoritative Source of Truth (ASoT) Use Case Specifications that derive from the ASoT user stories. Each Use Case Specification is a brief capability description for an ASoT stakeholder. We collected these specifications into groups according to shared characteristics: Access Control, Authoritative State, Autonomous Operation, Certification, Collaboration, Configuration Control, Custom Views, Metadata, Oracle, Recompete, Resilient Operation, Traceability, Tradeoff Analysis, and Workflows. We discuss each group in its own section below. Within each section, numbers in parentheses refer to Use Case Specifications that support this topic area. Some Use Case Specifications support multiple topic areas. We also include supporting rationale that we collected during the stakeholder interviews. The list of capabilities below is not exhaustive. It suggests only what an ASoT might provide. Existing tools provide many of these capabilities, and the ASoT may leverage those tools in some way. Some capabilities may be beyond the scope of what an ASoT should provide, but we include those capabilities for further discussion.

Performer Comments:

- Honeywell: As an inplementation example, this might involve a combination of tools: e.g. Git and Artifactory, both enabled on an appropriately accessible server. The point of the example is to make clear that the result of this may be a suite of tools rather than a single tool itself.

2.2. Access Control

The ASoT must restrict access to stakeholder intellectual property according to typical security policies (3179, 3152, 3145, 3127, 3126, 3124, 3053, 3052), restricting both regular users and ASoT administrators (3125). The ASoT must revoke access when required (3177), and stakeholders must be able to verify the access permissions for their own artifacts (3368).

Supporting Rationale. During stakeholder interviews, Capstone performers expressed concern about unauthorized access to their intellectual property. One performer noted that the current ASoT prototype (SharePoint) does not let the performer verify access permissions to the assets shared there.

Performer Comments:

- Boeing: works in progress aren't yet 'truth'. So why in ASoT? Our response: This is probably already covered by 3124.

2.3. Authoritative State

The ASoT must support virtual integration of digital artifacts (3366). The Government Program Manager (PM) for the system of interest defines what is authoritative for that system (3400), but the PM may not own or control all of the artifacts so designated. Some artifacts may be owned or controlled by other Government PMs or outside suppliers, each with their own ASoT to manage the artifact. As a result, the ASoT for this PM may be a distributed collection of ASoTs that this PM designates authoritative at a given point in time (3151). Regardless of the composition, the PM views its ASoT as a single, centralized respository (3138) that documents the design authorized for virtual integration (3129) along with evidence generated by approved analysis tools (3403) that the design obeys its constraints (3184) and that the design reflects an as-built system (3405) that will pass certification (3382, see Certification below) and the corresponding analysis results (3171).

Supporting Rationale: During stakeholder interviews, Capstone performers expressed concern about how digital artifacts would transition from their control to the Government's control and how the ASoT would handle the evolvng product line variants that are a fact of life for these performers. One performer suggested that the ASoT would largely manage relationships between digital artifacts rather than the artifacts themselves.

Performer Comments:

- Boeing: Is this to imply as-maintained will not be captured in a ASoT? That information is important when a system is sent back for upgrade/modification, life extension analysis, etc. Our response: I was thinking of the virtual model's analysis results to the system as built. A system as_maintained would be an as_built system. So the terms and purpose would need to be clarified. Also the as_maintained would be a version under 3173 that could be compared to the as_built.

- Raytheon: On 3400, does this mean government approved integration tools or artifacts showing successful integration? Our response: We meant artifacts that the government approves for integration, e.g., integrate version X not version Y.

- Honeywell: ASoT managing the relationships between digital artifcats and not the artifcats themselves sounds good. Maintaining the relationships should include build processes. This means things like Maven, Bamboo, etc. configurations.

2.4. Autonomous Operation

To support the virual integration of highly complex systems, the ASoT must operate as independently of the human as possible. This includes automatically running analysis on updated models (3172) and automatically detecting new artifacts for analysis (3350). The ASoT should run analyses continuously (3163), including custom analyses (3361) that expose problems earlier to reduce risk (3146). When necessary, the ASoT should automatically translate models to support analysis tools (3363). Analysis results should inform system-wide metrics that are more easily consumed (3379). Finally, the ASoT should automatically generate required evidence to support certification (3170) and automatically generate code from updated models (3349, 3185).

Supporting Rationale. From the Capstone activity, we know that the performers want to perform model analysis on their own to reduce risk prior to delivery. During future stakeholder interviews, we should ask these performers if their processes will support the autonomous operation suggested above.

Performer Comments:

- Boeing: This seems like a scope growth in what a ASoT was defined to be. You're now way beyond 'communicating' what is current truth. You must be able to run simulation thru it. I think you're now in a Digital Twin vs ASoT; that would grow the requirement set and context of answers provided to date. Our response: Our approach to ASoT supports more than simulations. It supports the capabilities to enable integration of digital models for analysis and implementation throughout the lifecyle.
- Raytheon: On 3163, is STPA analysis expected to be autonomous? Our response: Good point, STPA itself is not automated, but analysis that supports STPA may be autonomous.
- Honeywell: This should support Continuous Integration/Continuous Delivery (CI/CD). The autodetection of new artifacts, and to rerun analysis when new artifacts are introduced is great. The capability to translate models to standard modeling tools would be incredibly useful.

2.5. Certification

The ASoT must simplify the system's path to certification compared to typical approaches. The ASoT can simplify certification by including common tools and processes (3381), by capturing certification evidence (3147), and by automatically evaluating a design's impact on certification (3382). The ASoT should support new processes for certification (3143) and support the reuse of certification evidence where possible (3385).

Performer Comments:

- Boeing: This almost sounds like you more need a proper M&S Plan that requires all this meta-data be captured and turned in with submittals. I don't think you're going to actually create a link to a specific tool version vs capture which tool version the analysis we performed in? The mention of processes is also interesting in that process is where suppliers typically claim areas of IP; unless this was meant in a more general sense? Our response: For your first question, we would, as you suggest, capture the tool version that performed the analysis. This capability relates to 3403: track approved analysis tools, including those with waivers. For your second question, good point. We can only track processes (e.g., analysis processses) performed after digital artifact submission, not the processes, say, that a supplier follows to develop the digital artifact (before submission).

2.6. Collaboration

In addition to identifying digital artifacts using metadata, the ASoT provides collaboration features to support the processes that produce those artifacts. Government stakeholders require collaborative spaces to document decision rationale and synchronize on terminology (3367). The ASoT facilitates collaboration by providing artifacts in a standard, searchable format (3141).

2.7. Configuration Control

To track authoritative truth over time, the ASoT must implement or leverage the traditional capabilities of a version control system (3180, 3167, 3371, 3173, 3346, 3142). For the ASoT, the change control board may be the Government PM itself (3131), and the artifacts managed will include GFI (3166).

Supporting Rationale. During stakeholder interviews, Capstone performers noted they already have processes in place to transition artifact change control from their suppliers. In some cases when they discover an issue, they ask the supplier to fix it and make a new delivery. In other cases, they make the change to the delivered artifact and notify the supplier. The ASoT must support both approaches.

Performer Comments:

- Raytheon: When ASoTs are outside of a PM's control (i.e. the system ASoT is a collection of ASoTs), how is he envisioned to maintain CM on that ASoT for his program? Our response: Great question. We assume each ASoT (in the collection) has its own CM and a way to relate the controlled artifacts between them.

- Raytheon: On 3131, this will include changes from supplier products that are outside the control of the government and/or primes (especially COTS items).

- Honeywell: A standard for how the supplier or performer should handle artifact changes should be set, with the ASoT being able to adopt this standard. There may be a need to support multiple types of change requests. E.g., some might be minor editing, some might be additions (no downside impact), and some might be deprecations/removals (potential unknowable impact). Ideally the CR tool should be integratedf with the artifact management so that we support Continuous Integration/Continuous Delivery (CI/CD).

2.8. Custom Views

Stakeholders want different perspectives of the ASoT, so the ASoT must support user-defined views (3402). The ASoT could provide a self-service layer that enables stakeholders to build their own experience and desired level of interaction (3133). The layer must support an ASoT that is a collection with no single point of control (3150). Some stakeholders desire a system-level perspective that doesn't require detailed system knowledge (3376, 3162, 3159, 3154), while other stakeholders will require that detail (3365, 3135).

Performer Comments:

- Raytheon: On 3133, flexibility to build custom views is good, but a solid set of basic views is critical to minimize initial investment.

2.9. Metadata

Not all queries of the ASoT will originate from the digital artifacts themselves. In some cases, users will construct queries using attributes, or metadata, of those artifacts. The metadata may be user-defined (3407), and the search criteria may be user-selected (3364, 3182, 3181, 3183, 3175, 3174, 3160) and include the user-defined metadata itself (3347). Some artifacts may require additional rationale (3153) that is stored as metadata. Queries may even identify models with the ability to perform certain analyses (3397).

Supporting Rationale. Some attributes (e.g., 3160) may require input from the supplier, so this information may be required at the time of delivery.

Performer Comments:

- Raytheon: Supplier data such as phase-out dates (or software support) need to be accessible for life cycle management.

2.10. Oracle

The ASoT tracks not only digital artifacts but also the process that produced those artifacts, so the ASoT may serve as a helpful oracle for completing that process. Users may query for contracting guidance and modeling templates (3401, 3140, 3139, 3137, 3359) supporting required processes (3168, 3134).

Performer Comments:

- Raytheon: On 3139, is this a link to a larger repository or does a project have to maintain its standards library? Our response: For practical reasons, we assume this to be a link to a larger, external repository.

2.11. Recompete

Note

We added this topic after we sent the other topics for performer review.

The ASoT must support the Government's ability to recompete development of different parts of the system. This area overlaps with the other topic areas in this section. For example, the ASoT must provide guidance similar to what was originally needed to generate the contract (3544, 3543, 3540, 3539, 3538, 3537, 3132), track delivery of those artifacts (3545), track GFI provided under previous contracts (3541), track artifacts that passed certification (3384) and reuse that evidence (3385), and share those artifacts between different ASoTs (3533).

2.12. Resilient Operation

The design of the ASoT itself must include resiliency features (3399, 3398, 3370, 3369) that minimize downtime (3178, 3176) and support the ability to stand up a new ASoT quickly when required (3161).

Supporting Rationale. During stakeholder interviews, Capstone performers expressed concern about their ability to manage staff and other resources if their development activity depends on a Government-controlled ASoT that is suddenly not available for an extended period.

Performer Comments:

- Boeing: At some point would define a specific acceptable recovery time and data loss criterion. Ex. less than x hrs down. Nightly back up so less than 24 hr of data loss possible, etc.

2.13. Traceability

To support certification/qualification, the ASoT must generate evidence that the system analyzed is the system as-built. While a change control system provides some benefits (3393, 3345, 3387, 3358), a typical change control system may not track all required relationships. In particular, a change control system tracks individual artifacts, but the ASoT should also be capable of tracking, at the ASoT owner's discretion, the processes that produce those artifacts and the resulting analyses of those artifacts. The ASoT needs to track the use of government template models by performers (3358), and the tools that produced those results (3404). The ASoT must also identify which results pass certification (3384). The ASoT must track functional and performance specifications (3165) and relate similar but distinct artifacts (3155). If the ASoT is itself a collection of ASoTs, there must be traceability between distinct members (3386). As the ASoT evolves over time, the Government PM must identify where artifacts are in their lifecycle (3375). The ASoT should inform stakeholders automatically of actions affecting artifacts they own or control (3351). The ASoT must repeat analysis of evolving artifacts (3164) and be able to compare different analyses of the same artifact over time. Finally, the ASoT must support discovery of artifacts through these traceability links (3156).

Supporting Rationale. During future stakeholder interviews, we should ask whether stakeholders will be able to track the things called out here while the artifacts are in their control. Stakeholders may need to update their existing processes to support the ASoT.

Performer Comments:

- Raytheon: Traceability must include provisions for dealing with obsolescence issues.
- Honeywell: This should support Continuous Integration/Continuous Delivery.

2.14. Tradeoff Analysis

The ASoT distinguishes itself from conventional system development tools by this and the following capabilities. Stakeholders want the ability to instrument their models with variation points (3388, 3158), analyze those variations (3008, 3149), calculate change impact (3157), and explore the resulting trade spaces (3377). The ASoT can map feature sets to different environments (3396) and automatically introduce variation points as system constraints in the trade space (3378). The ASoT should also be capable of supporting enterprise, Family of Systems (FoS), and System Architecture tradeoff analyses.

Performer Comments:

- Raytheon: Another area that needs to support obsolescence (and trade space for resolving).
- Honeywell: The capabilities surrounding the tradeoff analysis in terms of calculating change impacts and the ability to explore the trade spaces is also incredibly useful. Sounds like support for Dependency Injection directly from the tool.

2.15. Workflow

In addition to the typical capabilities of workflow management (3380, 3362, 3348, 3148), the ASoT can include activities unique to certification (3128), including automated analysis at workflow-defined intervals (3352). The following capabilities support this discussion.

3. Use Case Specifications By Topic Area

3.1. Access Control

3368 The ASoT supports the government auditor by verifying that the ASoT properly restricts access to controlled data

User Stories: 2589, 2429, 3335

Derived Requirements:

3468 The organization owning the information shall define security policy protecting digital artifacts according to their information sensitivity.

3179 The ASoT supports all stakeholders by restricting access according to export control criteria

User Stories: 2693, 2702

Derived Requirements:

3468 The organization owning the information shall define security policy protecting digital artifacts according to their information sensitivity.

3432 The ASoT shall implement security policy protecting digital artifacts according to their information sensitivity.

3177 The ASoT supports the ASoT administrator by providing immediate access revocation for departing stakeholders/users

User Stories: 3110

Derived Requirements:

3465 The ASoT shall maintain a record of ownership for each model artifact.

3152 The ASoT supports the supplier by restricting visibility into the supplier's internal methods, processes, and artifacts

User Stories: 2609

Derived Requirements:

3465 The ASoT shall maintain a record of ownership for each model artifact.

3467 The organization owning the ASoT shall define security policy protecting digital artifacts according to their contractual rights.

3145 The ASoT supports the supplier by providing support to share source code with data rights to prevent modification or re-use

User Stories: 2620

Derived Requirements:

3465 The ASoT shall maintain a record of ownership for each model artifact.

3464 The ASoT shall provide a means to associate licensing information with a model artifact.

3127 The ASoT supports all stakeholders by restricting access to Works in Progress

User Stories: 2494, 2425, 2446

Derived Requirements:

3467 The organization owning the ASoT shall define security policy protecting digital artifacts according to their contractual rights.

3433 The ASoT shall implement security policy protecting digital artifacts according to their contractual rights.

3126 The ASoT supports all stakeholders by restricting access by Need to Know classifications

User Stories: 3394, 2591, 2432, 2429, 2448, 2694, 2702

Derived Requirements:

3468 The organization owning the information shall define security policy protecting digital artifacts according to their information sensitivity.

3432 The ASoT shall implement security policy protecting digital artifacts according to their information sensitivity.

3125 The ASoT supports all stakeholders by restricting ASoT staff to admin functions

User Stories: 2450

Derived Requirements:

3466 The organization owning the ASoT shall define security policy protecting digital artifacts according to organizational role.

3435 The ASoT shall implement security policy protecting digital artifacts according to organizational role.

3124 The ASoT supports all stakeholders by restricting access by contractual agreement

User Stories: 2585, 2583, 2494, 2430, 2416, 2410, 2452, 2454, 2465, 2694, 2702

Derived Requirements:

3467 The organization owning the ASoT shall define security policy protecting digital artifacts according to their contractual rights.

3464 The ASoT shall provide a means to associate licensing information with a model artifact.

3465 The ASoT shall maintain a record of ownership for each model artifact.

3428 The ASoT shall associate security attributes with digital artifacts and users to enforce access control.

3433 The ASoT shall implement security policy protecting digital artifacts according to their contractual rights.

3431 The ASoT shall enforce access to digital artifacts according to their security attributes and defined security policy.

3429 The ASoT shall implement security attributes to represent contractual obligations.

3053 The ASoT supports all stakeholders by restricting access by defined (e.g., SBIR) data rights

User Stories: 2587, 2430, 2421, 2416, 2458

Derived Requirements:

3467 The organization owning the ASoT shall define security policy protecting digital artifacts according to their contractual rights.

3464 The ASoT shall provide a means to associate licensing information with a model artifact.

3465 The ASoT shall maintain a record of ownership for each model artifact.

3052 The ASoT supports all stakeholders by restricting access by NDA or ACA

User Stories: 2427, 2460

Derived Requirements:

3464 The ASoT shall provide a means to associate licensing information with a model artifact.

3465 The ASoT shall maintain a record of ownership for each model artifact.

3467 The organization owning the ASoT shall define security policy protecting digital artifacts according to their contractual rights.

3.2. Authoritative State

3406 The ASoT supports all stakeholders by identifying expired artifacts for removal or archiving

User Stories: 2438, 2496

Derived Requirements:

3446 The ASoT shall provide a means to associate an expiration date with a model artifact as metadata.

3436 The ASoT shall associate metadata with digital artifacts to enable search and other functions.

3405 The ASoT supports all stakeholders by providing evidence that the models reflect the as-built/ as-maintained system

User Stories: 2496, 2438

Derived Requirements:

3458 The ASoT shall provide a means to sign a model artifact with a cryptographic checksum.

3499 The ASoT shall provide the means to designate a digital artifact as approved for virtual integration.

3403 The ASoT supports the Government program manager by tracking approved analysis tools, including those with waivers

User Stories: 2532

Derived Requirements:

3460 The ASoT shall maintain a registry of approved modeling and analysis tools, supporting different versions thereof.

3400 The ASoT supports all stakeholders by restricting virtual integration to Government-approved artifacts

User Stories: 2595, 2593

Derived Requirements:

3452 The ASoT shall provide a version control system for storing and managing digital artifacts.

3463 The ASoT shall provide a means to associate one or more certifications with a specific version of a model artifact.

3384 The ASoT supports certification authorities by tracking which specific digital artifacts passed certification.

User Stories: 2669, 2647

Derived Requirements:

3463 The ASoT shall provide a means to associate one or more certifications with a specific version of a model artifact.

3366 The ASoT supports all stakeholders by enabling the submission of digital artifacts for integration into the system design

User Stories: 3338

Derived Requirements:

3498 The ASoT shall provide a means to associate digital artifacts with a specific system design.

3184 The ASoT supports all stakeholders by enforcing constraints

User Stories: 2684, 2685, 2686 3050

Derived Requirements:

3485 The ASoT will manage constraints as digital artifacts.

3171 The ASoT supports the government approval authority by linking analytical results to the artifacts that produced those results

User Stories: 3121, 3119, 3123, 3051, 3050, 3049, 3046

Derived Requirements:

3452 The ASoT shall provide a version control system for storing and managing digital artifacts.

3454 The ASoT shall provide a means to associate a set of analysis results, or a portion thereof, with specific versions of model artifacts.

3453 The ASoT shall provide a uniform representation for stored analysis results that allows the ASoT to be aware of their structure down to the level of individual data.

3151 The ASoT supports all stakeholders by providing a capability to define which ASoT in a set of ASoTs is the authoritative source at a given point in time

User Stories: 2417, 2610, 3356

Derived Requirements:

3511 The ASoT shall provide a peer-to-peer interface facilitating the synchronization of several independent ASoTs in joint service.

3138 The ASoT supports the government approving authority by providing a central repository for artifact generation, action notification, and artifact routing

User Stories: 2478

Derived Requirements:

3452 The ASoT shall provide a version control system for storing and managing digital artifacts.

3129 The ASoT supports the supplier by providing evolving design details necessary for requirements, design, and integration

User Stories: 2465, 3051, 3050, 3049

Derived Requirements:

3433 The ASoT shall implement security policy protecting digital artifacts according to their contractual rights.

3470 The ASoT shall collect and store details required for virtual integration.

3.3. Autonomous Operation

3379 The ASoT enables all stakeholders to reduce program risk by automatically analyzing digital artifacts and reporting system architectures and metrics.

User Stories: 2651, 2661

Derived Requirements:

3502 The ASoT shall provide hooks to schedule automatic execution of user-defined actions on specified triggers.

3501 The ASoT shall provide an interface for user-selected tools to access artifacts programmatically.

3504 The ASoT shall provide an interface for automatically generating documentation for artifacts.

3363 The ASoT supports all stakeholders by supporting custom modeling tools with automatic translation to standard modeling tools

User Stories: 3340

Derived Requirements:

3506 The ASoT shall maintain a definitive list of standard modeling tools that it supports.

3507 The ASoT shall provide an interface facilitating translation of artifacts into any format required by the standard modeling tools it supports.

3361 The ASoT supports all stakeholders by supporting user-selected tools for automatic analysis

User Stories: 3344

Derived Requirements:

3501 The ASoT shall provide an interface for user-selected tools to access artifacts programmatically.

3502 The ASoT shall provide hooks to schedule automatic execution of user-defined actions on specified triggers.

3350 The ASoT supports all stakeholders by automatically detecting new artifacts and re-executing analysis where required

User Stories: 3270

Derived Requirements:

3455 The ASoT shall provide hooks allowing the storage of a new version of a model artifact to trigger actions.

3502 The ASoT shall provide hooks to schedule automatic execution of user-defined actions on specified triggers.

3349 The ASoT supports all stakeholders by generating code automatically at regular intervals from user-provided build scripts

User Stories: 3271

Derived Requirements:

3503 The ASoT shall provide hooks to trigger actions automatically at specified times or intervals.

3502 The ASoT shall provide hooks to schedule automatic execution of user-defined actions on specified triggers.

3185 The ASoT supports all stakeholders by generating the list of changes that led to a constraint violation

User Stories: 2419, 2685

Derived Requirements:

3453 The ASoT shall provide a uniform representation for stored analysis results that allows the ASoT to be aware of their structure down to the level of individual data.

3452 The ASoT shall provide a version control system for storing and managing digital artifacts.

3454 The ASoT shall provide a means to associate a set of analysis results, or a portion thereof, with specific versions of model artifacts.

3531 The ASoT shall support the capture and systematic storage of artifacts from builds and tool runs, including stack traces and other error reports.

3170 The ASoT supports all stakeholders by providing automatic documentation generation

User Stories: 2706

Derived Requirements:

3504 The ASoT shall provide an interface for automatically generating documentation for artifacts.

3505 The ASoT shall allow users to specify custom formats and layouts (a house style) for automatically generated documentation and reports.

3163 The ASoT supports the government approval authority by providing safety and cybersecurity analysis at all levels and models for review

User Stories: 2633

Derived Requirements:

3508 The ASoT shall provide tooling allowing subject matter experts to perform analysis on models, and to store the results of this analysis in a systematic way.

3154 The ASoT supports all stakeholders by providing support for integrated, cross-domain perspectives of continuous, virtual integration in the face of changing mission needs

User Stories: 2684

Derived Requirements:

3524 The ASoT shall provide a means of scripting actions exposed via its programmatic interface.

3522 The ASoT shall provide programmatic access to its artifacts and user interfaces.

3146 The ASoT supports all stakeholders by providing support for uncertainty modeling and management and risk assessment and management

User Stories: 2615

Derived Requirements:

3556 The ASoT shall enable the programmatic execution of user-defined analysis over digital artifacts.

3172 The ASoT supports all stakeholders by automatically rerunning analysis on updated models

User Stories: 3121, 3045

Derived Requirements:

3455 The ASoT shall provide hooks allowing the storage of a new version of a model artifact to trigger actions.

3452 The ASoT shall provide a version control system for storing and managing digital artifacts.

3502 The ASoT shall provide hooks to schedule automatic execution of user-defined actions on specified triggers.

3.4. Certification

3537 The ASoT supports all stakeholders by providing specific references to the contract requirements used to produce digital artifacts.

User Stories: 2396

Derived Requirements:

3547 The ASoT shall associate with each artifact the requirements used to produce it.

3385 The ASoT supports program managers by enabling reuse of certification digital artifacts across projects.

User Stories: 2647

Derived Requirements:

3463 The ASoT shall provide a means to associate one or more certifications with a specific version of a model artifact.

3384 The ASoT supports certification authorities by tracking which specific digital artifacts passed certification.

User Stories: 2669, 2647

Derived Requirements:

3463 The ASoT shall provide a means to associate one or more certifications with a specific version of a model artifact.

3382 The ASoT supports certification authorities and system engineers (stakeholder roles) by automatically evaluating the certification impact of design or requirements changes.

User Stories: 2648

Derived Requirements:

3452 The ASoT shall provide a version control system for storing and managing digital artifacts.

3455 The ASoT shall provide hooks allowing the storage of a new version of a model artifact to trigger actions.

3479 The ASoT shall evaluate the certification impact given the results from certification-related analyses.

3502 The ASoT shall provide hooks to schedule automatic execution of user-defined actions on specified triggers.

3381 The ASoT supports system certification authorities (stakeholder role) by allowing inclusion of certification tools, mechanisms, and processes.

User Stories: 2647, 2649

Derived Requirements:

3477 The ASoT shall provide for the execution of certification tools, mechanisms, and processes.

3147 The ASoT supports the system integrator by providing support for virtual airworthiness qualification and cybersecurity certification processes and evidence capture

User Stories: 2614

Derived Requirements:

3478 The ASoT shall enable the execution of a virtual airworthiness qualification process including evidence capture.

3143 The ASoT supports the government program manager by providing support to define a new process for certification of airframe modifications

User Stories: 2440, 2423, 2627

Derived Requirements:

3480 The ASoT shall provide a method to create or modify a virtual certification process.

3.5. Collaboration

3367 The ASoT supports all stakeholders by providing a collaborative space to exchange notes and synchronize on terminology

User Stories: 3337

Derived Requirements:

3436 The ASoT shall associate metadata with digital artifacts to enable search and other functions.

3526 The ASoT shall provide a means for creating collaborative glossaries of terms, and to establish equivalence relations among them.

3527 The ASoT shall provide a means to associate glossary terms with model artifacts.

3149 The ASoT supports the government program manager by providing support for Multi-Disciplinary Analysis and Optimization (MDAO) across assets

User Stories: 2612

Derived Requirements:

3556 The ASoT shall enable the programmatic execution of user-defined analysis over digital artifacts.

3141 The ASoT supports all stakeholders by providing deliverables in a standard, searchable format

User Stories: 2471

Derived Requirements:

3449 The ASoT shall provide a uniform representation for stored deliverable materials.

3450 The ASoT shall provide an interface for searching stored deliverable materials.

3.6. Configuration Control

3371 The ASoT supports all stakeholders by enabling the merging of downstream changes into baseline files

User Stories: 2419, 3360

Derived Requirements:

3452 The ASoT shall provide a version control system for storing and managing digital artifacts.

3456 The ASoT shall provide an interface to compare different versions of a model artifact.

3473 The ASoT shall provide an interface for recursive three-way merging between different versions of a given artifact.

3475 The ASoT shall provide a means to check out artifacts in a working-copy form, with a link retained to the upstream source.

3346 The ASoT supports all stakeholders by providing an issue tracking system for its own errors

User Stories: 3276

Derived Requirements:

3471 The ASoT shall support issue tracking and resolution.

3474 The ASoT shall have the capability to treat itself as one of its own artifacts.

3185 The ASoT supports all stakeholders by generating the list of changes that led to a (design) constraint violation

User Stories: 2419, 2685

Derived Requirements:

3453 The ASoT shall provide a uniform representation for stored analysis results that allows the ASoT to be aware of their structure down to the level of individual data.

3452 The ASoT shall provide a version control system for storing and managing digital artifacts.

3454 The ASoT shall provide a means to associate a set of analysis results, or a portion thereof, with specific versions of model artifacts.

3531 The ASoT shall support the capture and systematic storage of artifacts from builds and tool runs, including stack traces and other error reports.

3180 The ASoT supports all stakeholders with a design that supports rigorous version control

User Stories: 2695

Derived Requirements:

3452 The ASoT shall provide a version control system for storing and managing digital artifacts.

3173 The ASoT supports the government approval authority by providing the capability to compare different versions of the same artifact

User Stories: 3121

Derived Requirements:

3452 The ASoT shall provide a version control system for storing and managing digital artifacts.

3456 The ASoT shall provide an interface to compare different versions of a model artifact.

3172 The ASoT supports all stakeholders by automatically rerunning analysis on updated models

User Stories: 3121, 3045

Derived Requirements:

3455 The ASoT shall provide hooks allowing the storage of a new version of a model artifact to trigger actions.

3452 The ASoT shall provide a version control system for storing and managing digital artifacts.

3502 The ASoT shall provide hooks to schedule automatic execution of user-defined actions on specified triggers.

3167 The ASoT supports the government program manager by supporting the activities of a change control board

User Stories: 2632

Derived Requirements:

3476 The ASoT shall provide a means to stage proposed changes or new versions of artifacts for approval by an authority before their full entry into the version control system.

3166 The ASoT supports the government program manager by supporting modification/clarification of Government-furnished digital artifacts as new details emerge

User Stories: 2632

Derived Requirements:

3497 The ASoT shall manage Government-furnished digital artifacts.

3142 The ASoT supports all stakeholders by supporting the capture of issues and evidence for issue resolution

User Stories: 2487, 2488, 2489

Derived Requirements:

3471 The ASoT shall support issue tracking and resolution.

3531 The ASoT shall support the capture and systematic storage of artifacts from builds and tool runs, including stack traces and other error reports.

3131 The ASoT supports all stakeholders by supporting issue resolution by the Government

User Stories: 2465

Derived Requirements:

3471 The ASoT shall support issue tracking and resolution.

3.7. Custom Views

3402 The ASoT supports all stakeholders by providing a dashboard with custom views

User Stories: 2540, 2539, 2538

Derived Requirements:

3453 The ASoT shall provide a uniform representation for stored analysis results that allows the ASoT to be aware of their structure down to the level of individual data.

3461 The ASoT shall provide an interface for querying analysis results on the level of individual data.

3462 The ASoT shall provide a dashboard, of a user-configurable arrangement, for viewing analysis results.

3376 The ASoT supports review stakeholders in multiple organizations by providing a perspective for the review that does not require detailed system knowledge.

User Stories: 2667

Derived Requirements:

3510 The ASoT shall provide a dashboard, of a user-configurable arrangement, for viewing artifacts, their metadata, and the associations between them.

3530 The ASoT shall provide a dashboard configuration allowing for a high-level view of a system.

3365 The ASoT supports all stakeholders by providing a dashboard to interact with defined data models

User Stories: 3339

Derived Requirements:

3510 The ASoT shall provide a dashboard, of a user-configurable arrangement, for viewing artifacts, their metadata, and the associations between them.

3159 The ASoT supports all stakeholders by providing a dashboard to view model variant architectures and analysis results

User Stories: 2576

Derived Requirements:

3482 The ASoT will provide a method to select product variants for analysis.

3483 The ASoT will provide a method to compare the analysis results of selected product variants.

3481 The ASoT will manage product variants.

3462 The ASoT shall provide a dashboard, of a user-configurable arrangement, for viewing analysis results.

3154 The ASoT supports all stakeholders by providing support for integrated, cross-domain perspectives of continuous, virtual integration in the face of changing mission needs

User Stories: 2684

Derived Requirements:

3524 The ASoT shall provide a means of scripting actions exposed via its programmatic interface.

3522 The ASoT shall provide programmatic access to its artifacts and user interfaces.

3150 The ASoT supports the government program manager by provding a (peer) interface to other ASoTs with no central point of control

User Stories: 2611

Derived Requirements:

3511 The ASoT shall provide a peer-to-peer interface facilitating the synchronization of several independent ASoTs in joint service.

3144 The ASoT supports the government program manager by providing a dashboard of sign-off approval through airworthiness review

User Stories: 2626

Derived Requirements:

3478 The ASoT shall enable the execution of a virtual airworthiness qualification process including evidence capture.

3462 The ASoT shall provide a dashboard, of a user-configurable arrangement, for viewing analysis results.

3135 The ASoT supports the government program manager by providing a dashboard for master schedule and cost data

User Stories: 2469, 2482

Derived Requirements:

3462 The ASoT shall provide a dashboard, of a user-configurable arrangement, for viewing analysis results.

3513 The ASoT shall provide a means to associate scheduling metrics with a model artifact as metadata.

3512 The ASoT shall provide a means to associate cost metrics with a model artifact as metadata.

3514 The ASoT shall provide a means of performing cost and schedule analyses over models and model artifacts.

3133 The ASoT supports all stakeholders by providing a self-service layer that communicates clearly and lets users build their own experience at the right level of interaction

User Stories: 2540

Derived Requirements:

3505 The ASoT shall allow users to specify custom formats and layouts (a house style) for automatically generated documentation and reports.

3510 The ASoT shall provide a dashboard, of a user-configurable arrangement, for viewing artifacts, their metadata, and the associations between them.

3462 The ASoT shall provide a dashboard, of a user-configurable arrangement, for viewing analysis results.

3502 The ASoT shall provide hooks to schedule automatic execution of user-defined actions on specified triggers.

3515 The ASoT shall have the capability to be operated and configured through more than one front-end or interface.

3.8. Metadata

3539 The ASoT supports all stakeholders by providing CDRLs for the contract containing data right markings (Could be stated in box 16 CDRL Form 1423) and required distribution statement (Box 9 and Box 16 of CDRL Form 1423).

User Stories: 2400

Derived Requirements:

3546 The ASoT shall associate data rights with all artifacts.

3407 The ASoT supports all stakeholders by providing the capability to associate user-defined metadata with all artifacts

User Stories: 2436

Derived Requirements:

3436 The ASoT shall associate metadata with digital artifacts to enable search and other functions.

3437 The ASoT shall enable the user to enter and modify metadata associated with digital artifacts according to the defined security policy for the artifact.

3397 The ASoT supports all stakeholders by assessing models against desired characteristics such as the ability to perform certain analyses

User Stories: 2603

Derived Requirements:

3460 The ASoT shall maintain a registry of approved modeling and analysis tools, supporting different versions thereof.

3459 The ASoT shall provide a means to associate a set of analysis results with specific versions of analysis tools.

3463 The ASoT shall provide a means to associate one or more certifications with a specific version of a model artifact.

3487 The ASoT shall enable search across user-defined and user-selected metadata.

3454 The ASoT shall provide a means to associate a set of analysis results, or a portion thereof, with specific versions of model artifacts.

3457 The ASoT shall provide an interface to query known versions of a model artifact.

3364 The ASoT supports all stakeholders by enabling searches across defined digital artifacts using user-selected search criteria

User Stories: 3339

Derived Requirements:

3436 The ASoT shall associate metadata with digital artifacts to enable search and other functions.

3437 The ASoT shall enable the user to enter and modify metadata associated with digital artifacts according to the defined security policy for the artifact.

3487 The ASoT shall enable search across user-defined and user-selected metadata.

3347 The ASoT supports all stakeholders by identifying artifacts based on user-defined attributes

User Stories: 2533, 2495, 2436, 3274

Derived Requirements:

3436 The ASoT shall associate metadata with digital artifacts to enable search and other functions.

3183 The ASoT supports all stakeholders by providing the ability to search components against their source or country-of-delivery

User Stories: 2693

Derived Requirements:

3447 The ASoT shall provide a means to associate a source with a model artifact as metadata.

3448 The ASoT shall provide a means to associate a country-of-delivery with a model artifact as metadata.

3182 The ASoT supports all stakeholders by tracking country-of-delivery for all components

User Stories: 2693

Derived Requirements:

3448 The ASoT shall provide a means to associate a country-of-delivery with a model artifact as metadata.

3181 The ASoT supports all stakeholders by tracking the source for all components

User Stories: 2693

Derived Requirements:

3447 The ASoT shall provide a means to associate a source with a model artifact as metadata.

3175 The ASoT supports all stakeholders by providing a searchable version and build for each software component

User Stories: 3117

Derived Requirements:

3488 The ASoT shall associate a version and build with digital artifacts representing software.

3487 The ASoT shall enable search across user-defined and user-selected metadata.

3174 The ASoT supports all stakeholders by providing a searchable criticality rating for each software and hardware component

User Stories: 3117, 3119

Derived Requirements:

3487 The ASoT shall enable search across user-defined and user-selected metadata.

3489 The ASoT shall associate a criticality rating with digital artifacts representing software.

3160 The ASoT supports all stakeholders by providing support to specify non-functional requirements that support trade-off analysis

User Stories: 2727, 2733, 2735, 2740

Derived Requirements:

3490 The ASoT shall associate non-functional requirements with digital artifacts as appropriate.

3486 The ASoT will automatically consider constraints during tradeoff analysis.

3153 The ASoT supports all stakeholders by providing a method to capture requirements and rationale

User Stories: 2608

Derived Requirements:

3491 The ASoT shall associate requirements and ratonale with digital artifacts as appropriate.

3.9. Oracle

3545 The ASoT supports all stakeholders by assisting in tracking delivery methods and requirements from CDRLs of the contract to help avoid inadvertent nullification of its negotiated rights

User Stories: 2412

Derived Requirements:

3546 The ASoT shall associate data rights with all artifacts.

3544 The ASoT supports all stakeholders by providing guidance for language and process for all terms and conditions via documentation and guidance for special clauses

User Stories: 2412

Derived Requirements:

3547 The ASoT shall associate with each artifact all requirements used to produce it.

3543 The ASoT supports all stakeholders by providing access to information useful in preparing SOW language

User Stories: 2408

Derived Requirements:

3547 The ASoT shall associate with each artifact the requirements used to produce it.

3542 The ASoT supports all stakeholders by providing all necessary documentation to assist in the development of requirements for the artifact deliverable

User Stories: 2406

Derived Requirements:

3547 The ASoT shall associate with each artifact the requirements used to produce it.

3541 The ASoT will support all stakeholders by providing marking information and licensing information on the Government-furnished artifact that was associated with any previous contract.

User Stories: 2404

Derived Requirements:

3549 The ASoT shall associate all marking and licensing information with an artifact, even from prior contracts.

3540 The ASoT supports all stakeholders by providing documentation for example data rights models to support portability, reuse, integration, maintenance, sustainment, and upgrade

User Stories: 2402, 2398

Derived Requirements:

3546 The ASoT shall associate data rights with all artifacts.

3550 The ASoT shall provide example data rights models with documentation.

3539 The ASoT supports all stakeholders by providing CDRLs for the contract containing data right markings (Could be stated in box 16 CDRL Form 1423) and required distribution statement (Box 9 and Box 16 of CDRL Form 1423).

User Stories: 2400

Derived Requirements:

3546 The ASoT shall associate data rights with all artifacts.

3537 The ASoT supports all stakeholders by providing specific references to the contract requirements used to produce digital artifacts

User Stories: 2396

Derived Requirements:

3547 The ASoT shall associate with each artifact the requirements used to produce it.

3401 The ASoT supports the program manager by providing contract based guidance

User Stories: 2396, 2412, 2408, 2406, 2404, 2402, 2400, 2398

Derived Requirements:

3547 The ASoT shall associate with each artifact the requirements used to produce it.

3549 The ASoT shall associate all marking and licensing information with an artifact, even from prior contracts.

3359 The ASoT supports all stakeholders by enabling the discovery and use of government template models

User Stories: 3354, 3355

Derived Requirements:

3551 The ASoT shall associate with each artifact any template models used to produce the artifact.

3552 The ASoT shall enable the discovery of available template models.

3168 The ASoT supports the government program manager by supporting the DoDAF model development process

User Stories: 2631

Derived Requirements:

3547 The ASoT shall associate with each artifact the requirements used to produce it.

3140 The ASoT supports all stakeholders by providing a central repository for prior market research data per FAR 10

User Stories: 2484

Derived Requirements:

3452 The ASoT shall provide a version control system for storing and managing digital artifacts.

3139 The ASoT supports all stakeholders by providing a central repository for standards documents

User Stories: 2480

Derived Requirements:

3452 The ASoT shall provide a version control system for storing and managing digital artifacts.

3451 The ASoT shall provide a uniform representation for stored standards documents.

3137 The ASoT supports the government contracting officer by confirming the most recent contract clauses

User Stories: 2475, 2476

Derived Requirements:

3547 The ASoT shall associate with each artifact the requirements used to produce it.

3134 The ASoT supports the government program manager by providing all solicitation guidance

User Stories: 2473

Derived Requirements:

3547 The ASoT shall associate with each artifact the requirements used to produce it.

3132 The ASoT supports the supplier by describing required capabilities, functions, systems standards, software, and hardware to support a proposal bid

User Stories: 2464

Derived Requirements:

3547 The ASoT shall associate with each artifact the requirements used to produce it.

3487 The ASoT shall enable search across user-defined and user-selected metadata.

3450 The ASoT shall provide an interface for searching stored deliverable materials.

3436 The ASoT shall associate metadata with digital artifacts to enable search and other functions.

3.10. Recompete

3545 The ASoT supports all stakeholders by assisting in tracking delivery methods and requirements from CDRLs of the contract to help avoid inadvertent nullification of its negotiated rights

User Stories: 2412

Derived Requirements:

3546 The ASoT shall associate data rights with all artifacts.

3544 The ASoT supports all stakeholders by providing guidance for language and process for all terms and conditions via documentation and guidance for special clauses

User Stories: 2412

Derived Requirements:

3547 The ASoT shall associate with each artifact the requirements used to produce it.

3543 The ASoT supports all stakeholders by providing access to information useful in preparing SOW language

User Stories: 2408

Derived Requirements:

3547 The ASoT shall associate with each artifact the requirements used to produce it.

3541 The ASoT will support all stakeholders by providing marking information and licensing information on the Government-furnished artifact that was associated with any previous contract.

User Stories: 2404

Derived Requirements:

3549 The ASoT shall associate all marking and licensing information with an artifact, even from prior contracts.

3540 The ASoT supports all stakeholders by providing documentation for example data rights models to support portability, reuse, integration, maintenance, sustainment, and upgrade

User Stories: 2402, 2398

Derived Requirements:

3546 The ASoT shall associate data rights with all artifacts.

3550 The ASoT shall provide example data rights models with documentation.

3539 The ASoT supports all stakeholders by providing CDRLs for the contract containing data right markings (Could be stated in box 16 CDRL Form 1423) and required distribution statement (Box 9 and Box 16 of CDRL Form 1423).

User Stories: 2400

Derived Requirements:

3546 The ASoT shall associate data rights with all artifacts.

3538 The ASoT supports all stakeholders by providing data rights models for both contractor and Government for use in the negotiation process

User Stories: 2398, 2402

Derived Requirements:

3550 The ASoT shall provide example data rights models with documentation.

3546 The ASoT shall associate data rights with all artifacts.

3537 The ASoT supports all stakeholders by providing specific references to the Contract requirements used to produce digital artifacts

User Stories: 2396

Derived Requirements:

3547 The ASoT shall associate with each artifact the requirements used to produce it.

3533 The ASoT supports all stakeholders by managing digital artifacts in a format that can be shared with other ASoTs.

User Stories: 3529, 3528

Derived Requirements:

3507 The ASoT shall provide an interface facilitating translation of artifacts into any format required by the standard modeling tools it supports.

3535 The ASoT shall provide a means to export its state in a standard format that other, authorized ASoTs can consume.

3536 The ASoT shall provide the means to import its state using a standard format.

3393 The ASoT supports all stakeholders by enabling them to explore the history associated with digital artifacts.

User Stories: 2597

Derived Requirements:

3452 The ASoT shall provide a version control system for storing and managing digital artifacts.

3457 The ASoT shall provide an interface to query known versions of a model artifact.

3385 The ASoT supports program managers by enabling reuse of certification digital artifacts across projects.

User Stories: 2647

Derived Requirements:

3463 The ASoT shall provide a means to associate one or more certifications with a specific version of a model artifact.

3384 The ASoT supports certification authorities by tracking which specific digital artifacts passed certification.

User Stories: 2669, 2647

Derived Requirements:

3463 The ASoT shall provide a means to associate one or more certifications with a specific version of a model artifact.

3132 The ASoT supports the supplier by describing required capabilities, functions, systems standards, software, and hardware to support a proposal bid

User Stories: 2464

Derived Requirements:

3547 The ASoT shall associate with each artifact the requirements used to produce it.

3487 The ASoT shall enable search across user-defined and user-selected metadata.

3450 The ASoT shall provide an interface for searching stored deliverable materials.

3436 The ASoT shall associate metadata with digital artifacts to enable search and other functions.

3.11. Resilient Operation

3399 The ASoT supports all stakeholders by implementing cyber resiliency controls to recover from data loss

User Stories: 2601, 2599

Derived Requirements:

3517 The ASoT shall use a journaling process for any changes to its own state.

3518 The ASoT shall have the capability to archive a snapshot of its state at a given time.

3398 The ASoT supports all stakeholders by implementing system integrity controls to monitor for integrity breaches

User Stories: 2601

Derived Requirements:

3458 The ASoT shall provide a means to sign a model artifact with a cryptographic checksum.

3520 The ASoT shall provide cryptographic verification of each change to a model artifact.

3370 The ASoT supports all stakeholders by ensuring continuity of operations, such as data mirroring through a backup data center

User Stories: 3332

Derived Requirements:

3517 The ASoT shall use a journaling process for any changes to its own state.

3518 The ASoT shall have the capability to archive a snapshot of its state at a given time.

3369 The ASoT supports all stakeholders by archiving off-site any digital artifacts identified by user-defined policy

User Stories: 3333

Derived Requirements:

3518 The ASoT shall have the capability to archive a snapshot of its state at a given time.

3452 The ASoT shall provide a version control system for storing and managing digital artifacts.

3178 The ASoT supports the ASoT administrator with a design that maximizes fast boot up

User Stories: 3108

Derived Requirements:

3553 The ASoT shall become fully operational from a cold boot within ZZZ seconds.

3176 The ASoT supports the ASoT infrastructure provider with a design that maximizes up time

User Stories: 2709, 3112

Derived Requirements:

3554 The ASoT shall recover from a system fault to a fully operational state within ZZZ seconds.

3161 The ASoT supports all stakeholders by providing a capability to stand up a new ASoT based on one or more existing ASoTs

User Stories: 2493, 2434, 2730

Derived Requirements:

3452 The ASoT shall provide a version control system for storing and managing digital artifacts.

3518 The ASoT shall have the capability to archive a snapshot of its state at a given time.

3517 The ASoT shall use a journaling process for any changes to its own state.

3519 The ASoT shall provide an interface to create a certified copy of its store of artifacts.

3.12. Traceability

3533 The ASoT supports all stakeholders by managing digital artifacts in a format that can be shared with other ASoTs.

User Stories: 3529, 3528

Derived Requirements:

3507 The ASoT shall provide an interface facilitating translation of artifacts into any format required by the standard modeling tools it supports.

3535 The ASoT shall provide a means to export its state in a standard format that other, authorized ASoTs can consume.

3536 The ASoT shall provide the means to import its state using a standard format.

3404 The ASoT supports all stakeholders by traceability to the versions of the analysis tools that generated artifacts

User Stories: 2531, 2493

Derived Requirements:

3459 The ASoT shall provide a means to associate a set of analysis results with specific versions of analysis tools.

3460 The ASoT shall maintain a registry of approved modeling and analysis tools, supporting different versions thereof.

3393 The ASoT supports all stakeholders by enabling them to explore the history associated with digital artifacts.

User Stories: 2597

Derived Requirements:

3452 The ASoT shall provide a version control system for storing and managing digital artifacts.

3457 The ASoT shall provide an interface to query known versions of a model artifact.

3387 The ASoT supports system engineers by enabling inclusion of legacy system digital artifacts

User Stories: 2642

Derived Requirements:

3492 That ASoT shall manage digital artifacts associated with legacy systems.

3386 The ASoT supports program managers by enabling traceability between ASoTs

User Stories: 2640, 2644

Derived Requirements:

3424 The ASoT shall provide a means to designate associations between different artifacts.

3423 The ASoT shall provide a means to assign a globally unique identifier to each artifact.

3438 The ASoT shall provide the ability to access digital artifacts on other ASoTs.

3384 The ASoT supports certification authorities by tracking which specific digital artifacts passed certification.

User Stories: 2669, 2647

Derived Requirements:

3463 The ASoT shall provide a means to associate one or more certifications with a specific version of a model artifact.

3375 The ASoT supports reviewers (stakeholder role) by including a mechanism for tracking the digital artifacts under review (business process).

User Stories: 2667

Derived Requirements:

3495 The ASoT shall provide a means to track digital artifacts under review.

3358 The ASoT supports all stakeholders by providing storage and tracing of government template models and performer submission models

User Stories: 3355

Derived Requirements:

3493 The ASoT shall manage government template models as digital artifacts.

3494 The ASoT shall manage performer-submitted models as digital artifacts.

3351 The ASoT supports all stakeholders by tracking the owner of artifacts and notifying the owner of automated analysis results that impact the artifact

User Stories: 3269

Derived Requirements:

3496 The ASoT shall notify the owner of digital artifacts of changes in automated analysis results for those artifacts.

3345 The ASoT supports all stakeholders by providing the capability to track artifacts for future changes

User Stories: 3277

Derived Requirements:

3455 The ASoT shall provide hooks allowing the storage of a new version of a model artifact to trigger actions.

3457 The ASoT shall provide an interface to query known versions of a model artifact.

3452 The ASoT shall provide a version control system for storing and managing digital artifacts.

3165 The ASoT supports all stakeholders by tracking the evolution of functional and performance specifications

User Stories: 2632

Derived Requirements:

3463 The ASoT shall provide a means to associate one or more certifications with a specific version of a model artifact.

3456 The ASoT shall provide an interface to compare different versions of a model artifact.

3454 The ASoT shall provide a means to associate a set of analysis results, or a portion thereof, with specific versions of model artifacts.

3164 The ASoT supports the government approval authority by providing tracing from more abstract to more detailed analyses

User Stories: 2633

Derived Requirements:

3454 The ASoT shall provide a means to associate a set of analysis results, or a portion thereof, with specific versions of model artifacts.

3456 The ASoT shall provide an interface to compare different versions of a model artifact.

3483 The ASoT will provide a method to compare the analysis results of selected product variants.

3156 The ASoT supports all stakeholders by providing support to discover related model artifacts through traceability links

User Stories: 2640, 2419, 2546, 3005, 3340, 3051, 3044, 3043, 3042

Derived Requirements:

3424 The ASoT shall provide a means to designate associations between different artifacts.

3423 The ASoT shall provide a means to assign a globally unique identifier to each artifact.

3426 The ASoT shall provide a means to discover model artifacts using traceability link.

3155 The ASoT supports all stakeholders by providing support to capture traceability between related but distinct modelling languages (AADL, CAD, SysML, etc.)

User Stories: 2640, 2537, 2497, 2545, 2548, 3340, 3048, 3047

Derived Requirements:

3424 The ASoT shall provide a means to designate associations between different artifacts.

3426 The ASoT shall provide a means to discover model artifacts using traceability link.

3425 The ASoT shall provide a means to visualize associations among model artifacts.

3423 The ASoT shall provide a means to assign a globally unique identifier to each artifact.

3.13. Tradeoff Analysis

3396 The ASoT supports all stakeholders by mapping feature sets to the different environments those features support to facilitate dependency analysis and tradeoff analysis

User Stories: 2641

Derived Requirements:

3423 The ASoT shall provide a means to assign a globally unique identifier to each artifact.

3424 The ASoT shall provide a means to designate associations between different artifacts.

3425 The ASoT shall provide a means to visualize associations among model artifacts.

3426 The ASoT shall provide a means to discover model artifacts using traceability link.

3484 The ASoT will manage feature sets as digital artifacts.

3388 The ASoT supports multiple system stakeholders by enabling product variants.

User Stories: 3044, 3045

Derived Requirements:

3481 The ASoT will manage product variants.

3378 The ASoT enables all stakeholders to reduce effort required to conduct a trade study by automatically including constraints in the trade space.

User Stories: 2665

Derived Requirements:

3486 The ASoT will automatically consider constraints during tradeoff analysis.

3485 The ASoT will manage constraints as digital artifacts.

3377 The ASoT supports all stakeholders by providing a capability to explore trade spaces.

User Stories: 2498, 2665

Derived Requirements:

3483 The ASoT will provide a method to compare the analysis results of selected product variants.

3158 The ASoT supports the system engineer by providing support to instrument model variations to support trade space analysis

User Stories: 2576

Derived Requirements:

3482 The ASoT will provide a method to select product variants for analysis.

3157 The ASoT supports all stakeholders by providing support to identify model variation points, calculate change impact, and trigger execution of automated analyses

User Stories: 2547

Derived Requirements:

3482 The ASoT will provide a method to select product variants for analysis.

3481 The ASoT will manage product variants.

3479 The ASoT shall evaluate the certification impact given the results from certification-related analyses.

3455 The ASoT shall provide hooks allowing the storage of a new version of a model artifact to trigger actions.

3149 The ASoT supports the government program manager by providing support for Multi-Disciplinary Analysis and Optimization (MDAO) across assets

User Stories: 2612

Derived Requirements:

3556 The ASoT shall enable the programmatic execution of user-defined analysis over digital artifacts.

3008 The ASoT supports all stakeholders by providing analysis of multiple design variations

User Stories: 2576, 2612

Derived Requirements:

3483 The ASoT will provide a method to compare the analysis results of selected product variants.

3482 The ASoT will provide a method to select product variants for analysis.

3162 The ASoT supports all stakeholders by providing a view of the logistical supply chain to support country-of-origin tradeoffs

User Stories: 2728

Derived Requirements:

3447 The ASoT shall provide a means to associate a source with a model artifact as metadata.

3.14. Workflows

3380 The ASoT supports all stakeholders by providing configurable workflows.

User Stories: 2613

Derived Requirements:

3522 The ASoT shall provide programmatic access to its artifacts and user interfaces.

3524 The ASoT shall provide a means of scripting actions exposed via its programmatic interface.

3362 The ASoT supports all stakeholders by supporting centralized scheduling tools with automatic notification

User Stories: 3341

Derived Requirements:

3523 The ASoT shall provide hooks for transmitting notifications on specified triggers.

3502 The ASoT shall provide hooks to schedule automatic execution of user-defined actions on specified triggers.

3503 The ASoT shall provide hooks to trigger actions automatically at specified times or intervals.

3352 The ASoT supports all stakeholders by supporting automatic analysis at user-defined intervals with notification support

User Stories: 3267

Derived Requirements:

3522 The ASoT shall provide programmatic access to its artifacts and user interfaces.

3523 The ASoT shall provide hooks for transmitting notifications on specified triggers.

3503 The ASoT shall provide hooks to trigger actions automatically at specified times or intervals.

3502 The ASoT shall provide hooks to schedule automatic execution of user-defined actions on specified triggers.

3348 The ASoT supports all stakeholders by providing user-defined review processes, feedback collection, and approval sign-offs

User Stories: 3268, 3272

Derived Requirements:

3469 The ASoT shall collect and store review comments on digital artifacts.

3495 The ASoT shall provide a means to track digital artifacts under review.

3148 The ASoT supports all stakeholders by providing an interface to workflow automation capabilities

User Stories: 2613

Derived Requirements:

3522 The ASoT shall provide programmatic access to its artifacts and user interfaces.

3524 The ASoT shall provide a means of scripting actions exposed via its programmatic interface.

3128 The ASoT supports the system developer by facilitating government review and comment on deliverables

User Stories: 2456

Derived Requirements:

3469 The ASoT shall collect and store review comments on digital artifacts.

4. All User Stories

2469 A Government program manager and supporting contract management staff want to monitor a contractor's performance on a contract. They want to be able to capture information about expenditures against contract value and funded value, invoiced amounts, deliverable status on a dashboard. Why: This will create consistency in format of this information and ease of viewing this information quickly without going through reports or other various systems to obtain pieces of this information. How: ASoT could provide standardization of this information via contractor access to input the information and upload deliverables. Examples of such systems are DARPA's TFIMS and NASA's Electronic Handbook. These systems contain final executed contracts and CDRLS and provide deliverable and financial information provided by the contractor as part of performance of the contract giving the program manager a dashboard approach to viewing a quick status of a program.

DEVCOM Priority:

DEVCOM Comments: (per Dale Johnson) The dashboard information potentially available to the government program manager and contracting office coould be defined in terms of Earned Value Management variables to include Budgeted Cost of Work Scheduled (BCWS), Budgeted Cost of Work Performed (BCWP), Actual Cost of Work Performed (ACWP), Budget at Completion (BAC), and Estimate at Completion (EAC). Recommend adding language to that effect.

2396 A Program Manager wants to write a solicitation in which some system components and software meet FACE Technical Standards. Specifically, they want to know what deliverables to request and any special wording required. Why: In consideration of protecting the goal of the Government to have severable components and software developed which would protect the rights of the Government to recompete through different vendors through the life cycle, it is more efficient for the program manager to be able to look at a model that describes data right issues to consider, the CDRLs to request, the level (Units of Conformance) at which to request the CDRLS, and wording for the CDRLS which will help protect the Government's program goals and data rights interest. How: ASoT will provide guidance through work that has already been completed such as references to the FACE Contract Guide which would be helpful in this scenario. Even if the requirement did not include FACE standard requirements, the guidance in the FACE Contract Guide provides information relevant to severable components and software with regard to deliverables and data right issues.

DEVCOM Priority:

DEVCOM Comments: (via Marcell Padilla) Use case is too specific. ASOT needs to provide this capability for all requirements. The use case is just an example of FACE requirements. More specifically, ASOT doesn't provide this, the FACE contract guide provides this. (via Alex Boydston) Need to add certification and qualification artifacts in this scenario.

3344 A component supplier uses a desktop-based (thick client) analysis tool to perform computational fluid dynamics evaluations. The ASoT provider wishes to perform this analysis as part of the automated architecture evaluation. The ASoT provider acquires a server-based version of the tool and adds it to the automated evaluation workflow performed by the ASoT.

DEVCOM Priority:

DEVCOM Comments:

2537 A component vendor wishes to evaluate a component solid design in the context of a mission system integrator's larger chassis model. The MSI provides a URI referencing the location in the chassis model where the new component should be placed. The Vendor uses the URI to view the chassis model in the ASoT. The Vendor is able to evaluate their design against the MSI model to determine whether it will fit.

DEVCOM Priority:

DEVCOM Comments: (per Alex Boydston) Being able to understand the allotment of volumetric space and weight for nodal points would be good add.

2456 A contract is set up for payments to be made on milestones such as reviews and interim design 'documents', which may include delivery of interim models and the opportunity for the Government to provide feedback, comments, changes, etc. Related to this, the Government comments will need to be incorporated in a way that doesn't stop all forward progress on the project. During the 30 days the gov't typically has to review things, like final reports, the teams cannot just be sitting around. Why: If the deliverables are reduced to some other form, like powerpoint, word doc, etc., then there is a separation between the ASoT and the information being provided, plus it will take more time. How: The artifacts used in these reviews should be drawn directly from the ASoT and the Government comments back should be provided via ASoT too. Ideally, the comments are provided and recorded during the review session itself so there is no disruption in schedule waiting for acceptance. This may require homework on the part of the review attendees ahead of the review, the ASoT should support this. A record of the delivery and acceptance of the delivery will be needed for contract tracking and possibly payment. The proof of delivery needs to be compatible with the regular Gov't processes for milestone payments and delivery tracking.

DEVCOM Priority:

DEVCOM Comments:

2402 A contracting officer along with Government legal counsel want to develop a data rights strategy with regard to deliverables and artifacts when negotiating a contract with regard to software to ensure that the software and associated documentation will be reusable and portable for future use and be able to be used in future solicitations and programs. Why: Assurance that the Government will retain the appropriate license rights in technical data, such as interface documentation, required for future changes, including replacement, upgrade and integration of Units of Conformity and Software units. They will want to obtain the appropriate rights to support reuse and portability of software. How: ASoT provides documentation for example data rights

models to support portability and reuse. Artifacts, such as the FACE Contract Guide and in DoD OSA (Open Systems Architecture) Contract Guidebook, provide Data Right guidance to open system procurements along with commentary for consideration.

DEVCOM Priority: Low Priority

DEVCOM Comments: Cary Pool: Critical Priority Cary Pool: Contractual wording that forces delivery of verifiable portability and reuse is critical to success. (via Marcell Padilla) Use case is too specific. The ASoT does not provide the appropriate rights, the references do. ASoT needs to provide access to appropriate references, this set is too limiting.

2475 A contracting officer and contracting specialist want to confirm that the most up to date and appropriate FAR, DFAR and special contract clauses are incorporated into a draft contract for the type of contract that is applicable. Why: Not only do clauses get revised and have dates associated with the clause, but there are special clauses that should be incorporated to protect the government's interest.

DEVCOM Priority:

DEVCOM Comments:

2398 A contractor is responding to a solicitation for which they are bringing their current technology to the program. The program tasks performers with developing tools which meet the goal of severability of components and software while providing at minimal Government Purpose Rights for those items. The contractor also wants to understand how to prepare Architecture Date Rights Strategy that would protect rights in existing software and data but also meet the data rights goals of the Government in newly developed work. Why: There is significant research time that goes into defining and understanding the data rights models for the various components. If provided models for an Architecture Data Rights Strategy, the contractor could save time and money in the proposal and negotiation process preparing this information. Also, it would provide efficiency in the negotiation process on the Government side as well as a guide for the acquisition agency. How: ASoT provides data rights models for both contractor and Government for use in the negotiation process saving time and money for the contractor and provide the Government with a basis for negotiation.

DEVCOM Priority:

DEVCOM Comments:

2460 A large system development that encompasses the breadth envisioned for ASoT will have dozens, perhaps hundreds, of participating organizations. Even if the ASoT enables protected information to be hidden, there may be a need for additional agreements between the participants, e.g., to share SBIR rights data with non-Gov't performers. Getting an N-wise NDA or an associated contractors agreement (ACA) in place for all of these participants would be nearly impossible. What's more likely is that groups of NDAs or ACAs will be required. Why: The approaches to NDAs and ACAs is as varied as there are performers and some may not sign agreements with each other. How: For information that cannot simply be kept private, the ASoT should provide a way to a) bound the organizations that need to get an NDA, e.g., determine which participants touch or see the aforementioned SBIR data? can it be seen beyond them? and b) enforce the agreements that are in place, keeping in mind that participants may be merged or bought or changed for some other reason.

DEVCOM Priority:

DEVCOM Comments: Cary Pool: Medium Cary Pool: "Not sure the NDAs and system permissions should be directly linked together in the ASoT. Normally NDAs are handled through

legal groups through documentation, but ASoT would be an IT system with access rights controled by DD2875 System Access Requests or some other documenation clearly showing system access rights approved by multiple parties. It would be important that ASoT have the ability to document DD2875 SAR forms/signatures and enable admins to apply rights to match the SAR. It would also be important that gov't and IA personell be able to pull a report to audit the access rights and produce that report for any disputes. " (per Alex Boydston) Disagree with Cary's comments about NDAs not being linked together. NDAs are important wrt data rights.

3269 A mission system integrator stakeholder is tracked as the owner of a collection of model artifacts. Automated analysis of the model artifacts detects a latency error. The ASoT emails the MSI to alert them to the error and provides a reference to the error report.

DEVCOM Priority: High Priority

DEVCOM Comments: From Cary Pool: The email is low priority. The automated timeing analysis is high value.

2539 A mission system integrator wants to display analysis results in their internal status dashboard. The ASoT provides Respresentational State Transfer (REST) endpoints that provide listings of available information (e.g., models, test results, documents) along with formats available for each piece of information (e.g., models in AADL, analysis results in XML). The MSI configures their dashboard to poll the ASoT rest interface for information.

DEVCOM Priority: Medium Priority

DEVCOM Comments: Cary Pool: High Priority Cary Pool: Any REST API available must be secured to FedRAMP and NIST 800-53 standards and approved by an ATO. Any automation of verify and validation process and results would greatly speed testing and thus competition.

3357 A model is split between classified and unclassified domains. When working on the classified version, the engineer must use a high-side workstation. When working on the unclassified model, the engineer must use a low-side workstation. The ASoT provides a transfer facility to propagate changes made in the unclassified model into the classified model.

DEVCOM Priority:

DEVCOM Comments:

2473 A new program manager is tasked with procuring a system with the intent that the system meet MOSA requirements. The program manager wants to analyze, in a broad sense, the things that should be considered in both writing the solicitation and managing the program. Why: the program manager wants to be informed of the issues to consider and does not want to reinvent the wheel. How: ASoT will house documents created to provide guidance to Program Managers when writing a MOSA solicitation and managing a MOSA contract, such as DoD OSA (Open Systems Architecture) Guidebook, and FACE Contract Guide.

DEVCOM Priority:

DEVCOM Comments:

2627 A process owner wishes to define a new process for certification of airframe modifications. The process owner’s ASoT user has been granted the “create process” permission for the current project, so the ASoT allows him or her to create a new process. The owner creates a new process using the ASoT web interface, providing descriptions of the steps in the process as well as requirements for completing steps (e.g., required artifacts to upload or digital signatures of authorities). For each step of the process, the process owner indicates which users, stakeholders, or organizations can or must participate in the process.

DEVCOM Priority: Low Priority

DEVCOM Comments:

2620 A product developer needs to share source code with a mission system integrator. The mission system integrator uses a different compiler than the product developer. The product developer provides the source code with data rights that prevent modification or re-use of the source code by the mission system integrator. The ASoT enforces read-only access according to access rules specified by the product developer.

DEVCOM Priority:

DEVCOM Comments:

2626 A product owner wants to monitor the progress of a system modification through airworthiness review. The airworthiness review requires digital sign-off by authorities in several organizations, AAA, BBB, and CCC. The ASoT is configured with an airworthiness process that includes references to the organizations or stakeholders whose signoff is required at specific points in the process. As the review progresses, each stakeholder whose signoff is required is notified when it is their turn to sign-off, and is given reminder notifications for items that may have been forgotten in their queue. The progress through the signoff process is displayed on a dashboard for the product owner.

DEVCOM Priority:

DEVCOM Comments: (per Alex Boydston) This staged steps for Airworthiness release is also applicable foor overarching Materiel Release per Army Reg 700-142.

2400 A program manager wants to verify that a deliverable from the contract is provided with the appropriate Distribution Statement and Data Rights Markings as originally negotiated and stated in the CDRLs. Why: Before formal acceptance of a deliverable, a program manager wants to verify not only technical adherence to the requirements of the deliverable, but that it adheres to the terms of the negotiated contract from a Data Rights and Dissemination stand point. The Government must be vigilant in identifying and challenging any restrictive markings on deliverables that are inconsistent with the rights the Government has negotiated on the contract. How: ASoT Provides CDRLs for the contract containing data right markings (Could be stated in box 16 CDRL Form 1423) and required distribution statement (Box 9 and Box 16 of CDRL Form 1423). The Government should compare data language on code headers on software with the negotiated rights and distribution statements of the contract. (Note: sample software header language is provided in DoD OSA (Open Systems Architecture) Contract Guidebook, Appendix 4.

DEVCOM Priority:

DEVCOM Comments:

2599 A regional power outage disrupts portions of the cloud hosting the ASoT. The cloud provider restores the computing assets. The Government queries the ASoT to restore and recover, and the ASoT responds with the last stable state before the outage. Why: While it is not a real-time resource, the ASoT must try to recover with a minimum loss of state. How: The ASoT implements cyber resiliency mechanisms, such as redundancy and fail-over, to maintain a stable state.

DEVCOM Priority:

DEVCOM Comments:

3277 A safety reviewer is particularly concerned with the configuration of the harnesses in a vehicle design. He flags the harness modeled object in a CAD model as "to be tracked" by the ASoT. The ASoT models that harness for changes. If an updated digital artifact describing the harness is uploaded, the ASoT automatically emails the reviewer to notify him of the change.

DEVCOM Priority: Medium Priority

DEVCOM Comments:

2458 A small business is providing models for a system component that has SBIR rights (developed on a SBIR, including Phase III SBIR). Assuming the work being done on the ASoT is funded by the Gov't, the work on the new system will also have SBIR rights. The Gov't cannot force a small business to relinquish those rights. Why: SBIR rights are a special category of data rights, which are basically like proprietary data of the company. How: ASoT should offer the same ability for a small biz to hide it's more detailed parts of the models and also provide a way to mark even the 'public' facing parts as having SBIR rights with the detailed legend available (ideally autogenerated).

DEVCOM Priority:

DEVCOM Comments:

3276 A supplier is unable to add a new digital artifact to the ASoT. The ASoT provides an issue tracking system for users to report issues with the ASoT itself. The supplier submits an issue report via the issue tracking system.

DEVCOM Priority:

DEVCOM Comments:

3270 A vendor uploads a new version of a CAD model. The ASoT detects creation of new digital artifacts and automatically triggers re-execution of analysis.

DEVCOM Priority:

DEVCOM Comments:

2547 After procuring design models from contractor A, the government wishes contractor B to use an automated analysis tool to evaluate the impact to airworthiness verification of changing the material used to manufacture the landing gear from titanium to aluminum. Contractor B connects to the ASoT and uses the ASoT API to identify Variation Points in design models related to material selection. Contractor B's tool creates a new product variant in the ASoT that references Contractor A's design, then changes the material selection for the landing gear feature. Contractor B's analysis tool traverses all traceability links from the changed variation points to calculate a change impact score and triggers an execution of all available automated tests by the ASoT.

DEVCOM Priority:

DEVCOM Comments:

3114

DEVCOM Priority:

DEVCOM Comments:

3115 Airworthiness Qualification User Stories, v.0

DEVCOM Priority:

DEVCOM Comments:

2448 An ASoT for a real system will have information at a variety of classification levels, a small business (or other organization) is selected to provide some aspect of the system, but does not have the proper clearances and facilities. Putting those facilities in place could be time and cost prohibitive, if even possible (e.g., an academic or foreign national participant). ASoT should allow participants ranging from unclassified on up. Why: Requiring everyone to have the highest clearance level will stifle broad participation, erect barriers for non-traditional organizations to participate, add overhead, and over classify the information in the ASoT. How: ASoT should enforce that users can only access items for which they have clearance and need to know. For example, a provider of the power distribution system should not be able to see any details of, say, the mission computer, even if the former has a TS and the latter is unclassified. ASoT needs to also consider collateral classified, it may not be enough to simply screen on the classification of a particular artifact, may need to consider an integration of artifacts.

DEVCOM Priority:

DEVCOM Comments:

2452 An ASoT participant has a subcontractor providing portions of the participant's solution. The participant wants to limit the subcontractor's access to the participant's overall model and solution and the participant may want to hide the participation of the subcontractor from other participants using ASoT. Why: This could be for proprietary (specifics of participants role and solution), competitive (more than one sub competing for the follow-on role), or security reasons (participant is not cleared at the appropriate level). How: Establish some sort of prime-sub bubble that lets the sub participate to the extent required and that's all.

DEVCOM Priority:

DEVCOM Comments:

2446 An ASoT participant is developing models for a portion of the system and executing nightly virtual integration runs, possibly using models from other participants. The models are part of an ongoing competition with a downselect review coming up (note: competition could be for gov't, of course, but also for a prime). The participant will want to be assured that the results of their intermediate tests and, obviously, the models themselves are not available to its competitors and the decision makers in the competition. Why: For competitive reasons, the works in progress should be kept to the participant and those on its team that the participant wished to have access. In addition, to avoid a bid protest, the selection committee will want to be assured they have no access until the review version of the model is delivered by the participant – no need to see the sausage being made. How: Need to Know determination should incorporate teaming relationships and allow individual versions to have different Need to Know scope (works in progress would be tighter, delivered versions broader). The review version access could also be ephemeral, switched on just for the gov't feedback session and then off again afterwards.

DEVCOM Priority: Medium Priority

DEVCOM Comments: Cary Pool: Agree the feature would be useful however, don't agree that the selection committee would not want to have access. For example in the FARA CP we have access to draft deliverables long before final delivery. This enables pre-review and feedback to ensure final delivery is understood by reviewers.

3268 An airworthiness authority ASoT user signs into the ASoT dashboard and sees a review request. She opens the request and completes the review, adding comments but indicating approval. Her

digital signature is added to the review record and the review is automatically assigned to the next stakeholder in the airworthiness review business process.

DEVCOM Priority:

DEVCOM Comments:

3335 An auditor connects to the ASoT to inspect digital artifacts uploaded by contractors. The auditor checks the access permissions on artifacts to ensure that the ASoT is properly restricting access to controlled data.

DEVCOM Priority:

DEVCOM Comments:

3394 An engineer is editing a model that contains classified and unclassified information. The unclassified features of the model need to be shared on the low side. The ASoT keeps the unclassified and classified models synchronized without exposing the classified information. The ASoT reduces duplicate and error prone effort by the engineer that would otherwise occur via manually splitting and maintaining the separate portions of the models. The ASoT provides a secure guard capability that can recognize and control model information.

DEVCOM Priority:

DEVCOM Comments:

3360 An engineering firm relies on a .face model that is updated from several sources managed by other stakeholders. Each time one of the sources is updated, the ASoT notifies the engineering firm that an upstream resource has changed. The stakeholder uses the ASOT's change tracking and merging capability to merge the change into the baseline .face file.

DEVCOM Priority:

DEVCOM Comments:

2540 An engineering manager, contracts specialist, SysML modeler, and solid modeler all need to collaborate on common artifacts (e.g., schedule) but have different priorities. To accommodate these different viewpoints, the ASoT interface provides customizable views that are tailored to individual user roles. Upon login, the contracts specialist sees a status dashboard that highlights contracts deadlines, milestones, and priorities. The SysML modeler sees pending tasks, recent model changes, etc.

DEVCOM Priority:

DEVCOM Comments:

2601 An external attacker launches a successful ransomware attack against the cloud hosting the ASoT. Detecting the attack, the ASoT automatically blocks access to all stakeholders. The Government queries the ASoT to restore and recover, and the ASoT responds with the last stable state before the attack. Why: The ASoT must monitor its own integrity and warn stakeholders immediately of integrity breaches. The ASoT must enable the Government to recover lost data. How: The ASoT implements system integrity controls to monitor for integrity breaches. The ASoT implements cyber resiliency controls to recover from data loss.

DEVCOM Priority:

DEVCOM Comments:

3047 Army Simulation Directorate wants to integrate multiple system simulations to evaluate battlefield performance in a particular scenario. These simulations would be from each weapon system involved in the common DoD simulation format. The ASoT should support the integration of the simulations into a large simulation environment with relatively high fidelity (estimate kill ratios).

DEVCOM Priority:

DEVCOM Comments:

2464 As a competing potential supplier for a mission system for an FVL aircraft, I want to identify in the government solicitation what software and hardware I am to supply in my B-Kit deliverables; and what layers of network protocols and what information flows to and from the mission system are to be supported by my B-Kit. I want to know what Isolation/Guard capabilities and functions are allocated to the mission system and are to be provided by software and hardware that I am to supply. I am concerned there will be ambiguity in this area because the Digital Backbone GFI says that both mission system and air vehicle will provide B-Kits to support the exchange of information between them; and Isolation/Guard capabilities are to be provided somewhere; but leaves open the question of exactly what is to be provided by the mission system and what by the air vehicle system. I want to query the ASoT during the solicitation phase to determine what capabilities, functions, open systems standards, software, and hardware I need to include in my proposal for a mission system. A successful query will return sufficient information to make a reasonable cost estimate with minimal time spent querying (the time available during a proposal effort).

DEVCOM Priority:

DEVCOM Comments:

2611 As a government Program Office, I am developing a system that must operate in a joint services system-of-systems. The other services and their PEOs and POs have their own ASoT infrastructure. I need a flexible interface between my ASoT and other service ASoTs, some of which are likely to be more primitive than my really cool one. According to the DoD SoS guidelines, much of this will involve coalitions of the willing rather than there existing any single organization trying to manage the overall system-of-systems. That is, my ASoT should be usable in an environment where there is no centralized management of the overall environment into which my system must integrate.

DEVCOM Priority: High Priority

DEVCOM Comments: Cary Pool: Irrelevant Cary Pool: Don't understand requirement as it appears to ask that you develop features to interface with an unorganized ASoT that is different than yours. Believe this is vague and should be re-scoped to just provide interaces that would enable another ASoT to integrate on their own. Also believe interface/api's are covered in a later requirement already.

2498 As a government or supplier systems engineer, I want to do trade studies to assess various architecture alternatives that involve a number of Measures of Performance and Quality Attributes that directly relate to Measures of Effectiveness (MoEs say how well missions are accomplished using that system). I can obtain these MoPs and QAs that by analyzing models. I need to be able to configure a model for the different points in the design space that I want to evaluate. I need to analyze different models for the different MoPs and QAs, e.g., some from a solid model, some from a CFD model, some from a dynamical systems model, some from a computer software and systems architecture model. Sometimes, the information needed to configure a model has to be obtained by analyzing an earlier model (e.g., stability and control derivatives used in a dynamical model of the air vehicle are obtained from a CFD analysis of vehicle surfaces that are found in

the solid model). I then want to allow multiple stakeholders to visualize the results in a variety of ways so they can make trade-off decisions. (Examples of current tools to do this are the NDAC helicopter trade studies tool, a domain-specific tool that integrates a variety of different modeling methods behind a single user interface, {LINK title="https://rotorcraft.arc.nasa.gov/ndarc/" uri=https://rotorcraft.arc.nasa.gov/ndarc/}; and the ModelCenter black-box model integration framework from Phoenix Integration, {LINK title="https://www.phoenix-int.com/product/modelcenter-integrate/" uri=https://www.phoenix-int.com/product/modelcenter-integrate/} and the 3DS (Dassault Systems) iSight tools. A number of such frameworks use the Trade Space Visualizer to decide on design points to sample and visualize results, {LINK title="http://www.atsv.psu.edu/whatisatsv.html" uri=http://www.atsv.psu.edu/whatisatsv.html}. Stakeholders want to visualize trade-offs between system MoPs such as range, payload capacity, and cruise speed; and not worry about engineering design decisions made to satisfy their choices, such as specific suite of sensors or electronics packaging. I would also like to be able to integrate design optimizers so these details could be automatically handled during trade studies, such as the OpenMDAO tool, {LINK title="https://openmdao.org/" uri=https://openmdao.org/} . Success is measured by the ability of end users (e.g., mission planners) to explore the trade space of Measures of Effectiveness, where analyses and detailed design decisions are largely automated.

DEVCOM Priority:

DEVCOM Comments: (per Alex Boydston) This example is a good one for tradespace analysis.

2632 As a government stakeholder who has expertise in the missions that we want the system to perform; a stakeholder who was involved in creating the DoDAF Operational Views and Capability Views; I want to track the evolution of the functional and performance specifications as they are detailed by the suppliers. As that detail emerges, we (the potential users) will want to modify our original capabilities specifications in order to make sure our intentions are unambiguous and complete. Changes will have to go through a Change Control Board, and we want to use the ASoT to assess the impact of changes we would like to make as part of those CCB reviews. Success would be that we can clarify ambiguities, fill gaps, or make corrections in the required capabilities information earlier than before because the ASoT supports more rapid impacts assessment and more agile tracking and collaboration with the suppliers.

DEVCOM Priority:

DEVCOM Comments:

2631 As a government stakeholder, I must collaborate with multiple other government stakeholders in the development of a DoDAF enterprise model for my system. DoDAF models are verified and validated largely by human review and are used to inform a variety of stakeholders. I want the ASoT to support the 6 step DoDAF model development process ({LINK title="https://dodcio.defense.gov/Library/DoD-Architecture-Framework/dodaf20_arch_development/" uri=https://dodcio.defense.gov/Library/DoD-Architecture-Framework/dodaf20_arch_development/}). DoDAF specifies a meta-model and information content but does not specify a single representation; the OMG UPDM Profile, an extension of the SysML profile, is an example of one commonly used option. DoDAF does not specify a single presentation technique; different presentations should be created to aid decisions by different stakeholders (https://dodcio.defense.gov/Library/DoD-Architecture-Framework/dodaf20_presentation/). I want the ASoT to support a collaborative process in which multiple stakeholders from multiple government organizations review, comment, and update DoDAF models. I want to be able to add plugins or tools to help with traceability and consistency management and assurance between all the different views. I want to be able to add plugins to create different presentation views for different stakeholders. Success is measured by the reduced time and effort needed to produce a model with fewer inconsistencies, gaps, and ambiguities that better represents the desires of all the stakeholders.

DEVCOM Priority:

DEVCOM Comments:

2493 As a member of the team managing the ASoT for the government, I must support upgrades to the ASoT itself during its lifecycle. This will sometimes mean replacing a COTS tool from one supplier with a competing tool from another supplier. In addition to the ASoT implementation having a MOSA, the data architecture – the model of models – must also be structured to support this. I want to query the ASoT to determine what information is accessed by a subsystem that is being replaced. The result should identify the inputs and formats as necessary to carry out automated export, conversion, and import during the switch-over from the old to the new. I want to query to determine which updates to information are needed to maintain consistency and referential integrity throughout the ASoT. Success means upgrades of ASoT tools can be performed with sufficiently low cost and effort that the government can benefit from the latest ASoT technologies and maintain a competitive market for ASoT tool suppliers.

DEVCOM Priority:

DEVCOM Comments:

2495 As a member of the team that is doing a study of potential upgrades to a system. This information is initially captured as a set of alternative changes to the capability and functional models, such as a modified set of DoDAF Capabilities Views. We are to estimate the cost and schedule to implement various combinations of new capabilities. To do this, we must query the ASoT to do a change impact analysis. In this user story, for one upgrade we will need to add new sensors at surface air vehicle nodal points, a new software application to the mission system processor, and add new overlay information to existing cockpit displays. Examples are queries to see what desired input data is available from existing sources and what output data needs to go to applications managing the displays to be overlaid; whether nodal point and conduit space is available to hold sensors and cabling; whether processing and network capacity exists; impact of additional weight and power on vehicle performance characteristics such as ceiling, rate-of-climb, range, etc.; and where not, what changes to existing software and equipment are needed to support adding the new capability. These queries span multiple models and involve multiple kinds of relationships between models. Success means our studies result in estimated cost and schedule that are acceptably accurate.

DEVCOM Priority:

DEVCOM Comments:

2494 As a supplier, I collaborate with other suppliers via the ASoT. I download information I need and upload information that others need. My internal development environment uses different tools and models than the ASoT does. Many of my models contain very detailed and proprietary information, and what I need to upload is only a selected portion of the information in my development environment. I need an interface to the ASoT that allows me to establish relationships between information visible to me in the ASoT and information I have in my internal development environment that I am willing to upload subject to conditions on access and data rights. The interface must be stable and use common formats so that I can leverage existing software with minimal internal development in order to perform complex import, export, and conversion of information. I need to understand how the ASoT manages relationships between different information assets so that relationships to and from my uploads and other ASoT information are correctly recorded. Success means that the supplier investment (much of which gets passed on to the government) to interface with the ASoT is minimal and the manual effort required to do an import or export is minimal (largely automated).

DEVCOM Priority:

DEVCOM Comments:

2608 As a supplier, I want to engage my customer very early in the process using simulations and early prototypes in order to validate HMI requirements. I expect this to require a lot of equipment and development tools, and modeling and simulation formats, that are not supported in the government ASoT. However, we need some mechanism to capture the detailed requirements and specifications for the HMI that were arrived-at through these collaborative simulation and prototyping exercises. I recognize that the government needs requirements and specifications that are relatively independent of the specific equipment and tooling used to arrive at them. I recognize that rationale in the form of human factors captured evidence and analysis also needs to be captured and provided to the government ASoT.

DEVCOM Priority:

DEVCOM Comments:

2497 As a supplier, a system that I am delivering must interface with a system that another vendor is delivering. Models of both systems are uploaded to the ASoT. There are consistency conditions that apply across different models, though. For example, a “logical” (confusing word to use) AADL model of a computer architecture must be consistent with the input and output data of a software application generated from a SimuLink specification, the nodal points and conduits and enclosures in the 3D solid model, and requirements that flow down from a SysML model.

DEVCOM Priority:

DEVCOM Comments:

2609 As a supplier, much of my “secret sauce” is in my product line models and in customizations to my internal development environment, methods, and processes. To use a manufacturing analogy, it is my assembly line that is key to my business rather than any individual product. This is also where much technical innovation occurs, and the government wishes to permit and encourage rapid technical innovation across its industrial base. The government ASoT interfaces and processes should not unduly constrain internal supplier ASoT, methods, and processes. The interface between the government ASoT and supplier ASoT should be loosely-coupled while still enabling easy exchange of technical data needed to perform contracts. That interface should be relatively simple and employ widely-used exchange formats. Even though the government and supplier may both want to be able to do similar kinds of things, such as trade studies and upgrades, suppliers do not want the interface to the government ASoT to be such that they must reveal and provide a significant amount of information or capabilities from their own internal ASoT.

DEVCOM Priority:

DEVCOM Comments:

2465 As an air vehicle supplier, the government has directed me to collaborate with a different supplier they have selected for the mission system. I will be responsible for integrating into my air vehicles the mission system kits delivered to me by that other supplier. I will be responsible for allocating physical mission system equipment (e.g., enclosures, antennas) to nodal points in the air vehicle; installing cables and wiring harnesses through conduits and routing spaces; providing an acceptable power, thermal, electromagnetic, etc., environment for mission system equipment; and integrating protocols and information flows necessary for the air vehicle system and mission system to communicate with each other. I am concerned (in fact, positive) that I will not be able to integrate an over-the-transom mission system into my air vehicle using only non-proprietary data. I want to query the ASoT during collaboration with the mission system supplier to make sure their evolving design will be integrate-able into my air vehicle. I want all technical agreements

we reach to be captured in the ASoT. I must do this without accessing any information that will not be provided to the government with rights to share with other suppliers. I want the ASoT governance procedures and practices to be such that the government can resolve issues that may arise between me and the mission system supplier, the government will be the final authority on the interface specifications between mission system and air vehicle in the ASoT. Success means that integrations of delivered mission systems into my air vehicles will occur within planned cost and schedule and without requiring any proprietary data from the mission system supplier.

DEVCOM Priority:

DEVCOM Comments:

2633 As an airworthiness qualification expert, we issued an AQP that encouraged the use of STPA. The contractor responded with a System Engineering Management Plan and AQS that proposed STPA starting with the DoDAF Capability and Operational Views, then using that to guide development of a functional architecture in SysML, and subsequently allocating functional specifications to hardware and software components in an AADL model of the mission system architecture. As an airworthiness qualification expert, I want to review the STPA analysis produced at all levels and models, in particular to assure that the flow-down of more abstract STPA analysis at higher levels of modeling correctly and consistently flows down to the lower levels of modeling.

DEVCOM Priority:

DEVCOM Comments:

2610 As government and supplier, the ASoT for individual pieces of information will flow back and forth between the two of us. During a contracted procurement of a new system, the ASoT for much information will be the supplier ASoT, and the government ASoT will mirror that. During a contracted procurement for an upgrade, the ASoT for much information will be the government ASoT. Both government and supplier ASoT should support a flexible scheme for assigning who is the ASoT for specific pieces of information, this should not (for example) be tied to a specific model on in a specific repository throughout the entire system life cycle.

DEVCOM Priority:

DEVCOM Comments:

2612 As the government, I want my ASoT to support Multi-Disciplinary Analysis and Optimization (MDAO) across the assets in my ASoT. This goes beyond trade studies based on statistical sampling of the design space. An assembly of appropriate design optimization tools, with an appropriate overarching manager to coordinate their use, should be supported. Note that trade studies and design optimization may depend on design variation points and not just on “features” visible to the customer (in the sense they are variations on the functionality and capabilities of the product). The optimization and trade study capabilities do not need to be an integral part of the ASoT, but the ASoT must minimally provide interfaces that enable this.

DEVCOM Priority:

DEVCOM Comments:

2487 Contractor A is required to compile code provided by contractor B. There is a bug in the code that prevents it from compiling with Contractor A’s RTOS compiler. Contractor opens an issue report in the ASoT and assigns it to contractor B. Contractor B fixes the issue, associates the fixed digital artifact with the issue report, and assigns the issue back to Contractor A to validate.

DEVCOM Priority:

DEVCOM Comments:

2489 Contractor A provides a software delivery to the mission system integrator. The mission system integrator discovers several problems with the delivered software and opens issues using the ASoT. Contractor A uses an in-house issue management system. To manage the relationship between their in-house issue tracking system and the ASoT, Contractor A references URIs for each of the ASoT issue reports in their internal system to provide traceability back to the ASoT.

DEVCOM Priority:

DEVCOM Comments:

2417 Contractor A uses several modeling tools including MagicDraw and Simulink, each of which has its own digital storage format. Working copies of these models are stored in private Contractor A systems including a network shared drive and in MagicDraw teamwork cloud. When Contractor A is ready to deliver work products to the government, Contractor A synchronizes its local storage with the ASoT.

DEVCOM Priority: High Priority

DEVCOM Comments: Cary Pool: Low Priority Cary Pool: This appears to be a convienance feature for a contractor to eaisly deliver updates to gov't. Automatic delivery would be useful but is less important than methods to verify / validate once delivered.

2545 Contractor uploads a collection of design artifacts to the ASoT including a set of AADL models, four CAD models in Solidworks, and a SysML model. During the Critical Design Review, the government asks the contractor to show how an item in the CAD model is related to items in the AADL models and SysML model. Contractor uses the ASoT to list traceability links between the artifacts, showing that the chassis shown in the CAD model is represented as a device named steel_chassis in the AADL model and as a block called “chassis” in the SysML model.

DEVCOM Priority: Critical Priority

DEVCOM Comments:

2546 During airworthiness qualification review the government determines that the SysML diagram describing the components of the cockpit door is missing a block for the door’s locking mechanism. The government asks whether this is an error in the SysML diagram or whether the locking mechanism has been forgotten throughout the design process. The contractor uses the ASoT to find other models of the cockpit door in Solidworks and determines that the lock has been included. The contractor updates the SysML diagram to include a block for the door and creates a SAME_ITEM_DIFFERENT_REPRESENTATION link between the CAD model’s door lock and the SysML block door lock.

DEVCOM Priority:

DEVCOM Comments:

3112 Employees choose what OS they prefer to work on, so IT cannot enforce company wide system updates. Company want to monitor who uses what OS, whether the OS is fully up to date and is running approved antivirus software. Why: Company must be able to prove they meet government defined security standards. Company must also plan for future costs associated with non-free software updates. How: ASoT could provide up to date records of system status, software updates, and virus scans.

DEVCOM Priority:

DEVCOM Comments:

3045 Fielded systems that have the FACE component incorporated must be notified that a new variant has been produced correcting the original. The correction impacts execution time and interface behavior protocol. Any changes in the FACE component involving interface, behavior or execution time during the system engineering phase and all subsequent phases would cause a message notifying the system engineer to re-run his analyses, notification of the changes would also be made to all subsequent users of the FACE component on this project and on all using projects. Each system architecture using the component would have to be evaluated. Adaptation of the FACE component might be required creating variants by system. Behavioral, timing, scheduling analyses need to be run to estimate the effect in all using systems.

DEVCOM Priority:

DEVCOM Comments:

2412 For a program, rather than deliver software via electronic transfer or CD/DVD to the Government where the Government has contractually obtained Government Purpose Rights or Unlimited Rights in the deliverable, the contractor points the Government to a website to download the software. Upon downloading the software, a pop-up menu comes up which states, “Accessing this website and downloading the code it contains indicates the User’s consent to all the terms and conditions of the site.” By clicking OK, the user could potentially inadvertently override its rights stated in the contract with the rights specified in the websites End User License Agreement/”Click Wrap” License Agreement. The terms in End User License Agreements and/ or “Click Wrap” agreements, may be in conflict with the terms negotiated in the contract and may contain unacceptable terms to the Government. The Government wants to ensure that any software delivered on the program are delivered by means which will are consistent with and uphold the rights specified in the contract. Why: The Government does not want to inadvertently give up its negotiated rights. How: ASoT can provide guidance for language and process for the contract terms and conditions via documentation and guidance for special clauses noted in the DoD OSA (Open Systems Architecture) Guidebook. ASoT can assist in tracking delivery methods and requirements from CDRLs of the contract to help avoid inadvertent nullification of its negotiated rights.

DEVCOM Priority:

DEVCOM Comments:

3043 For reliability analysis links to the controller reliability data also needs to be integrated. Test results should be integrated to verify assurance case requirements.

DEVCOM Priority:

DEVCOM Comments:

2476 How: Governance of ASoT would ensure that most recent clauses are contained in ASoT. Standardization could be obtained by a user interface that would have various data elements to choose from, like contract type, contractor institution type, subcontractor information, deliverable information, description of technology and other information. Note: Acquisition.gov now has access to various clauses through a system called FAR Smart Matrix to pull up various FAR clauses that are up to date that can be searched or applied based on criteria such as contract type etc. It includes references to supplemental Regulations such as DFARS but not with the same filtering capability.

DEVCOM Priority:

DEVCOM Comments:

2496 I support government Physical Configuration Audits, which “verifies that the related design documentation matches the Configuration Item (CI) as specified in the contract…” ({LINK title="http://acqnotes.com/acqnote/tasks/physical-configuration-audit" uri=http://acqnotes.com/acqnote/tasks/physical-configuration-audit}). A PCA is normally conducted when the government plans to control the detail design of the item it is acquiring via the Technical Data Package. However, I also want to assure that information about key software and hardware interfaces (necessary to satisfy MOSA requirements) that has been uploaded to the ASoT accurately describes the as-built and delivered system. A variety of methods will be employed, such as inspection and examination of evidence provided by the supplier. I want to assure that ASoT information remains accurate as the system undergoes updates throughout its lifecycle. For the delivered items, I need to determine what ASoT information needs to be verified against the as-built system and record what evidence is used to provide that assurance. Success is determined by the degree to which a post-delivery activity such as upgrade, test, or maintenance is hampered because the ASoT does not accurately describe the as-built system.

DEVCOM Priority:

DEVCOM Comments:

2614 I want to be able to do a virtual model-centric airworthiness qualification process. This is a controlled process, so I want to model and track the process. This is a regulated process, so features like auditing and assurance and non-repudiation (such as digital signatures) are needed. This is an evidence-based process, so I need some way to globally organize and manage evidence such as a safety case that supports satisfaction of safety requirements. Some of the evidence may be proprietary and stored only with the supplier, such as detailed design and test results; I need to be able to cite this evidence in a tamper-resistant and unique way so that it could be audited if necessary at some future time.

DEVCOM Priority: High Priority

DEVCOM Comments: Cary Pool: Could this be summerized to :Provide certified individual test results and provide a certified summary of all test reports?

2615 I want to be able to do uncertainty modeling and management and risk assessment and management, primarily in the areas of safety; security; and project execution. The ASoT must store and manage information needed for model and simulation Verification, Validation, and Accreditation (MVV&A). VV&A includes constraints on the applicability of a model, the “envelope” or domain of use in which that model is trustworthy. This concept can be expanded to a broader desire to make queries about the accuracy, completeness, trustworthiness, etc., of information obtained from the ASoT. Where I am using the ASoT to make decisions, I want to assess the trustworthiness of all the ASoT information that I am relying on to make that specific decision (for that specific purpose or “envelope” of use). I want to associate information (meta-data) about uncertainty, sensitivity, and margins to pieces of information.

DEVCOM Priority:

DEVCOM Comments:

2613 I want workflow automation capabilities. These do not necessarily have to be an integrated part of the ASoT, but minimally the ASoT must provide interfaces that enable this.

DEVCOM Priority:

DEVCOM Comments:

2583 Integrator 1 acquires a Reusable Software Component (RSC) from Vendor A for System X. Integrator 2 acquires the same RSC for System Y. Integrator 1 queries the ASoT for the list of System X artifacts, and the ASoT responds with a list that includes the RSC from Vendor A. Integrator 1 queries the ASoT for a list of systems using the RSC from Vendor A, and the ASoT responds with just System X. Why: Different integrators wish to keep their source selection private. How: The ASoT implements discretionary access controls for the Government to isolate the development activities and artifacts of Integrator 1 from Integrator 2.

DEVCOM Priority:

DEVCOM Comments:

2593 Integrator 1 acquires an RSC (version 1) from Vendor A for System X. Vendor A performs an update (version 2) on the RSC that introduces software from an unauthorized third party. The Government queries the ASoT for the version of the RSC with verified provenance, and the ASoT responds with version 1. Why: The Government must prevent the introduction of unauthorized software updates. How: The ASoT implements software integrity controls for the Government to confirm the authenticity of all software updates.

DEVCOM Priority:

DEVCOM Comments:

2595 Integrator 1 acquires an RSC from Vendor A for System X, and the Government approves the RSC for System X. Vendor A updates the RSC (version 2) per request from a different integrator, but the Government does not approve version 2 for System X. Integrator 1 queries the ASoT for the authorized version of the RSC to integrate into System X, and the ASoT responds with version 1. Why: The Government must authorize all changes to System X. How: The ASoT implements access controls for the Government to restrict access to only approved system updates.

DEVCOM Priority: Critical Priority

DEVCOM Comments: Cary Pool: Obviously Gov't should control it's CM for System X. However Integrators should control CM for each of their components submitted to ASoT for evaluation. So the ASoT should evaluate all RSC versions submitted and provide compatability reports for both Integrator submitting and gov't.

2589 Integrator 1 acquires an RSC from Vendor A for System X. Integrator 1 tests System X using CUI and directs the ASoT to store test artifacts. Vendor A queries the ASoT for test results related to its RSC, and the ASoT responds with artifacts that do not contain CUI. Why: Integrator 1 must prevent the disclosure of CUI to external parties, including its vendors. How: The ASoT implements discretionary access controls for Integrator 1 to isolate build and test artifacts containing CUI from all vendors.

DEVCOM Priority: High Priority

DEVCOM Comments: Cary Pool: Irrelevant Cary Pool: I expect that ALL data submitted to ASoT will be CUI of some sort. All groups which access the ASoT should be vetted and access should be secured. All other access issues should not be CUI related but Data Rights related.

2591 Integrator 1 acquires an RSC from Vendor A for System X. Integrator 1 tests System X using information classified at security level “beta” and directs the ASoT to store the test artifacts. Vendor A is cleared to security level “alpha”, and “alpha” is strictly less than “beta”. Vendor A queries the ASoT for test results related to its RSC, and the ASoT responds with artifacts classified

no higher than “alpha”. Why: Integrator 1 must prevent the disclosure of classified information to those without need to know. How: The AsoT implements mandatory access controls for the Government to restrict Integrator 1’s build and test artifacts to those with need to know.

DEVCOM Priority:

DEVCOM Comments:

2585 Integrator 1 acquires an RSC from Vendor A for System X. Later Integrator 1 replaces Vendor A’s RSC with an RSC from Vendor B. Next the Government replaces Integrator 1 with Integrator 2. Integrator 2 queries the ASoT for the list of System X artifacts, and the ASoT responds with a list that includes the RSC from Vendor B (but not the RSC from Vendor A). Why: Vendor A protects its IP based on acquisition contracts. How: The ASoT implements discretionary access controls for Vendor A to restrict access to its IP to Integrator 1.

DEVCOM Priority:

DEVCOM Comments:

2587 Integrator 1 builds its own RSC with protected IP and integrates it into System X. Next the Government replaces Integrator 1 with Integrator 2. Integrator 2 queries the ASoT for all artifacts relating to System X, and the ASoT responds with a list that includes Integrator 1’s protected RSC. Integrator 2 queries the ASoT for internal details about the RSC, and the ASoT rejects the query. Why: Integrator 1 protects its IP based on contracts with the Government. How: The ASoT implements discretionary access controls for Integrator 1 to restrict access to its IP to the Government.

DEVCOM Priority: Critical Priority

DEVCOM Comments:

2597 Integrator 1 checks in an update to System X that corrupts the system build. Integrator 1 queries the ASoT for the previous build of System X before the update. Why: The ASoT must ensure that vendors, integrators, and the Government can recover and restore a system following system update errors. How: The ASoT implements a change management system that can roll back to a previous safe state.

DEVCOM Priority:

DEVCOM Comments:

3110 Management sends request that all access to company resources be removed for a former employee. IT team wants to know what resources each employee has access to, and what data resources were for their use only. Why: Employee departure requires access to all resources be removed. In the case of employee termination this access must be removed quickly. The team does not want to have to go through every VM on the network to find where access must be revoked. They also don't want to waste CPU time on VMs that were only used by former employees. How: ASoT could provide records of all employee resource access.

DEVCOM Priority:

DEVCOM Comments:

2484 Market Research is a critical part of the knowledge-based acquisition and should be conducted prior to developing new requirements and releasing solicitations so there is a clear understanding of the products/services available in the marketplace. The key for people conducting market research is to have requirements stated as a problem statement and desired outcome and the

process should facilitate robust communications between solution providers and Government to gain knowledge for follow on acquisitions resulting in a better match of capability to desired outcome. Full guidance regarding Market Research is provided in FAR 10. The Government wants to centralize and document the procedure and results of Market Research. The Government is obligated by FAR 10 to research whether there are commercially available solutions or non-developmental items to satisfy its needs. Why: Complete market research so that the most cost-effective way of procuring an item/service is met. How: ASoT may contain prior market research data including notes from face to face discussions, demonstration information, current procedures to meet FAR 10 guidance and up to date information on contractors and their products/deliverables on recent programs.

DEVCOM Priority:

DEVCOM Comments:

3012

DEVCOM Priority:

DEVCOM Comments:

3046 OSD is considering adding a new security control based on threat information which requires forcing it on legacy aircraft systems. Normally each PM would be required to provide the information on the update cost. However, OSD wants the DE databases to permit generating a rough cost is estimated based on system information stored in the ASoT for each system. This information might be based on whether the change can be inserted without forcing the computing hardware to be updated, using processor, bus, and memory utilizations and estimates of the resources required to insert the new security control. Other concept of operations analysis would be desired, to chart new strategic capabilities that can be added without significant cost.

DEVCOM Priority:

DEVCOM Comments:

3123 Proposed new restrictions in the safety profile for the FACE standard means many legacy software components will no longer conform. The Airworthiness Qualification Expert queries the ASoT for the impact to existing mission systems as a result of this proposed change, and the ASoT responds with a detailed list including links to architecture models that illustrate the affected components and their current AQ analyses. Why: The AQ Expert needs to understand the impact of the proposed change on current mission systems. How: The ASoT links results to the specific models that produced those results. The models include searchable details such as FACE conformance.

DEVCOM Priority:

DEVCOM Comments:

2478 Rather than routing word or pdf documents for signature, such as funding approvals, marketing analysis, and other contractual documents required in the contract award process, the Government wants to have a system for management to signoff without having to route documents and to have all documents in stored one place. Why: Routing can take a significant amount of time and often can sit with one person or get lost in an inbox. Latest versions may not be easily accessible making document version control difficult. How: ASoT could provide central system for document generation, notification for action, and routing.

DEVCOM Priority:

DEVCOM Comments:

2450 Related to the classified information access for a participant story, the ASoT staff should not be able to access any of the model information and details, nor should they be able to figure out roles in the system development unless that is public. Why: Admin super users for the ASoT could be in a position to access the entire system design and exfiltrate it. How: ASoT staff should be limited to things required to administer the ASoT. The data and information within the ASoT should be encrypted or otherwise not accessible to those staff members.

DEVCOM Priority:

DEVCOM Comments:

3042 Scenario 1 - Control System model for the autopilot must be integrated into the architecture model for code generation. User must find the model, link it into the AADL model, the AADL generator will generate the control code, extract the timing information for its execution, verify that it is the correct and latest version, verify that the rate is unchanged from the AADL specification, and that the code can be executed within the AADL specified execution time by running a Worst Case Execution Analysis toolset on that code. If the control system model is changed, the AADL environment would be notified of the change. It would notify the AADL modeler of the change to the architecture, who then would have the option to allow the system to re-run all relevant architecture analyses to verify that all requirements still hold, including timing, scheduling, safety and security, etc. These would all be up to date since the same process would occur relative to the ASoT data sources. If the system development is to the point where it can be re-generated, and the prototype would be tested on an instruction set simulator, if it passes the qualification related models and qualification users would all be notified of the change and the results of the analyses. Code would be autogenerated for the physical platform. Then qualification tests would be re-run to validate timing and ability to pass the qualification tests. All properties that can be measured on the physical hardware would be re-validated or updated. A second review of the models for correctness and qualification would be run.

DEVCOM Priority:

DEVCOM Comments: (per Alex Boydston) This scenario is a good example for timing analysis.

2480 Standard documents needed in the execution of the contract or during negotiation would be made available to contractors in one system and submitted back to the government via the system. Why: A number of standard documents such as Reps and Certs, IP Assertion Tables, SF298, DD882, Cover Pages for Final Reports etc., are currently emailed to contractors for completion during the negotiation or contracting process. The method for receiving the information can be inefficient and time consuming. Emailing out a document or two at a time and waiting for response slows the award process, for example. How: By accessing these documents and uploading them to a system upon completion would save time, money and ensure that all the required documents are completed and stored in a common access database.

DEVCOM Priority: Low Priority

DEVCOM Comments:

3116 Story: Change in component software criticality

DEVCOM Priority:

DEVCOM Comments:

3122 Story: Change in conformance to the FACE standard

DEVCOM Priority:

DEVCOM Comments:

3118 Story: Change in environment requirement for software criticality

DEVCOM Priority:

DEVCOM Comments:

3120 Story: Change in software to requirement traceability

DEVCOM Priority:

DEVCOM Comments:

3048 SysML models across organizations need to be integrated to understand system interoperability related to timelines, data protocols, definition, precision, and data flow. The SysML models involve integration across multiple SysML tool vendors.

DEVCOM Priority:

DEVCOM Comments:

2576 System Engineer has several variants of a design for a digital flight bag that are actively under consideration. One has a larger battery; one has a smaller battery. The engineer wishes to delay the decision as to which implementation to use, so he or she leaves both options in the model. The ASoT runs analysis on both variations, producing results for both variants that are reported in the web interface to the ASoT.

DEVCOM Priority:

DEVCOM Comments:

2548 System engineering team designs a rotor in SysML from concept to deployment. That thread has many places where it breaks off. If I start talking CAD, there are data standards for that. There is a relationship between the CAD and the systems engineering model. What SysML brought is a system engineer's modeling a language. Before we just had spreadsheets and word documents. The aeronautical engineer has CFD models. When I go down the different disciplines, you've got different data standards. There is not a single engineering discipline that doesn't use some form of a model. What we're trying to do is connect them together. Not translating them, but connecting them. The gap that exists today is the interoperability between standards. For 2.0 we're defining an API so that we can be more interoperable. Tyler – interoperability means drop in? Gene – Tony is more correct, integration might be a better word. Tony – we have sort of a large set of different domain specific languages that we have to reference, interoperability is dependent on some kind of neutral way to link things. At some level you run into fundamental incompatible translations or places where trying to make sure that the system is integrated such that Y=X you have degradation in Y=X.

DEVCOM Priority:

DEVCOM Comments:

3108 Team wants to know order of operations to restore network. Why: Networks are composed of components that depend on each other to function. A server can depend on DHCP, network storage and other network services to boot. How: ASoT can contain documentation on dependence relationships of system components.

DEVCOM Priority:

DEVCOM Comments:

3041 Thanks, Jerome raises some issues related to qualification and changes to a qualified system which is captured in a model that was used to auto-generate the system. Issues related to acceptable models would be similar - that assumptions/qualifications can be provided. Tying a use case to DevOps is also good. Integrating data from multiple tools where there is a one to one mapping is the starting point, and integrating from tools where there is a complex mapping to derive the needed data for the AADL (or other) model is the next. There were some interesting cases in backward mapping discussed in the SysML to AADL meeting, to map back discovered errors and are the possible choices that should be supported. Trace back of at least an error indicator and cause of the error to the appropriate owner of the authoritative truth. I think there are interesting cases where semantics are not close but data needs to be extracted. Establishing who is the authoritative truth holder, both vertically and horizontally, across engineering domains. For users to determine the consistency of data and the definition of data, the owner must populate the database with significant amounts of extra information. Must be a responsible person at the higher or horizontal level for the data he owns. The user must create the transformation into his model since he knows what he needs, what does he have to know to establish consistency within his model? For each transformation, he must be notified if the definition is changed, as well as the value, and the value must be traceable back to it's owner when an issue arises. The recalculation can be automatic as long as the relationships hold. But when you go to the next level of decomposition, new data extractions, and consistency verifications are required. To what level of fidelity is the data required? the user has to know if he is extracting imprecise or abstract data into his model and the owner has to have defined to what level of fidelity and abstractness the data was provided. Is it an estimate or measured, is it a rough budget, a goal or a requirement? As things become more formal and qualified, the definitions have to be more precise. What is the effect in the model of imprecise data, perhaps a range is needed to help. The owners of the data will change over time, perhaps from SysML model, then from an engineer who measured the property and provided it in another place in the database, for actual performance measurements. What about the govt's need to inspect information and aggregate it for costing purposes - for instance, how much time and cost did you spend on requirements, or on deriving this requirement, what did it cost to implement or what will be the cost to change this requirement to a stronger requirement. What about extracting data across SysML models that are in different customized expressions of SysML, will the semantics be defined such that consistent integrated models can be constructed. What about internal data in models that is not exposed for analysis use but is inconsistent across the models, similar to strong typing concerns. It could be inconsistent at this lower internal level even though outputs have the same units and definition.

DEVCOM Priority:

DEVCOM Comments:

3529 The ASoT Provider uses Product Line Management (PLM) tool RedCat as part of its ASoT. The government selects MSI A for a project. MSI A wishes to continue to use its own PLM tool, BlueDog, because parts of its solution are in a product line with a variety of uses. The ASoT provider configures the ASoT RedCat tool to exchange data with the BlueDog PLM.

DEVCOM Priority:

DEVCOM Comments:

3117 The Airworthiness Qualification Expert learns of a severe vulnerability in a software component commonly used in mission systems. The Expert queries the ASoT for a list of affected mission systems, and the ASoT responds with a detailed list including links to architecture models that

illustrate the use of the component and the AQ analyses that depend on the component. Why: The Expert needs to understand how invalidated assumptions about the component will impact current missions. How: The ASoT links results to the specific models that produced those results. The models include searchable details such as version and build.

DEVCOM Priority:

DEVCOM Comments:

3267 The Architect requests that automated system power load analysis be run at hourly intervals. The ASoT automatically executes the load analysis at hourly intervals, emailing the Architect when the result of the analysis does not meet a pre-determined limit.

DEVCOM Priority:

DEVCOM Comments:

3049 The DoD is requiring a new safety device to be installed in all aircraft. The new device requires internet connection to the mission processor on each system. It also must be wired through the structure of each aircraft. The safety device must be redundant and wired such that a 12 inch projectile penetrating the structure at any point will be able to function, with its location, wiring, and processing capability. Models of the system involved are the structural, wiring, and AADL specification of processors upon which the software is bound. Wiring must be such that the network radiation does not effect other functions of the aircraft and within engineering specifications for proper operation of the network.

DEVCOM Priority:

DEVCOM Comments:

3528 The Government begins a project using ASoT provider X. Two years into the project, the Government decides to change ASoT providers to provider Y. Provider Y has similar capabilities to provider X, but uses different internal tools (for example, using a different product line management tool). Despite this difference, provider X is able to transfer all of the project information to provider Y.

DEVCOM Priority:

DEVCOM Comments:

2410 The Government may desire to obtain greater rights in technical data and Computer software than those rights normally granted in accordance with the DFARS. These special rights or licensing terms are typically documented in Section H special provisions of the contract. The Government will also need the delivery of and appropriate rights in (1) Interface Control Documents (or similar) at a level sufficient to address the integration of a Contractor's commercial or proprietary software into the procured system and (2) deliverables and artifacts required to identify physically and functionally interchangeable/replaceable items. Why: Government will need appropriate data rights in all of the supporting documentation for integration of existing systems for reuse in the future. Any special data rights and licensing terms should be considered clauses in section H of the contract. The Government may want to license certain proprietary software to third parties in the future. How: DoD OSA (Open Systems Architecture) Guidebook contains an extensive collection of special Section H clauses which may be considered for use to capture wording or clauses relevant to the Governments data rights needs. ASoT could provide data with regard to reference material and also data on existing clauses in contracts associated with special data rights or licenses to assist the program manager and contracting officer to create contract terms to meet the Government's goals.

DEVCOM Priority: High Priority

DEVCOM Comments: Cary Pool: Low Priority Cary Pool: I may not be clearly undestanding this, however it appears to low level in the details to be key to the demonstration.

3341 The Government publishes the program schedule through the ASoT. The MSI and suppliers reference the schedule in their internal plans, linking to it from their scheduling tools via the ASoT API. When the Government makes a schedule change, other stakeholders are notified.

DEVCOM Priority:

DEVCOM Comments:

2603 The Government solicits bids for an update to System X, operating at security level “beta”, that will permit coalition partners, cleared to security level “alpha” (where “alpha” is strictly less than “beta”), to access System X. This change moves the system from a single-level, system-high mode of operation to a multi-level secure mode of operation. The Government queries the ASoT to evaluate each bid for its information assurance provisions. Why: The Government must confirm that the bid includes the appropriate architectural isolations and required security controls to assure correct MLS operation. How: The ASoT implements methods to assess models included in the bid for required characteristics.

DEVCOM Priority:

DEVCOM Comments:

2482 The Government wants to be able to track performance and costs against a Integrated Master Schedule (IMS) or Integrated Master Plan (IMP) for the program and by performer. Why: It is time and cost effective to be able to track contractor performance in a single place (ie. Dashboard or system) without having to reinput information stemming from progress reports or other schedule data provided by contractors. How: ASoT could provide a dashboard for master schedule and cost data, provided that contractor data is entered into the system or a common template.

DEVCOM Priority:

DEVCOM Comments:

2406 The Government wants to create a standard set of requirements for contractors who supply a data model as a deliverable for a Unit of Conformance (UoC). They would like consistency from various performers on the program in data model deliverables. The FACE Technical Standard has a detailed set of data model requirements. Why: Government wants consistency in data model development and results. How: ASoT contains FACE documentation to assist in the development of requirements for the deliverable. The requirements detailed in the FACE Technical Standard could apply regardless if FACE Conformance Verification is required.

DEVCOM Priority:

DEVCOM Comments:

2471 The Government wants to standardize reporting formats for monthly reports, summary charts, quarterly reports, final reports etc. and be able to view these electronically. Why? Contractors may be sending in reports in their own formats (pdf, WORD) via DOD SAFE sites. Government has to then take those reports and store them in a system. It is not easy to do searches or find information without opening the documents. File management takes time. How: ASoT could provide web based standardized input areas for user interface (for example) for sections of reports or summary charts. The reports could be stored in a system used both by the government and by

the contractor (along with other data like financial progress, deliverable status etc). Because it is in a database, instead of individual files, information is easily accessible. Example of this in practice is DARPA's TFIM system (pre 2.0 version) and the ARMY SBIR website uses a similar model for their summary chart requirements.

DEVCOM Priority:

DEVCOM Comments:

2404 The Government will provide software as GFI from a previous contract to a new contract with a different performer. The awarded contractor of the new program wants to know the licensing of the the GFI and also wants to understand any associated open source licensing rights in open source components of the software. The contractor is tasked with adapting the software to a new platform. Why: The contractor wants to know how to mark the newly developed software and what, if any, licensing wording or issues may come up with open source components. How: ASoT will contain previous marking information and licensing information on the GFI.

DEVCOM Priority:

DEVCOM Comments:

3119 The Govt AO monitoring the mission system deployment learns of a new mission requirement that will increase the required criticality rating for a key software component. The Govt AO directs the Airworthiness Qualification Expert to query the ASoT for the analysis results that the Expert must update as a result of this new requirement, and the ASoT responds with a detailed list including links to architecture models that illustrate the use of the component and the AQ analyses that depend on the component. Why: The AQ Expert needs to understand the impact of the new requirement on existing AQ results. How: The ASoT links results to the specific models that produced those results. The results include searchable details such as the derived software criticality rating for each component.

DEVCOM Priority:

DEVCOM Comments:

3337 The MSI and Supplier use the shared documentation wiki provided by the ASoT to exchange notes and synchronize on terminology.

DEVCOM Priority:

DEVCOM Comments:

3005 The MSI changes the power requirements of their mission system computer. They make this change in their AADL model and upload it to the ASoT. The ASoT automatically inspects the change and propagates it to other models that depend on the power requirements of the mission system computer. The ASoT automatically triggers analysis of the models and reports the results back to the MSI and to the Government.

DEVCOM Priority: High Priority

DEVCOM Comments: Cary Pool: Does this scenario mean they changed the power requirements before an actual unit was delivered to the government so the government doesn't have CM of this? Or is this a new revision of their system that the gov't needs to approve the change for?

3339 The MSI needs to determine where a data interface is defined. The MSI connects to the ASoT Dashboard and uses its search function to search for the data interface. The search encompasses multiple data representations and metadata including author, organization, owner, etc.

DEVCOM Priority:

DEVCOM Comments:

3340 The MSI uses a modeling tool that their suppliers do not have a license to use. When suppliers view the model through the ASoT dashboard, the ASoT provides a translated version of the model that is loadable with the suppliers' tools. The ASoT maintains a traceability link between the representations, noting that the translated version of the model is not the official version.

DEVCOM Priority: High Priority

DEVCOM Comments:

3271 The Mission System Integrator has a deployment workflow that includes generation of C++ code from SysML and AADL models. The Mission System Integrator uploads a build script that guides the code generation process. The ASoT executes the code generation process at a regular interval, making the resulting code available to the appropriate stakeholders.

DEVCOM Priority:

DEVCOM Comments:

3274 The Program Manager for the MSI configures a dashboard that shows her the elements of the system design that are flagged as high risk based on her selected metrics: change frequency and change scale. The ASoT automatically queries the digital artifacts and reports to the program manager which modeled elements are highest risk according to the metrics on her dashboard.

DEVCOM Priority:

DEVCOM Comments:

3272 The Program Office and Mission System Integrator form a joint Change Control Board that provides approval to major system changes. The Program Office creates a new business process in the ASoT that describes the review steps and the necessary stakeholders required for each step. During the CCB meeting, the required stakeholders sign into the ASoT web interface and provide review feedback, either signing off on required changes or sending them back to their authors for additional input.

DEVCOM Priority:

DEVCOM Comments:

3338 The Supplier uses the ASoT to submit digital artifacts to the MSI for integration into the system design.

DEVCOM Priority:

DEVCOM Comments:

3044 The System Engineer has developed his SysML models to the point where he would like to generate analyses in AADL to verify that the FACE components he has included in his models for reuse can be functionally integrated and that timelines for execution can feasibly not exceed the end to end timing requirements for their interaction on a generic processor environment. The SysML model would be translated into AADL with the appropriate properties creating links to the SysML model for the components and the properties. The repository for FACE components

would be linked to extract the FACE specifications, including behavior specifications in AADL or in SysML. The FACE to AADL translator would be run to integrate and formally verify that at the functional level the integrated behavior of the components is feasible. Execution times of the FACE components on the generic processor would be scaled to the generic processor in the SysML model. Timing analysis would would be run to establish a minimum end to end latency, which if it exceeds the requirement, would cause the System engineer to consider alternative FACE components or other trades to enable the meeting the requirement, such as a faster generic processor. The user, the SysML modeler needs to link to all the appropriate resources to accomplish the analysis.

DEVCOM Priority:

DEVCOM Comments:

2430 The Vendor provides a SysML model of the system with some intellectual property included. Components that are to be delivered with GPR are annotated in the SysML model so that their data rights are clear when the government procures the model. (See Digital Backbone SysML).

DEVCOM Priority:

DEVCOM Comments:

3356 The government has selected PLM tool X and the MSI has selected PLM tool Y. Both the government and the MSI wish to maintain their own "ASoT" for different programs (the government's being the platform under development, the MSI's being a product line). The government ASoT provides the capability for the government to query the MSI's ASoT and provides recordkeeping to track when ownership of an digital artifact passes from the MSI to the government.

DEVCOM Priority:

DEVCOM Comments:

2416 The government instructs contractor A to conduct all coordination with contractor B using the government-owned ASoT. Contractor A uploads design documents and models to the ASoT and the government grants Contractor B permission to view but not edit the digital artifacts uploaded by Contractor A. Contractor B creates model artifacts, but does not want to share the entire model with Contractor A. Contractor B uploads some digital artifacts that only the government can see and other artifacts that both the government and Contractor A can see.

DEVCOM Priority:

DEVCOM Comments:

2423 The government issues a solicitation in the form of a BAA. The government decides that this BAA will be the top-level project managed by an ASoT. The government initiates a formal process in the ASoT for receiving and tracking proposals. Each received proposal is stored in the ASoT and has access and use restrictions according to the formal process.

DEVCOM Priority:

DEVCOM Comments:

2432 The government or ASoT provider configures the underlying hardware and software for the ASoT such that it can store classified information. An airgap between classified and unclassified hardware is required, so the logical ASoT is split into two physical infrastructure items (e.g., two

separate servers). Associations between classified and unclassified information are stored in the classified ASoT partition because including links to classified information on the unclassified partition could leak information.

DEVCOM Priority:

DEVCOM Comments:

2421 The government procures a helmet design from vendor A with Government purpose rights. After an initial manufacturing run, the government decides to use SysML models of the helmet in a BAA GFI package. Using the ASoT the government creates a derivative "snapshot" of the helmet model with critical details removed to protect the intellectual property of vendor A. The elements of the snapshot are traceable to the original.

DEVCOM Priority:

DEVCOM Comments:

2488 The government receives a draft final report from a contractor. The government reviews the final report and makes several comments using the ASoT. To make the comments, the government creates a new issue in the ASoT and creates comments with references to specific elements in the final report.

DEVCOM Priority:

DEVCOM Comments:

2440 The government wants to conduct a remote System Requirements Review with vendors A and B, who are on a team. Prior to the review, the government creates a review process instance. Following that process, the government creates a milestone in the ASoT and requests that Vendors A and B indicate artifacts relevant to that review.

DEVCOM Priority:

DEVCOM Comments:

3354 The government wishes for contractors to deliver models as part of a contract. The government writes Contract Deliverable Data Items, specifying a standard DID for models and customizing it according to features needed for planned government traceability of analysis. Why: The Government wants standardized models so that the models can be usable and maintainable. How: The contractors look up the model based DID provide their deliverables through the ASoT.

DEVCOM Priority: High Priority

DEVCOM Comments: There is a lack of DIDs for models and there is much need to have developed.

3355 The government wishes for contractors to deliver models as part of their proposals. As part of the RFP, the government includes a template model for contractors to modify. The contractors submit their proposals to the ASoT, including their populated templates. Why: The government wants to provide a framework for proposal data that enables analysis. How: The ASoT enables storage and tracing of original government template models and performer submission models.

DEVCOM Priority:

DEVCOM Comments:

2538 The government wishes to have a dashboard displaying status from a variety of projects so that their status can quickly be monitored. The government users log onto an ASoT server and assemble custom dashboards from available information sources, such as requirements databases, analysis servers, and model repositories. The ASoT creates each user's dashboard by polling all of the linked services for information, then rendering it for the user.

DEVCOM Priority:

DEVCOM Comments:

3121 The mission system Integrator wishes to replace two components with a single component that will handle all required capabilities. The Integrator notifies the Airworthiness Qualification Expert who queries the ASoT for the impact to existing AQ analysis as a result of this change. Why: The AQ Expert needs to understand how the component replacement will affect existing AQ results for the mission system. How: The ASoT links results to the specific models that produced those results. The ASoT automatically reruns analyses on updated models, and the AQ Expert compares the old and new results.

DEVCOM Priority: High Priority

DEVCOM Comments: From Cary Pool: This would be a huge value if it were possible to do this. Clear measureable testable AQ requirements with clear results are absent in the enduring fleet. This is currently a huge cost and time synch due to requiring human committee analysis.

3332 The primary data center for the ASoT encounters a power outage. Remote stakeholders need to access data, so the ASoT switches to continuity of operations mode in which a secondary data center provides access to mirrored data.

DEVCOM Priority:

DEVCOM Comments:

2408 The program manager is writing a SOW and is considering language to ensure reuse of existing software and code. The PM wants to trace the actual reuse of software and source code as the program develops and analyze actual versus projected reuse. The PM wants to know the costs associated with reuse (licensing needed, qualification challenges). The PM wants to understand the data rights they already have in existing software. Why: Reusing existing software for which the Government already has rights to may be cost effective throughout the lifecycle and beyond. Writing these requirements into the SOW are important to maintain traceability for rights, funding sources, historical information and future use etc. How: ASoT provides access to information useful in preparing SOW language through sources such as the FACE Contract Guide, and the DOD OSA (Open Systems Architecture) Guidebook. ASoT can provide historical information on data rights for specific preexisting SW and documentation.

DEVCOM Priority:

DEVCOM Comments:

2454 This is probably a special case of the subcontractor example where a participant is the lead of one system design and contributing subsystems or components to other parts. Contractual firewalls can be established to avoid a conflict of interest. The ASoT should provide the mechanisms to enforce those safeguards. Why: Same as the single subcontractor case – competitive, proprietary, and/or security reasons. How: Need to know approach needs to be fine grained enough to consider contractual relationships, maybe designating certain users go with certain contracts.

DEVCOM Priority:

DEVCOM Comments:

2438 To allow reclamation of storage space, the ASoT tracks expiration dates on artifacts. Some artifacts, such as contractual deliverables, are kept indefinitely. Other artifacts, such as interim work products, expire after the conclusion of the project and can be deleted or archived offline.

DEVCOM Priority:

DEVCOM Comments:

2436 To make artifacts easily discoverable, engineers providing content to the ASoT provide metadata about each artifact describing its content, intent, authorship, etc. This metadata does not require specialized tools to view, so the government can search for content without acquiring specific tools.

DEVCOM Priority:

DEVCOM Comments:

2434 To minimize infrastructure costs, the government contracts with ASoT supplier RRR to provide the ASoT servers. The government certified RRR's ability to handle the sensitivity levels of information to be stored in this ASoT. At the end of RRR's contract with the government, the government opts to take control of the ASoT infrastructure and invokes a contract option for RRR to support transfer of ASoT information to government servers.

DEVCOM Priority:

DEVCOM Comments:

3333 To save storage space on the primary ASoT infrastructure, the ASoT provides an archive capability through which old information is either archived or removed accordingly to policy. The ASoT provider uses this infrastructure to keep costs down.

DEVCOM Priority:

DEVCOM Comments:

2730 User Story: ASoT Commissioning Speed: Why: As the ASoT becomes state of practice approach to system design and development, not only will it need to be maintained over years, but new ASoTs will be created for new developments. How: ASoT should include a mechanism to commission a new ASoT based on one or more existing ASoTs and provide a structured approach for rapidly standing up a new ASoT.

DEVCOM Priority:

DEVCOM Comments:

2709 User Story: ASoT Physical Infrastructure: ASoT will be deployed on a range of different infrastructures. Why: A cloud-based solution is perhaps the most logical one today, given the current IT infrastructure support. These need not all be hosted on internet facing installations though. Some may be on private clouds that are disconnected from the main internet. Checks should be in place to prevent one of these private cloud installations to be accidentally connected to the Internet. How: The infrastructure provider will need to provide ASoT overall bug fixes, feature enhancement, user support, code maintenance, and technical support in such a way as to maximize up time. Good designs will have redundancy that includes backups across a broad geographic range, so issues one part of the world does not prevent the ASoT from going down.

DEVCOM Priority:

DEVCOM Comments:

2684 User Story: ASoT-wide constraints: System-level, or non-functional, constraints will permeate the ASoT, from individual components, up through subsystems, systems including the software and hardware that forms those parts. These different constraints will have different scoping, some will be scoped only to a particular part of the system, for example, power budget for a mission computer. While others will be broader, up to the overall system itself. As the system is designed, the bounds on those constraints will be in flux as different tradeoffs are made. Why: The ASoT needs to support a constraint change process that enables the appropriate stakeholders to make changes within their span of control. How: The ASoT should help to avoid two key situations: a) every change, even locally scoped, has to be made at the highest level or go through a long bureaucratic change and b) changes at any level are made within the scope of the constraints at the higher level. For example, adding new CPUs should be supported so long as the CPU doesn't invalidate a higher level power budget.

DEVCOM Priority:

DEVCOM Comments:

2735 User Story: Capturing Manufacturing Constraints: Among the non-functional requirements should be manufacturing constraints. Why: Directly impacts the speed that a system can be physically built. How: Probably can be addressed like other properties and constraints in the system models. Some thought will need to be given to scope of those constraints and enabling flexibility to different ways of manufacturing the same system. One vendor might do it all in one factory while another pulls integrated systems from all over.

DEVCOM Priority:

DEVCOM Comments:

2649 User Story: Certificate of Correctness: The DARPA Meta BAA has a notion of certificate of correctness. This seems like a model-based version of the certification standards above. Why: The ASoT should support inclusion of additional certification processes that may not be necessarily required by the certification authority, but that are used in the system development process itself. How: Provide some means of associating certification 'tags' with particular versions (and subsets) of the ASoT and for defining program or system specific tags. For example, certification that program review X determined all requirements are met in the design, or tag that a particular version was the one used in a particular review. The tagging should be robust enough to allow all of the same review artifacts (exactly the same) to be generated (or accessed) at an arbitrary time after the review. This should include analyses.

DEVCOM Priority:

DEVCOM Comments:

2647 User Story: Certification and Airworthiness: A system will have several certification steps to go through before being authorized to operate, these would include security, airworthiness, and probably others. ASoT should provide support for these certifications. Why: Different standards and processes may be in place for each of items. For example, FAA rules for airworthiness differ from the military and security certification will be different all-together. Some parts of the system may not be involved in every part of the certification effort, they could be involved in several efforts, or not involved at all. How: In all cases, the ASoT should maintain the information needed for the analyses. The appropriate documentation should be able to be generated from the ASoT.

This implies a separate representation of the certification requirements and a means to conduct analyses and generate the documentation for each type of certification. Also, once a particular version of a system in an ASoT is certified for a given standard, all features in that version that were certified should be marked as approved, point to the documentation approving it, and indicate which certification process was used, and maybe for which operational environments the certification applies to. Then that version should be secured against changes. Ideally, if parts of that certified system are used in different ASoTs, that certification evidence would be applied to the context of the different ASoTs.

DEVCOM Priority:

DEVCOM Comments:

2699 User Story: Collaboration Support: Users of the ASoT will necessarily have to work with each other. However, not every user needs to work with every other user. Why: Collaboration tools including a wiki, message board, mailing lists, and other features that enable effective collaborative development should be supported. How: These should be supported in a way that prevents message overload and preserves the scoping concerns listed in the international collaboration user story. The ASoT should automatically determine scoping to be a minimal set, allowing a given user to increase the breadth. For increasing the breadth for a given thread to a different user category (e.g., different system or non-US person vs. US person), ASoT may need to enforce an approval step, which should be lightweight, but nonetheless meaningful.

DEVCOM Priority: Medium Priority

DEVCOM Comments:

2640 User Story: Cross Model Analysis. ASoT encompasses a manufacturing model library for a given airborne or ground vehicle system, including manufacturing models and code generation, all the way to digital twins. Why: Safety engineers, for example are concerned with locations of critical functions. If two VMs are hosting redundant functions, but placed on the same hardware base, then the redundancy assumption fails. If some part of a system is causing problems, the ASoT should guarantee that you can find all aspects of the system that use or rely on that failing part of the system. How: In order to draw conclusions about system level properties (aka nonfunctional properties) and track influences from different type models (solid model, software model, for example), ASoT will need to include desirable and spurious interactions, dynamics, and properties of all components down to the numbered part level.

DEVCOM Priority:

DEVCOM Comments:

2661 User Story: Design Flow: Assuming a set of metrics (see metrics user story above) the metrics should be integrated into a natural part of the ASoT supported design flow. Why: Some capabilities that could be implemented with this integration include design optimization, tradespace analysis including complexity, system attributes, adaptability, security, resiliency, and others. ASoT will have to scale to large, multi-dimensional tradespaces. How: At any given time, ASoT should be able to report out on how the system (or a part of a system) contributes to the metrics, how it is progressing towards a final state (which would probably be tied to meeting the reqts). It could also track how many resources are being expended (time, money, etc.), although an accurate assessment of this might be outside the scope of the ASoT. Ideally, this could be compared to a conventional development somehow to show savings, but this might be an apple/oranges sort of comparison, making it meaningless.

DEVCOM Priority:

DEVCOM Comments:

2665 User Story: Design Trade Space Review: For program reviews and interim checks, the ASoT should support generating and presenting the results of design trade space exploration. Why: This will be a key discussion point at reviews and the ASoT should make this easy. How: The exploration could be based on constraints derived from system specifications and based on the space of composible systems given the library of components and manufacturing methods. The ASoT should support required analyses across the whole system or selected parts (or abstractions) given a set of design choices to tradeoff and requirements (e.g., nonfunctional ones, likely) to assess.

DEVCOM Priority: High Priority

DEVCOM Comments: Cary Pool: Trade Space review should include aspects like Cyber Resilience implications of these choices which are often left out.

2728 User Story: Distributed ASoT participants: The idea is that ASoT participants could be distributed around the country, even the globe. Why: When it comes to manufacturing vs. simply designing, these distances matter in terms of lead times, integration timelines, and overall system construction. How: The ASoT should include a notion of the logistical supply chain in the representation of the system or collection of systems. Tradeoffs can be supported that look at different supply chain options. These could also include country of origin-based tradeoffs.

DEVCOM Priority:

DEVCOM Comments:

2706 User Story: Documentation generation: Why: ASoT should support automated document generation that ensures that the documentation for the system is consistent with the system being represented in the model. How: ASoT could include in its change management a means to flag documentation that requires review or update. For example, if an interface to a subsystem changes, that subystem's documentation should be flagged as obsolete. The ASoT user tasked with updating it should be given a report by the ASoT for why the documentation was determined to be out of date and which specific documents (and, even better, which specific parts of the documents) need to be updated. It is possible that updating a document could cascade other documents that need to be updated, so the analysis should be rerun until it shows no further updates needed.

DEVCOM Priority:

DEVCOM Comments:

2686 User Story: Enforcing Constraints: Related to the reporting constraint violations user story, the ASoT should help to enforce constraints, allowing variation only on specific approval. Why: If the ASoT cannot enforce constraints then the gaining benefit from ASoT will be harder. How: Similar to the ASoT wide constraints user story, constraints could be defined at different levels as parts of the ASoT, with decisions about what do to about violations made at those levels. Certain changes will require additional approvals and review. As constraints are changed, all tools should be rerun to ensure that the change didn't create other constraint violations.

DEVCOM Priority:

DEVCOM Comments:

2702 User Story: Information Assurance: Several user stories have mentioned separation and scoping roles. Why: Overall, the ASoT should include information assurance controls sufficient to protect

export controlled and contractor proprietary information in accordance with Department of Defense, Department of State, Department of Commerce, and any other statutory or regulatory requirements or best practices. See, e.g., ASD(NII), Directive-Type Memorandum (DTM) 08-027, "Security of Unclassified DoD Information on Non-DoD Information Systems." How: The approach should enable support for information assurance testing, certification, and accreditation activities.

DEVCOM Priority: Critical Priority

DEVCOM Comments:

2694 User Story: International Collaborators/Personnel: Personnel from different countries may require access to the ASoT to perform their tasks. This could be to integrate products in the above use case, or just as employees of a company working on the system. Why: It would help eliminate vendor lock if the staff supporting a system development could include non-US personnel, it would open the range of organizations that could help. How: The ASoT could provide a structured, guaranteed mechanism to enforce the scope of access these personnel have – limiting access to only their span of interest and specific ins/outs required for them to integrate into the system. No system-level reporting should be visible, for example. In fact, for good information security, reporting access should be limited to a user's need to know regardless of their citizenship or other affiliation.

DEVCOM Priority:

DEVCOM Comments:

2693 User Story: International Components: Systems being developed may include components and systems from non-US manufacturers, an easy example is a variant being created for a foreign military sale (FMS). Other examples can be functionality purchased from an international supplier for use on a US system. Why: For supply chain analyses, the sources of the components should be tracked and along with it respective releasability, export controlled, classified, etc. In addition, any licensing or export paperwork and permissions should be recorded with the items and steps taken to make designers aware of the country of origin of the system's components. How: The ASoT should support indicating the component/system/etc parts as having different releaseability and support an analysis of supply chain risk and, conversely, an analysis to minimize the risk of leaking US technology as part of a FMS.

DEVCOM Priority:

DEVCOM Comments:

2642 User Story: Legacy system. ASoT will need to support legacy systems for which models may not exist for as many aspects of the system as would be for a new build. Why: Recent block upgrades may have more detailed models than others with higher confidence that those models reflect the actual implementation. Some aspects of the system may not have any models or there may not be confidence that what is modeled is actually implemented. How: One approach would be to allow an ASoT for a legacy system to be scoped only to what is needed for a given upgrade activity, with the rest essentially a giant black box. For example, if the mission computer is being upgraded, maybe not all of the solid model needs to be represented.

DEVCOM Priority:

DEVCOM Comments:

2648 User Story: Maintaining certification. ASoT should support analyses for certification impacts of proposed new changes against certified systems and their components and subsystems. Why: If an ASoT has a record of what aspects have been certified for what standards and operational

environments, it should be possible to minimize the certification costs. How: Certification impact analysis would be another one of the non-functional requirements that would feed into trade space analysis and system-level report generation and metric tracking.

DEVCOM Priority:

DEVCOM Comments:

2740 User Story: Manufacturability/Maintainability constraint feedback to system design: Why: Early analyses of manufacturability and maintainability could be very useful as part of managing system costs. Avoid issues like needing to remove a large subassembly to perform simple preventative maintenance (e.g., needing to lift the engine to change spark plugs (e.g., from the internet, Buick Skyhawk, Chevy Monza, Pontiac Sunbird, and Olds Starfire of the '75-'80 model years were equipped with a 3.8 liter V-6). How: Treat like other properties and constraints in the system. The challenge is to enumerate them, define scoping, and develop the appropriate analyses so that developers can pick up issues early in design.

DEVCOM Priority:

DEVCOM Comments:

2727 User Story: Manufacturing Attribute Tracking: Why: To support system development all the way to manufacturing, the ASoT will need to model manufacturing attributes. For example, the parts that can be made by different contractors (work packages) or similar that could bid out. That parts would want to be swapped in and out, either for different variants or different suppliers of the same variant. How: The ASoT model library should represent the salient attributes of each modality of fabrication: lead time, cost, speed, range of applicability, speed of reconfigurability, etc. and have a way to organize a design in terms of manufacturing pieces and a process that will be followed in building the end system.

DEVCOM Priority:

DEVCOM Comments:

2651 User Story: Metrics: Users of ASoT (and their managers and funding sources) will want to see how well the work is progressing, where there are problems, how those impact the design, etc. Why: Each part of the system, and possibly each way of modeling the parts of the system, will have metrics specific to it and likely each will have their own way of representing it. For example, bandwidth is not relevant to a solid model and volume is not relevant to scheduling a software bus. In addition, system level (or non-functional reqts), by definition, apply across the system. How: Every part of an ASoT will be expected to contribute information to a system-level analysis in a way that can be composed to provide a common result. ASoT either needs a common approach for the whole scope of the ASoT, or a way to map (compose) information from each part to a common view. The goal is for system stakeholders to be able to get a system-wide picture without having to know the specifics of where all the numbers came from.

DEVCOM Priority:

DEVCOM Comments:

2669 User Story: Model-based Design Verification: Related to the certification user story above, ASoT can be used to generate a certificate that the collection of systems in the ASoT meets a particular target definition. For example, that the system meets requirements for MILS or RMF or that the requirements have all been addressed. Why: Provides a straightforward way to identify key design milestones and objectives. How: Could tie to the certification user story above. May also require appling probabilistic model composition techniques that presents a stochastic result based on the

uncertainty of the underlying model form and parameters [from the BAA]. For adaptable features of the system, the ASoT should support analyses to re-verify the system either as changes are made or that an exhaustive set of system configurations is verified.

DEVCOM Priority:

DEVCOM Comments:

2738 User Story: Modeling the manufacturing process: This level of modeling of the manufacturing process seems to be out of scope for ASoT.

DEVCOM Priority:

DEVCOM Comments:

2644 User Story: Models Used Across ASoTs: Separate ASoTs will be implemented for different organizations, collections of systems, systems, or some other granularity. Why: It is possible the same component, subsystem, or system is included in more than one ASoT. For example, an aircraft engine is a product of a company and models of that engine might be represented in several ASoTs and those ASoT may be owned by different organizations. How: The ASoT configuration management approach should ensure that the owner of that common item is in control of any changes to those models, that any changes to that common item are evaluated across all of the ASoTs in use, and that all the ASoTs get that change. The analysis would determine, for example, if the change could be made to the common product or if a variant would need to be created for a subset of the ASoTs.

DEVCOM Priority:

DEVCOM Comments:

2641 User Story: Operational Environments. Any given system will be required to operate in different operational environments (called a context model in the source DARPA BAA) and there may need to be different variants needed to support those environments. The variants will have many features in common. Why: ASoT will need to support a description of the operational environment in terms that can be mapped to system features. How: As common features are updated, ASoT should support a capability to make the update once and then promulgated or otherwise automatically made available to the variants that depend on it. The variants should maintain a mapping of how they are different from the base feature set (or which features of the feature set they include). ASoT also needs to record the operational environments a given variant can operate in. This is useful for identifying redundancy and tradeoffs. Is it better to build a more capable variant to cover several operational environments or does the tradespace analysis indicate benefits to tailored, special purpose variant to support a given operational environment?

DEVCOM Priority:

DEVCOM Comments:

2685 User Story: Reporting Constraint Violations: Inevitably, there will be constraint violations (i.e., problems identified by analysis tools). Why: Identifying them and fixing them is a key part of ACVIP and should be an integral part of ASoT. How: Analysis tools and other ASoT mechanisms will be used to detect the violations, for example a power budget exceeded or a MILS violation discovered and these violations could cross several different model types. ASoT should generate a minimal set of changes that led to the problem, and notify the smallest number of people that can fix it. That is the ASoT should not overly report problems, but leave it to those in a position to fix them – at least during development. A review would be a different situation. This could be scoped by the scope of models involved in the problem and notification sent to those development teams.

DEVCOM Priority:

DEVCOM Comments:

2667 User Story: Review Support: ASoT support for system reviews should include a means to use the ASoT directly to support the presentation. So that the design review is being conducted is actually on a specific version of the system (or part of the system) in the ASoT. Why: The performer would be able to present the results of design trade space exploration based on constraints derived from system specifications and based on the space of composible systems given the library of components and manufacturing methods. How: The live presentation should be structured so that higher level decision makers can see a system-level perspective that does not require detailed knowledge, but then support drill down when questions are asked. Specific comments, requests, changes, etc. that are discussed in the review could be captured as things to do on a given aspect of the system. Then, at the next reporting period or review, it would be easy to report on progress made in addressing them.

DEVCOM Priority:

DEVCOM Comments:

2715 User Story: See user story on international components. Open source is in this same category.

DEVCOM Priority:

DEVCOM Comments:

2733 User Story: Tracking Flow of Goods in ASoT: This was touched on earlier. Why: A key part of an ASoT is to support rapid reconfiguration to accommodate new designs or design changes. How: The ASoT should enable trade-offs between system reconfigurability, including reprogramming existing capabilities and also adding new capabilities, and should present the designer with a tradespace of options based on a variety of tradable parameters. To support these trades and to measure program performance, offerors should propose specific figures of merit that encompass the degree of adaptability, cost, reconfiguration speed, and fabrication timelines associated with a given foundry design.

DEVCOM Priority:

DEVCOM Comments:

2695 User Story: Version Control: Why: A key part of an ASoT is version control. A few examples were listed in previous user stories for its importance (e.g., tagging releases and versions used in reviews, etc.). How: The version control approach should enable check-out and check-in, reversion, bug/issue tracking, and other features that enable effective cross organizational collaborative development and project maintenance that simultaneously supports organizational and other boundaries. A means to support public and private branching, with the ability to do some sandbox development. Version control could provide the basis for collecting metrics and statistics on access patterns, user activity, and design (re)use.

DEVCOM Priority:

DEVCOM Comments:

2419 User X at Contractor A uploads a SysML model to the ASoT. User Y uploads a revision to the SysML model. User X recognizes that the revision is incorrect and inspects the history of the SysML model to determine when the change occurred. Upon discovering the source of the change,

User X reverts the incorrect change. The ASoT retains history of the original version uploaded by User X, the incorrect version uploaded by user Y, and the revision executed by user X.

DEVCOM Priority:

DEVCOM Comments:

2425 Vendor A supplies a “mock up” model to the government as part of a proposal. The government (having not yet made a selection) puts this model in the ASoT, noting that it cannot be used unless Vendor A’s proposal is selected. The government also provides a low trust rating for the model, as its informal description as “mock up” indicates that it should not be trusted for detailed design planning.

DEVCOM Priority:

DEVCOM Comments:

2427 Vendors A and B separately confirm to the government that they have signed NDAs enabling them to view one another’s design artifacts for a given project stored in the ASoT. The government updates each Vendor organization’s permissions to enable viewing of the others’ artifacts in the scope of the given project.

DEVCOM Priority:

DEVCOM Comments:

2533 Version 1.3 of a code generator is found to have a defect that can result in invalid code. Version 1.3 was previously certified at TQL-4. The MSI uses the ASoT to find all artifacts and test results generated using toolchains in which version 1.3 played a part, then marks all of the results and artifacts as invalid in the ASoT, citing the error in version 1.3 as the reason.

DEVCOM Priority:

DEVCOM Comments:

2531 Version 4.3 of a static analysis tool is certified for TQL-3. The MSI records this certification in the ASoT by creating a tool entry and a tool version. When the ASoT invokes the static analysis tool as part of its automated testing, the results of the test include a reference to the applied tool and its version so that the resulting artifacts can be traced to the tool that generated.

DEVCOM Priority:

DEVCOM Comments:

2532 Version 4.4 of a compiler has not passed formal TQL verification, but the government wishes to provide a waiver based on time constraints so that testing can proceed on schedule. The government uses the ASoT to log the waiver’s application to version 4.4 of the compiler.

DEVCOM Priority:

DEVCOM Comments:

2429 When creating a new project in the ASoT, the government sets default access restrictions based on the allowed content of the project. When adding content, users are informed of the allowed information (e.g., NO CLASSIFIED INFORMATION) so that they do not unwittingly add private information and are able to mark supplied information with appropriate restrictions (e.g., CUI) so that sensitive information is treated appropriately by the ASoT.

DEVCOM Priority:

DEVCOM Comments: Cary Pool: High Priority Cary Pool: "What mechnisms enable the gov't or contractor to purge data that is inappropriately added to the system? What mechanisms exist to prove data is only apropriate data?"

3051 An engineer is setting up the data that he needs from the ASOT but can not find it without re-constructing it from multiple other inputs. He needs to know if this data is measured and to what degree of accuracy or is an estimate. He needs to identify the owner of that data and create a dependency on that data for his model. He must understand if it is adequate just for this phase of development or if he needs to refine his query as his design proceeds . He needs to understand any assumptions behind the data that might invalidate it's use. He also needs to publish the data his models make available and provide all the qualifying information. As his model improves, for instance now based on actual measurements in the system, he needs to post this information also, such that users can refine their models.

DEVCOM Priority:

DEVCOM Comments:

3050 In a DevSecOps scenario, the data from the field indicates that an aircraft or missile system is occasionally locking up, requiring rebooting at unpredictable times. Runtime monitoring indicates that the bus is being overloaded from time to time causing events to be missed or delayed beyond the expected time causing missed or invalid data input. Information is needed from the field reports to understand which function is causing the overload. Software developers in the software factory will need to re-develop the offending functionality. Then architectural verification will be required to insure that other issues have not been created in the real-time concurrent execution of the system. Data recorded from the missions that seem to trigger the issue must be used to simulate and test the fixes as well as overall operational capability and re-qualifying the system for flight worthiness.

DEVCOM Priority:

DEVCOM Comments:

B ASoT Ontology

Ontology Introduction We created the ASoT ontology using Sparx Enterprise Architect. While conducting our literature survey and stakeholder interviews, we took note of key terms and added a UML class for each key term, populating its **notes** property with references to the term's source. We created UML class diagrams to show the relationships between the terms. The remainder of this section is generated by Sparx Enterprise Architect.

Table of Contents

ASoT Ontology

Package in package 'Model'

The Authoritative Source of Truth (ASoT) Ontology describes terms used in defining the capabilities of an ASoT. In addition to textual definitions, the ontology describes the *relationships* between terms, indicating cardinality and association.

Domain Model diagram

Class diagram in package 'ASoT Ontology'

This model provides a top-down view of the key terms in the ASoT Ontology.

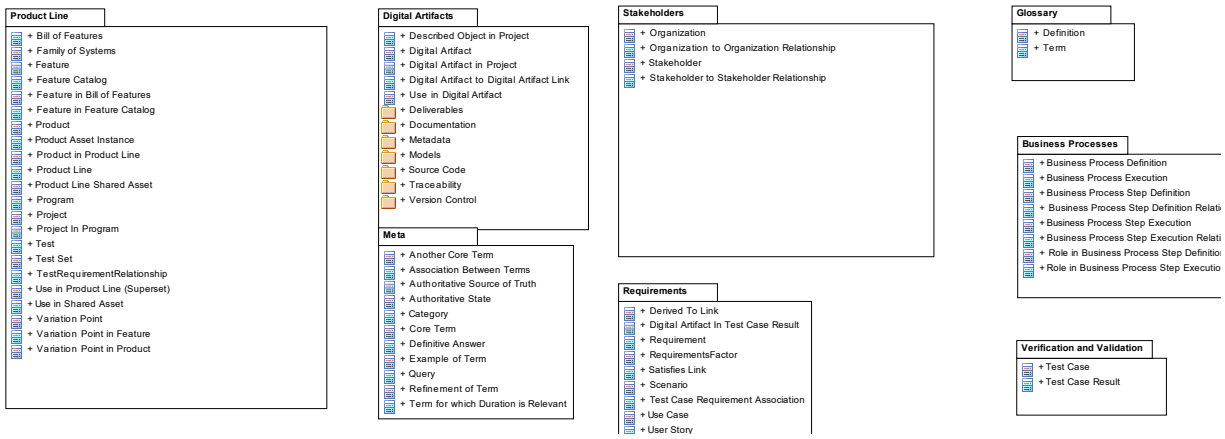

Figure 1: Domain Model

Domain Package Model diagram

Package diagram in package 'ASoT Ontology'

This model provides a high level organizational view of the different areas of the ontology.

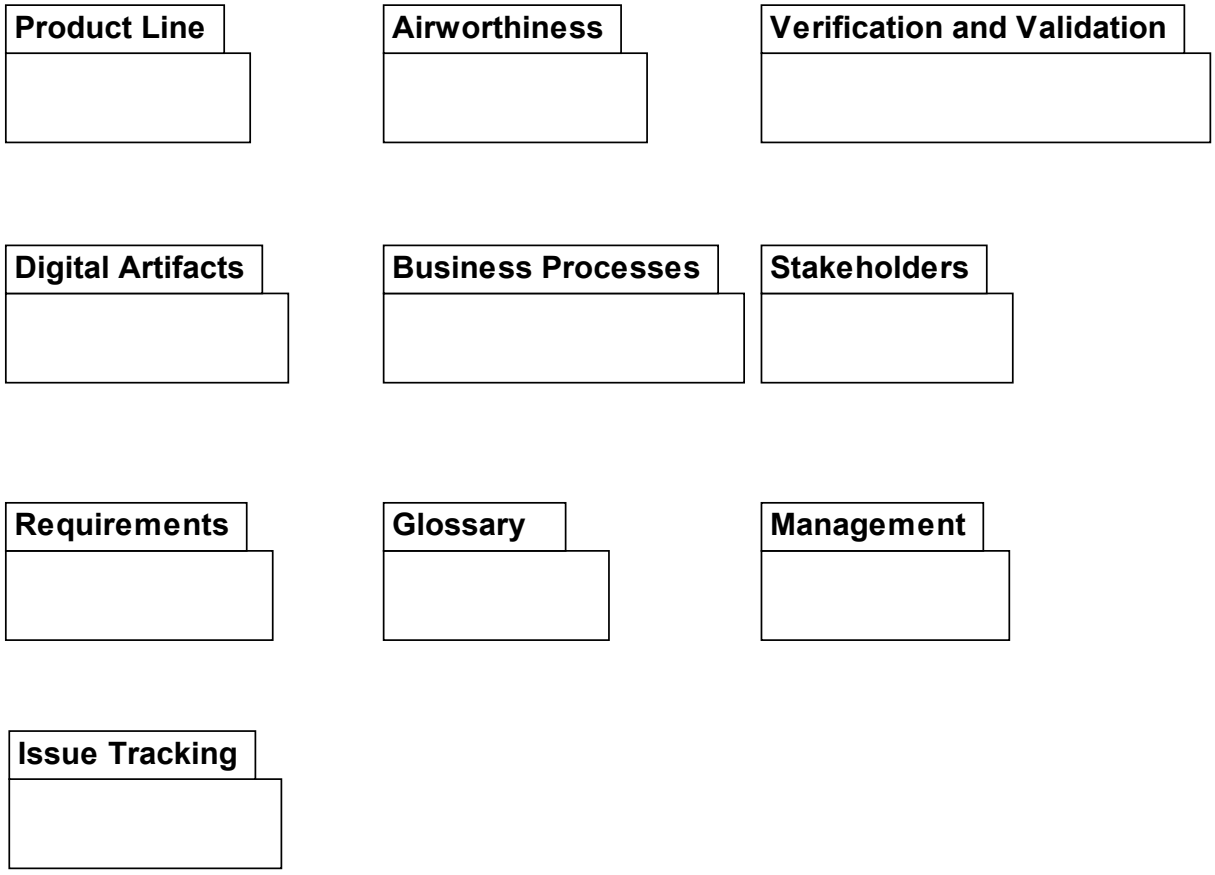

Figure 2: Domain Package Model

Airworthiness

Package in package 'ASoT Ontology'

As defined by AR70_62,

[Airworthiness] prescribes policies, responsibilities, processes, and procedures for life cycle airworthiness (design, production, and continued) of manned and unmanned aircraft systems and subsystems, including the installation of allied equipment and modifications to Army aircraft.

Airworthiness diagram

Class diagram in package 'Airworthiness'

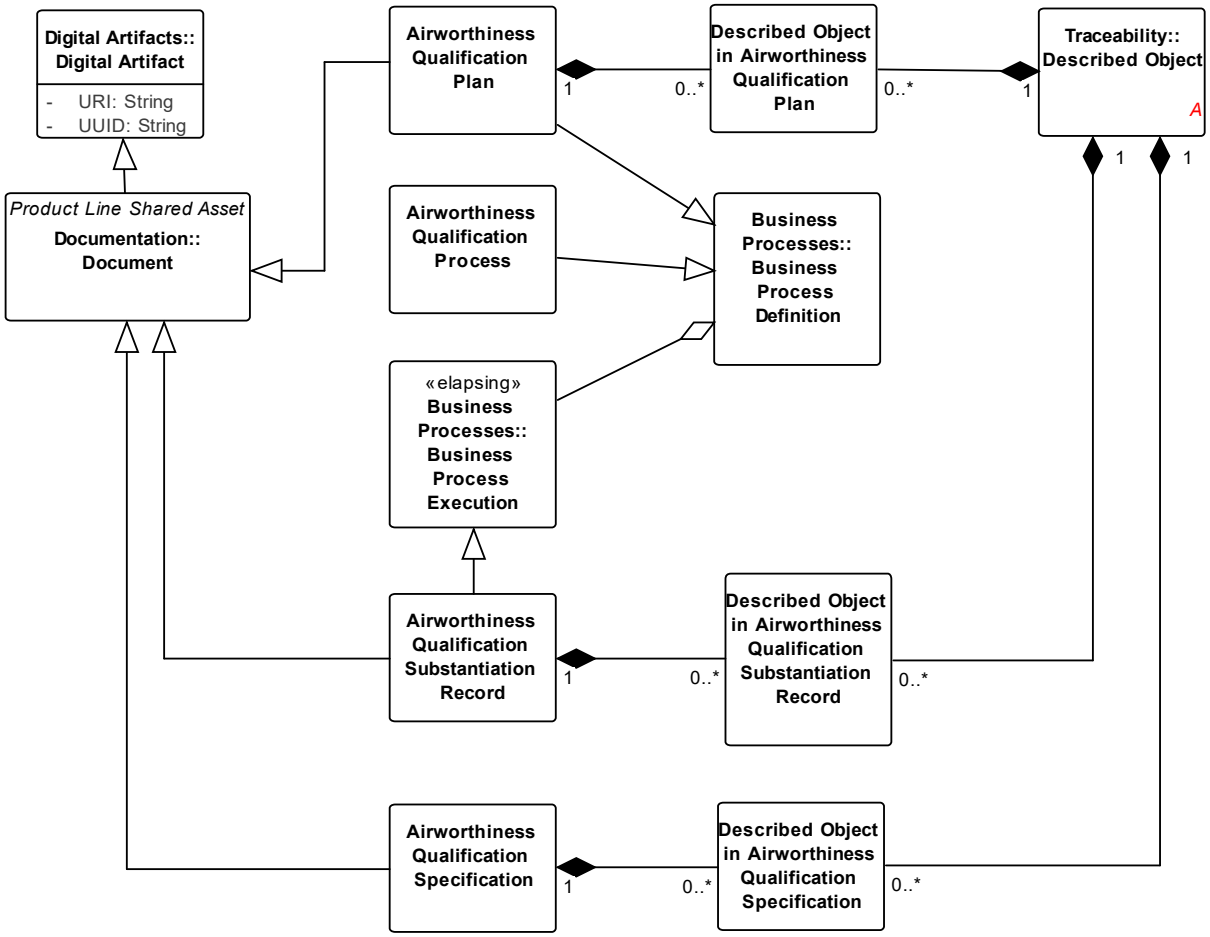

Figure 3: Airworthiness

Airworthiness::Airworthiness Qualification Plan

From https://armypubs.army.mil/epubs/DR_pubs/DR_a/pdf/web/r70_62_FINAL.pdf

Requirements for airworthiness qualification developed by the airworthiness authority when requested by the procuring activity.

Airworthiness::Airworthiness Qualification Process

For example, the Airworthiness process defined by AR70_62 is an airworthiness qualification process.

https://armypubs.army.mil/epubs/DR_pubs/DR_a/pdf/web/r70_62_FINAL.pdf

Airworthiness::Airworthiness Qualification Specification

From https://armypubs.army.mil/epubs/DR_pubs/DR_a/pdf/web/r70_62_FINAL.pdf

The AQS defines the contractor's obligation to conduct specific analyses, reviews, tests, surveys, and demonstrations to fulfill the requirements and objectives specified in the AQP.

Airworthiness::Airworthiness Qualification Substantiation Record

From https://armypubs.army.mil/epubs/DR_pubs/DR_a/pdf/web/r70_62_FINAL.pdf

A technical summary describing the scope of the qualification and its results, including prescribed limits, and a compilation of each requirement indexed to its status of demonstrated compliance and references to the verifying technical substantiation (including analyses, inspections, drawings, modeling, simulations, test plans and test results, and any other relevant technical data).

Airworthiness::Described Object in Airworthiness Qualification Plan

An Airworthiness Qualification Plan may refer to specific objects that are described in other digital artifacts.

Airworthiness::Described Object in Airworthiness Qualification Specification

An Airworthiness Qualification Specification may refer to specific objects that are described in other digital artifacts.

Airworthiness::Described Object in Airworthiness Qualification Substantiation Record

An Airworthiness Qualification Substantiation Record may refer to specific objects that are described in other digital artifacts.

Business Processes

Package in package 'ASoT Ontology'

This package describes procedures to be carried out by people or machines.

Business Processes diagram

Class diagram in package 'Business Processes'

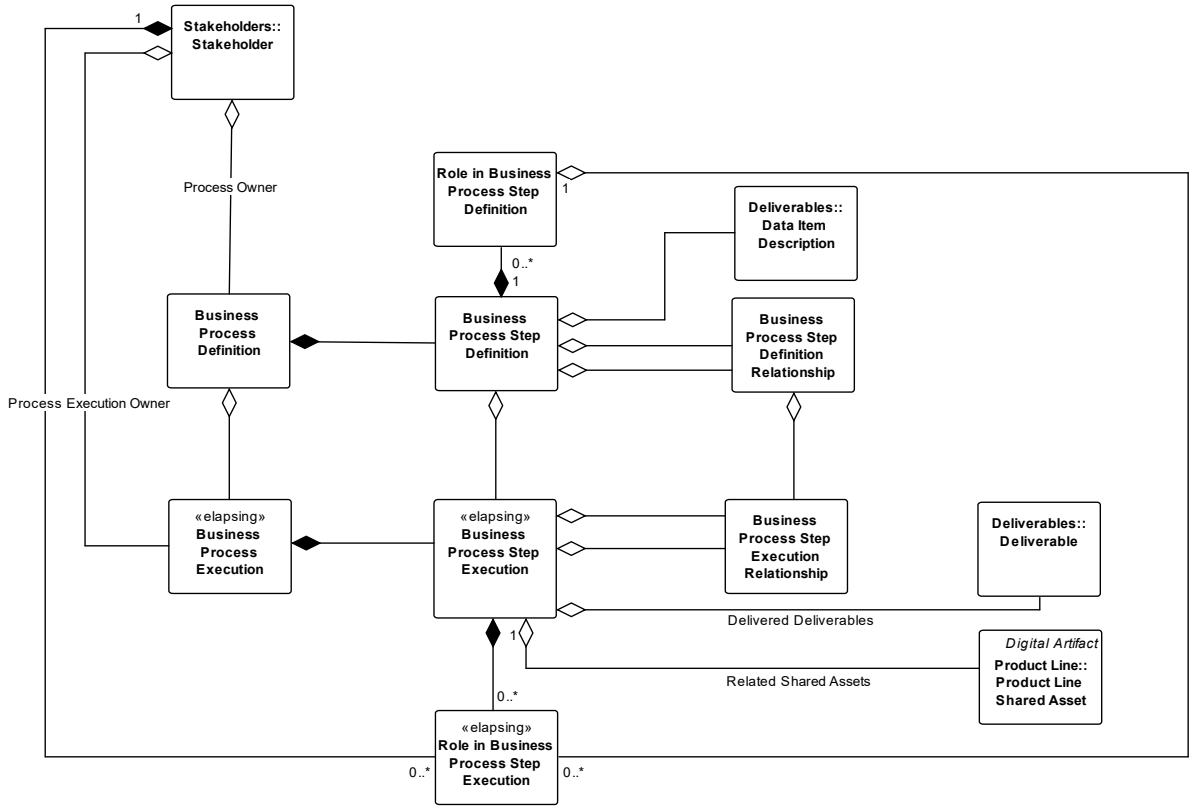

Figure 4: Business Processes

Business Processes::Business Process Definition

A process definition describes a set of steps required to complete a formal procedure. Note that "Business Process" is used to contrast with the POSIX "Process" concept.

For example, a process definition could describe the steps required to complete a formal code review.

From https://sebokwiki.org/wiki/Business_Process_(glossary)

An inter-related set of cross-functional activities or events that result in the delivery of a specific product or service to a customer. (ISACA 2012)

From the DAU Glossary:

Process
1.) The combination of people, equipment, materials, methods, and environment that produces a given product or service. A process can involve any aspect of a business. 2.) A key tool for managing processes is statistical process control, a planned series of actions or operations that advances a material or procedure from one stage of completion to another. 3.) A planned and controlled treatment that subjects materials to the influence of one or more types of energy for the time required to bring about the desired reactions or results.

Business Processes::Business Process Execution

A specific execution of a business process, for example the execution of the Airworthiness Certification Business Process on the new ABC helicopter.

Business Processes::Role in Business Process Step Definition

Various stakeholders may be involved in the execution of a given business process. For example, an airworthiness signoff may require official signoff from the airworthiness authority as well as the program manager.

Cybersecurity

Package in package 'ASoT Ontology'

Cybersecurity diagram

Class diagram in package 'Cybersecurity'

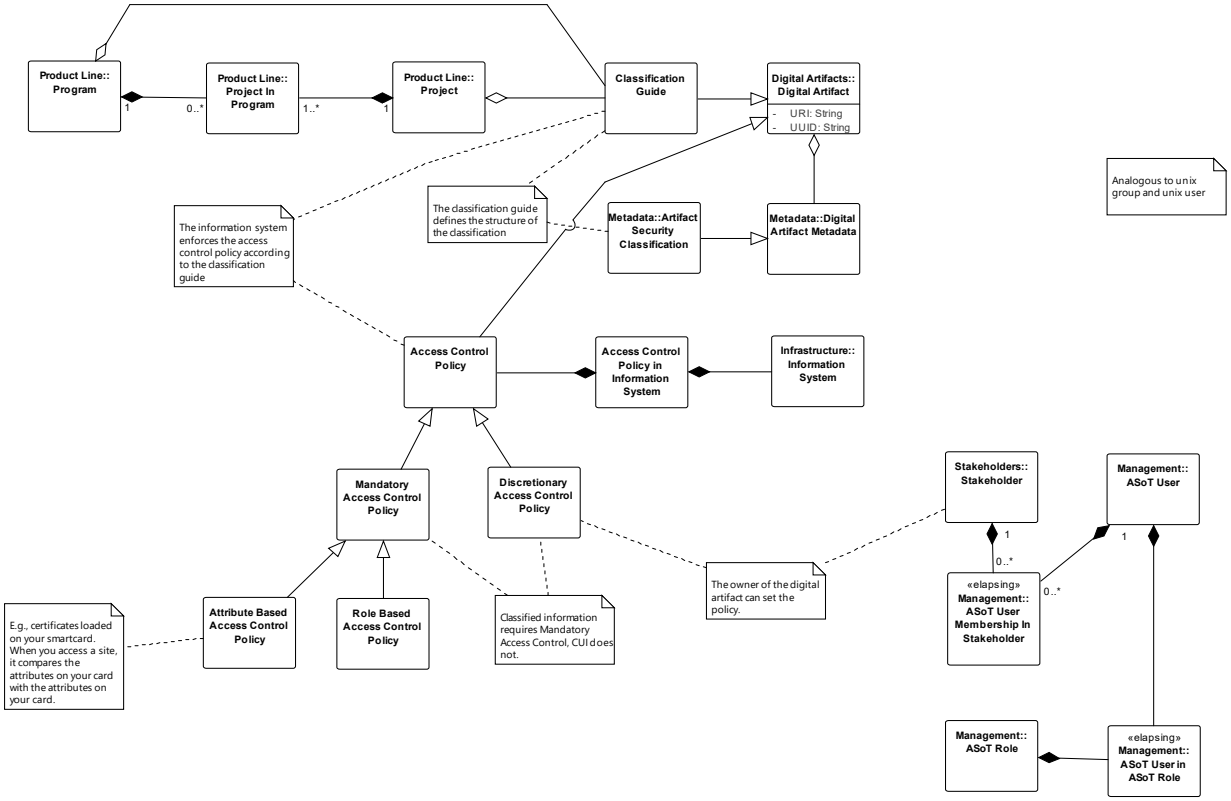

Figure 5: Cybersecurity

Cybersecurity::Access Control Policy

From CNSS:

Access Control:

The process of granting or denying specific requests: 1) for obtaining and using information and related information processing services; and 2) to enter specific physical facilities (e.g., Federal buildings, military establishments, and border crossing entrances).
Source: FIPS PUB 201-1 (adapted)

Cybersecurity::Attribute Based Access Control Policy

From CNSS Glossary:

Access control based on attributes associated with and about subjects, objects, targets, initiators, resources, or the environment. An access control rule set defines the combination of attributes under which an access may take place. See also identity, credential, and access management (ICAM).

Cybersecurity::Discretionary Access Control Policy

From the CNSS Glossary:

An access control policy that is enforced over all subjects and objects in an information system where the policy specifies that a subject that has been granted access to information can do one or more of the following: (i) pass the information to other subjects or objects; (ii) grant its privileges to other subjects; (iii) change security attributes on subjects, objects, information systems, or system components; (iv) choose the security attributes to be associated with newly-created or revised objects; or (v) change the rules governing access control. Mandatory access controls restrict this capability.
Source: NIST SP 800-53 Rev 4

Cybersecurity::Mandatory Access Control Policy

From the CNSS Glossary

An access control policy that is uniformly enforced across all subjects and objects within the boundary of an information system. A subject that has been granted access to information is constrained from doing any of the following: (i) passing the information to unauthorized subjects or objects; (ii) granting its privileges to other subjects; (iii) changing one or more security attributes on subjects, objects, the information system, or system components; (iv) choosing the security attributes to be associated with newly- created or modified objects; or (v) changing the rules governing access control. Organization-defined subjects may explicitly be granted organization-defined privileges (i.e., they are trusted subjects) such that they are not limited by some or all of the above constraints.
Source: NIST SP 800-53 Rev 4

Cybersecurity::Role Based Access Control Policy

From the CNSS Glossary, Role Based Access Control is defined as:

Access control based on user roles (i.e., a collection of access authorizations a user receives based on an explicit or implicit assumption of a given role). Role permissions may be inherited through a role hierarchy and typically reflect the permissions needed to perform defined functions within an organization. A given role may apply to a single individual or to several individuals.
Source: NIST SP 800-53 Rev 4

Digital Artifacts

Package in package 'ASoT Ontology'

This package describes information managed by the ASoT.

Artifacts Packages diagram

Package diagram in package 'Digital Artifacts'

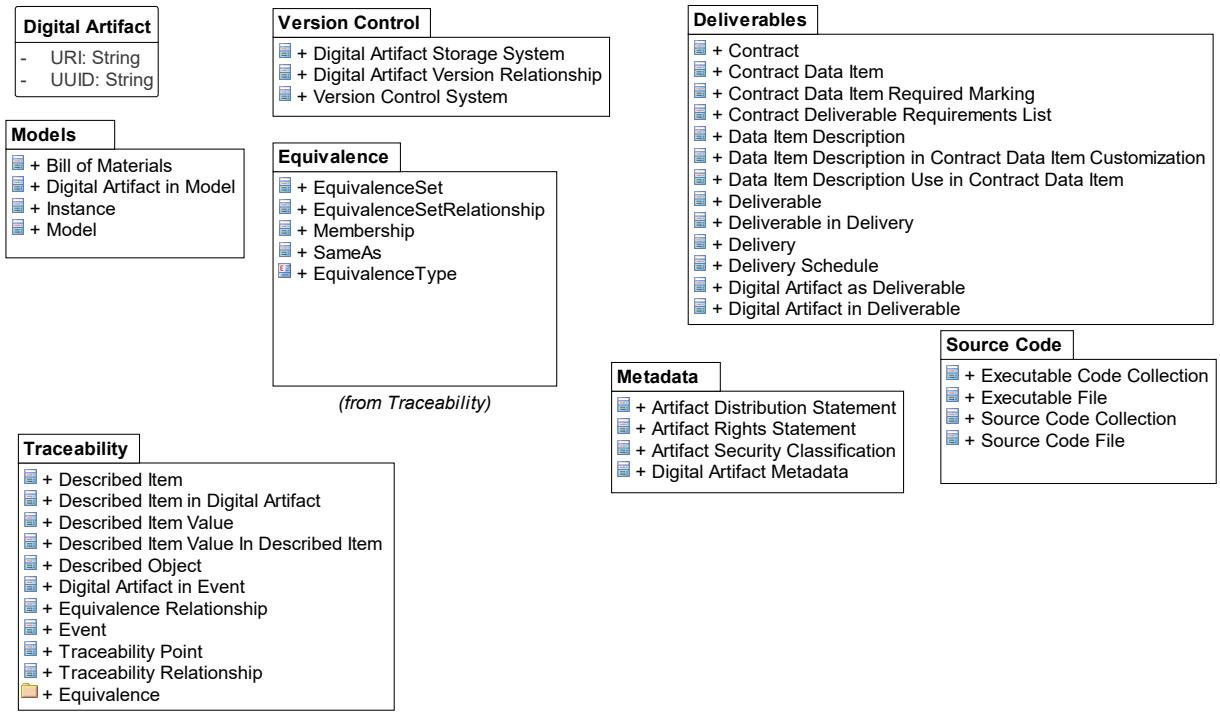

Figure 6: Artifacts Packages

Artifacts in Projects diagram

Class diagram in package 'Digital Artifacts'

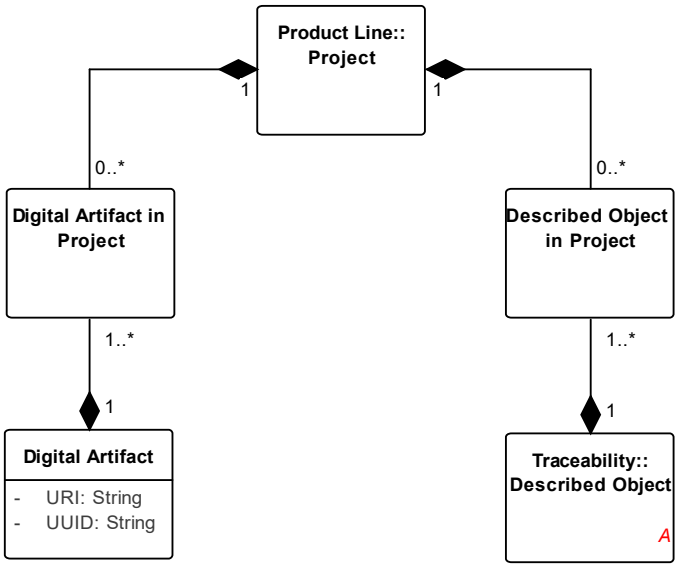

Figure 7: Artifacts in Projects

Digital Artifacts diagram

Class diagram in package 'Digital Artifacts'

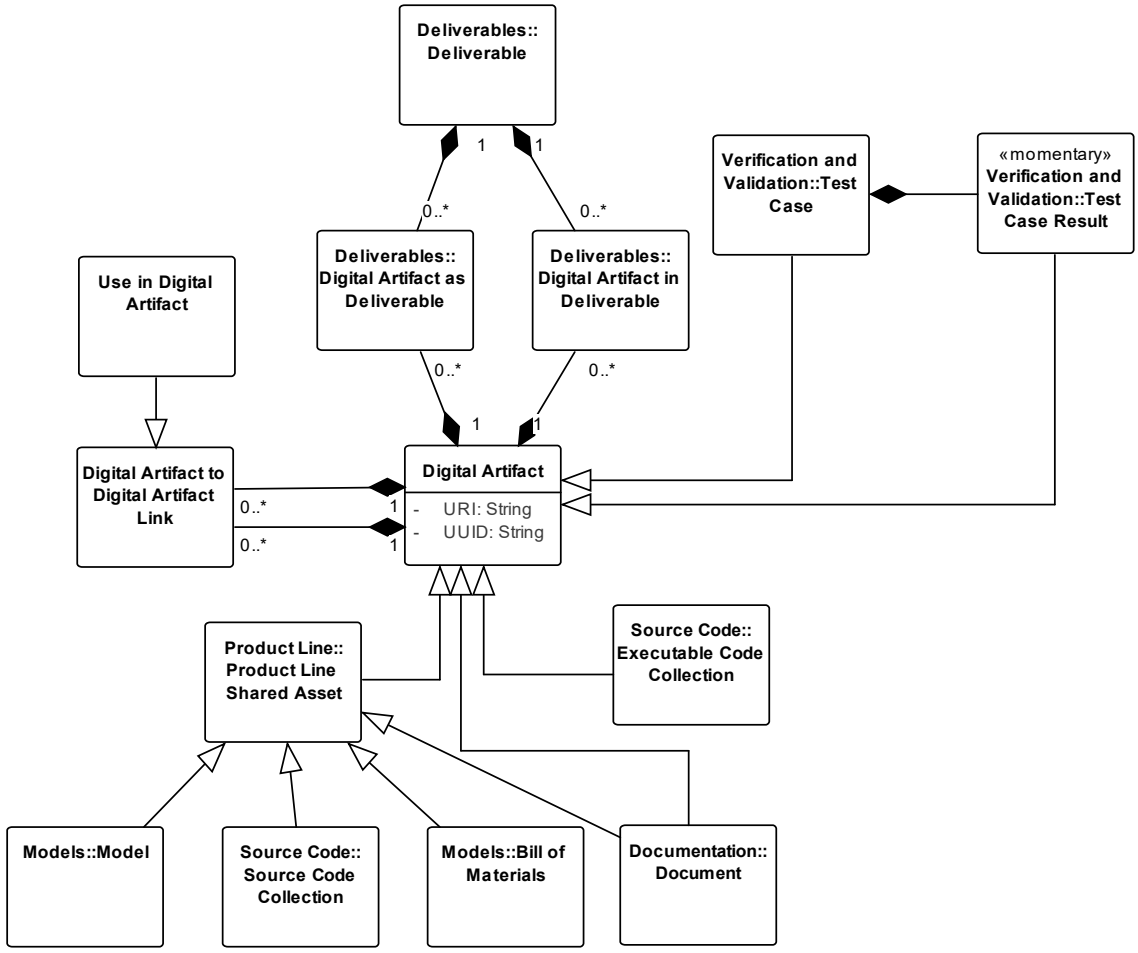

Figure 8: Digital Artifacts

Digital Artifacts::Digital Artifact

A digital artifact is a specific, unique, and immutable piece of information. A digital artifact has a fixed length and fixed internal structure.

In practice a digital artifact is analogous to a file tracked by a version control system, except that each version of the file is a distinct digital artifact.

A digital artifact can be raw data, or it can be a collection of references to other digital artifacts.

A "digital artifact" is analogous to an "object" as defined by the National Information Assurance Glossary, but with the added constraint of immutability.

Object: "Passive information system-related entity (e.g., devices, files, records, tables, processes, programs, domains) containing or receiving information. Access to an object implies access to the information it contains"

Digital Artifacts::Digital Artifact in Project

The use of a digital artifact in a project.

Digital Artifacts::Digital Artifact to Digital Artifact Link

Digital Artifacts can be linked to one another in a variety of ways.

Digital Artifacts::Use in Digital Artifact

A digital artifact can be used in another digital artifact. For example, an AADL model project digital artifact may be a list of references to individual aadl model file digital artifacts.

Deliverables

Package in package 'Digital Artifacts'

Deliverables diagram

Class diagram in package 'Deliverables'

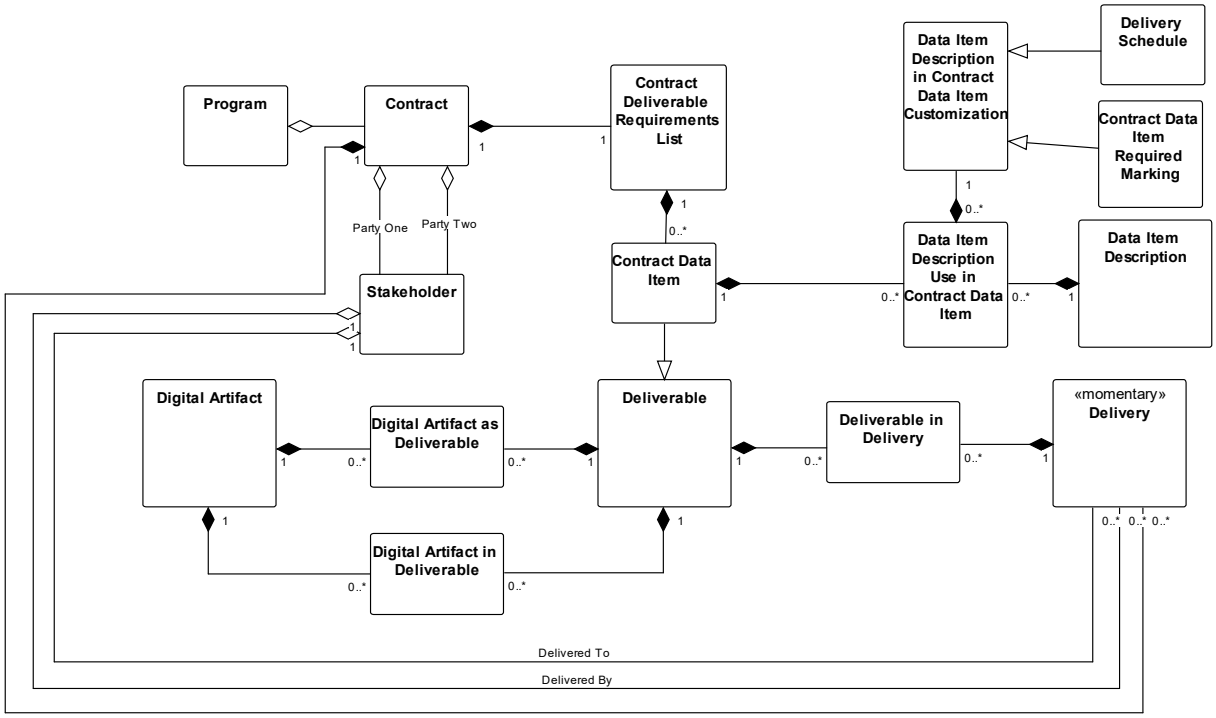

Figure 9: Deliverables

Deliverables::Contract

A legally binding agreement between two stakeholders.

From the DAU Glossary:

A mutually binding legal relationship obligating the seller to furnish supplies or services (including construction) and the buyer to pay for them.

Deliverables::Contract Deliverable Requirements List

Definition of CDRL is as follows (ref: http://acqnotes.com/acqnote/careerfields/contract-data-requirements-list-cdrl):

The Contract Data Requirements List (CDRL) is a list of authorized data requirements for a specific procurement that forms part of a contract. It is comprised of either a single DD Form 1423, or a series of DD Forms 1423 containing data requirements and delivery information. The CDRL is the standard format for identifying potential data requirements in a solicitation, and deliverable data requirements in a contract. DFAR Subpart 215.470 requires the use of the CDRL in solicitations when the contract will require delivery of data.

From the DAU Glossary:

Contract Data Requirements List (CDRL)
A DD Form 1423 list of contract data requirements that are authorized for a specific acquisition and made a part of the contract.

Deliverables::Data Item Description

A DID is a completed document that defines the data required of a contractor and is included in a CDRL. The document specifically defines the data content, format, and intended use. There are standard DIDs for all topics

reference: http://acqnotes.com/acqnote/careerfields/contract-data-requirements-list-cdrl

From the DAU Glossary:

Data Item Description
A document that specifically defines the data required of a contractor in terms of content, format and intended use.

Deliverables::Deliverable

A deliverable is a digital artifact provided to a customer according to contractual obligations.

A "deliverable" is broad in nature and can either be data or an item/HW/system etc can be defined as follows (ref. http://dictionary.sensagent.com/Deliverable/en-en/):

In the US DoD, a deliverable is any item delivered to the government under a contract, whether it is a physical product or an item of data. A “Nonseverable deliverable” means a deliverable item that is a single end product or undertaking, entire in nature, that cannot be feasibly subdivided into discrete elements or phases without losing its identity.[3]

Note: Ref 3: "Nonseverable Deliverable" is defined in DFARS 204.7101

Deliverables::Deliverable in Delivery

Used to track deliverables used as part of a delivery.

Deliverables::Delivery

A delivery is a collection of deliverables provided to a stakeholder according to terms of a contract.

Deliverables::Delivery Schedule

Delivery Schedule: Timing or rate of delivery as required by a buyer, or as agreed between a buyer and a seller, for goods or services purchased for a future delivery period.

From: http://www.businessdictionary.com/definition/delivery-schedule.html

Deliverables::Digital Artifact as Deliverable

The use of a digital artifact as a deliverable.

Deliverables::Digital Artifact in Deliverable

A digital artifact can be contained in a deliverable as well as being itself a deliverable.

Documentation

Package in package 'Digital Artifacts'

Documentation diagram

Class diagram in package 'Documentation'

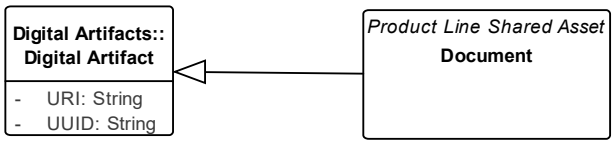

Figure 10: Documentation

Documentation::Document

A document is an unstructured, ordered collection of text and graphics, such as a Microsoft Word file.

Metadata

Package in package 'Digital Artifacts'

Digital Artifact metadata includes information about a given digital artifact, such as its classification markings.

Markings diagram

Class diagram in package 'Metadata'

As written, this model only handles marking digital artifacts. Future work should consider the need to mark individual elements within a digital artifact for redaction purposes.

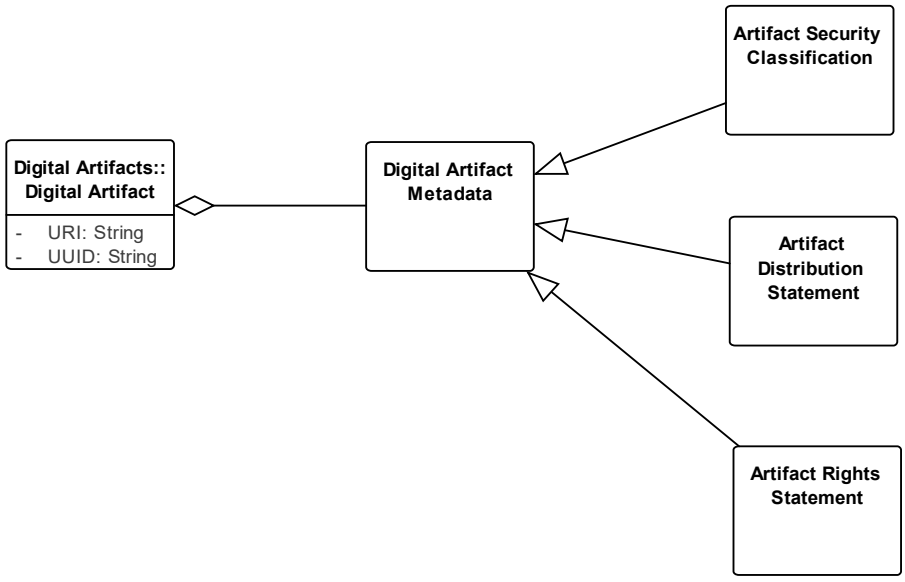

Figure 11: Markings

Metadata::Artifact Security Classification

Should reference to NIST and/or NSA guide for terminology.

Metadata::Digital Artifact Metadata

Digital artifact metadata is extra information about the digital artifact used to administer it. Metadata is used to determine access, execution, and sharing authorization.

Digital artifact metadata includes an Access Control List (ACL) as defined by the CNSS Glossary:

A list of permissions associated with an object. The list specifies who or what is allowed to access the object and what operations are allowed to be performed on the object.

Digital artifact metadata supports Role-Based Access Control, as defined by CNSS Glossary

"Role-Based Access Control (RBAC) Access control based on user roles (i.e., a collection of access authorizations a user receives based on an explicit or implicit assumption of a given role). Role permissions may be inherited through a role hierarchy and typically reflect the permissions needed to perform defined functions within an organization. A given role may apply to a single individual or to several individuals.
Source: NIST SP 800-53 Rev 4

Models

Package in package 'Digital Artifacts'

This package broadly describes models, encompassing a variety of model types relevant to ASoT including AADL, SysML, solid models, etc.

Models diagram

Class diagram in package 'Models'

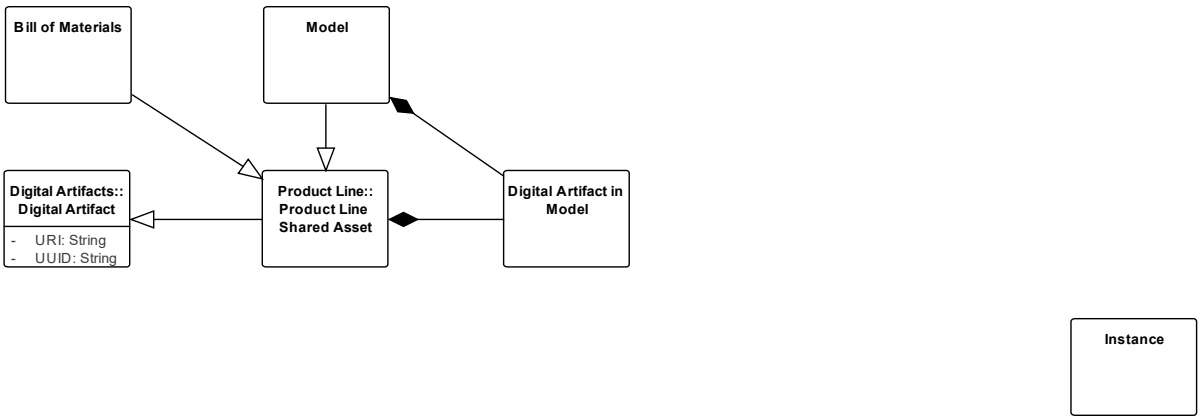

Figure 12: Models

Models::Bill of Materials

Bill of Material: A formal statement acquired from a contractor that specifies and describes the quantities of materials required for the production or manufacture of a product, assembly or subassembly.

from Department of Defense Instruction, April 25, 1972:
https://biotech.law.lsu.edu/blaw/dodd/corres/pdf/i42108_042572/i42108p.pdf

In the acquisition world there is another Bill of Material term called the "Consolidated Bill of Material (CBOM)" which is defined as a consolidated priced summary of individual material quantities included in the tasks or contract line items (CLINs) being proposed and provides the basis for pricing (vendor quotes, invoice prices, etc). CBOM is a more current and widely used term than BOM in DOD as it is also referenced in FAR and DFAR.

When evaluating the proposal for adequacy the evaluator uses the proposal adequacy checklist found in DFARS 252.215-7009. One of the items on the adequacy checklist is a CBOM as specified in FAR 15.408 Table 15-2 Section II.A which must
Provide a consolidated priced summary of individual material quantities included in the various tasks, orders, or contract line items being proposed and the basis for pricing (vendor quotes, invoice prices, etc.). Include raw materials, parts, components, assemblies, and services to be produced or performed by others. For all items proposed, identify the item and show the source, quantity, and price… These requirements also apply to all subcontractors if required to submit certified cost or pricing data".

https://govcon360.com/2017/01/delivering-an-adequate-consolidated-bill-of-materials-in-your-proposal/

Models::Digital Artifact in Model

A model may consist of several shared assets (for example, multiple .aadl files).

Models::Model

Via https://www.msco.mil/MSReferences/Glossary/TermsDefinitionsI-M.aspx

A physical, mathematical, or otherwise logical representation of a system, entity, phenomenon, or process. (DoDI 5000.61, DoDI 5000.70)

Source Code

Package in package 'Digital Artifacts'

Source Code diagram

Class diagram in package 'Source Code'

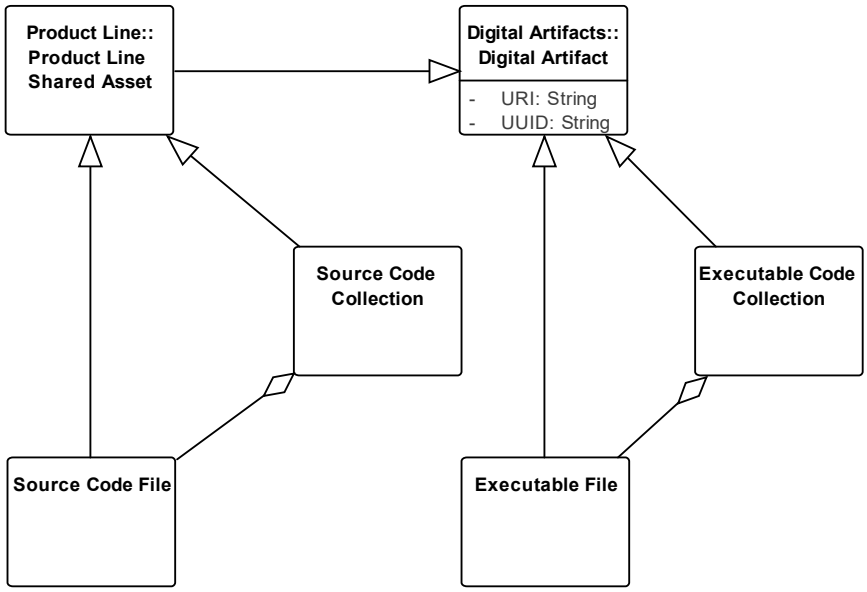

Figure 13: Source Code

Source Code::Executable Code Collection

An executable code collection is a digital artifact that can be executed (either directly or using an interpreter) by a computer.

The contents of the collection can be references to other digital artifacts or a single piece of information (such as a .zip file).

Traceability

Package in package 'Digital Artifacts'

Traceability refers to the capability of an Authoritative Source of Truth to relate digital artifacts to one another.

Traceability relationships between digital artifacts support the concept of a "Digital Thread" of information connected through various phases of project develoment.

https://www.nist.gov/programs-projects/digital-thread-smart-manufacturing

Equivalence Traceability diagram

Class diagram in package 'Traceability'

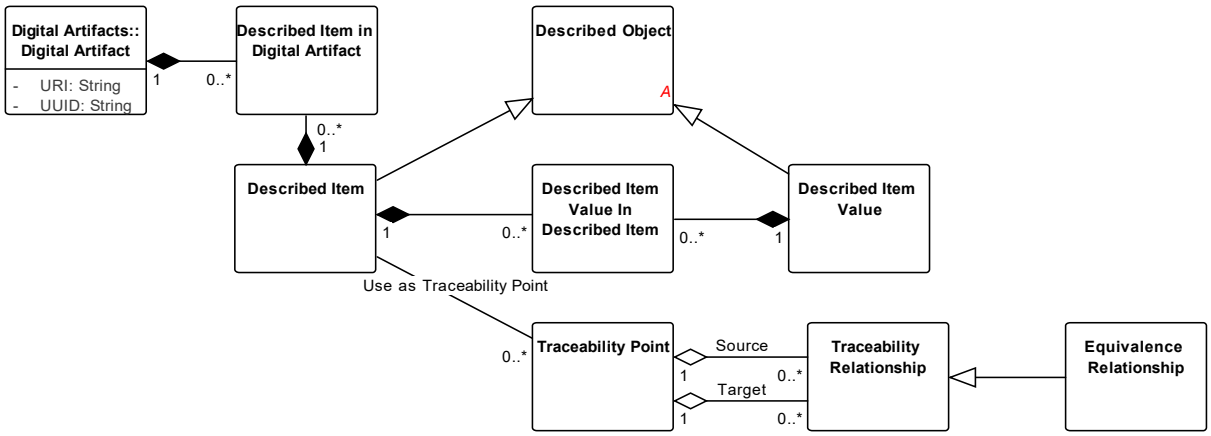

Figure 14: Equivalence Traceability

Traceability diagram

Class diagram in package 'Traceability'

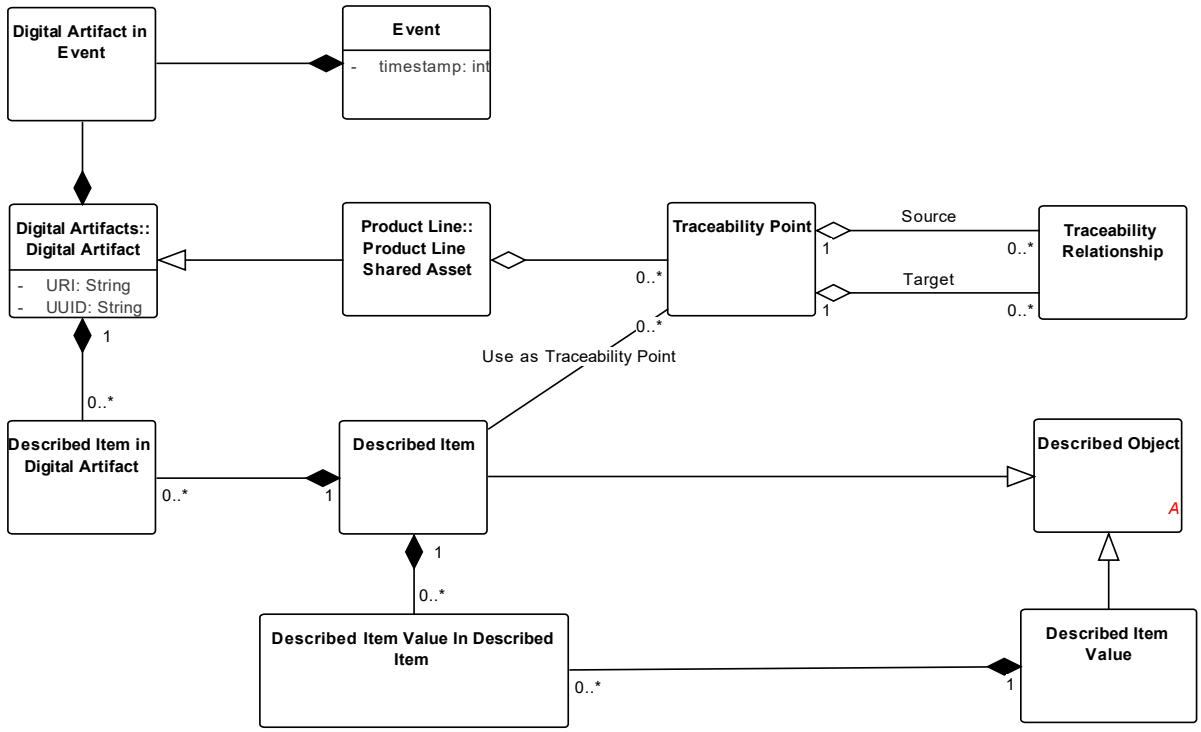

Figure 15: Traceability

Traceability::Described Item

A Described Item is an organizational concept used to construct graphs of Described Objects.

See SAVI Behavior Model Consistency Analysis
Mike Kerstetter, SAVI PM Kurt Woodham, NASA LaRC

Traceability::Described Item Value

A Described Item Value is an organizational concept used to construct graphs of Described Objects.

See SAVI Behavior Model Consistency Analysis
Mike Kerstetter, SAVI PM Kurt Woodham, NASA LaRC

Traceability::Described Item Value In Described Item

A relationship between two Described Objects

Traceability::Described Item in Digital Artifact

A given Digital Artifact may describe many things.

Traceability::Described Object

A Described Object is an abstract description of some thing independent of the representation of that thing. A Described Object may have representations in several different forms.

See SAVI Behavior Model Consistency Analysis
Mike Kerstetter, SAVI PM Kurt Woodham, NASA LaRC

Traceability::Event

An event is a record of a momentary state of one or more digital artifacts.

For example, creation and deletion of a digital artifact are both events. A change to a digital artifact is an event that has pointers to two digital artifacts (the old and the new).

Note that a single digital artifact has no concept of a "version" so a changed digital artifact is a new digital artifact.

Equivalence

Package in package 'Traceability'

Equivalence diagram

Class diagram in package 'Equivalence'

This diagram is based on a presentation from the 2017 INCOSE International Workshop,

System Architecture Virtual Integration (SAVI) Project :
Intermodel Error Checking and Consistency Review

and Demonstration
An Aerospace Vehicle Systems Institute Project (AVSI)
Presented by Greg Pollari (Rockwell Collins) and Nigel Shaw (Eurostep)

www.incose.org/IW2017

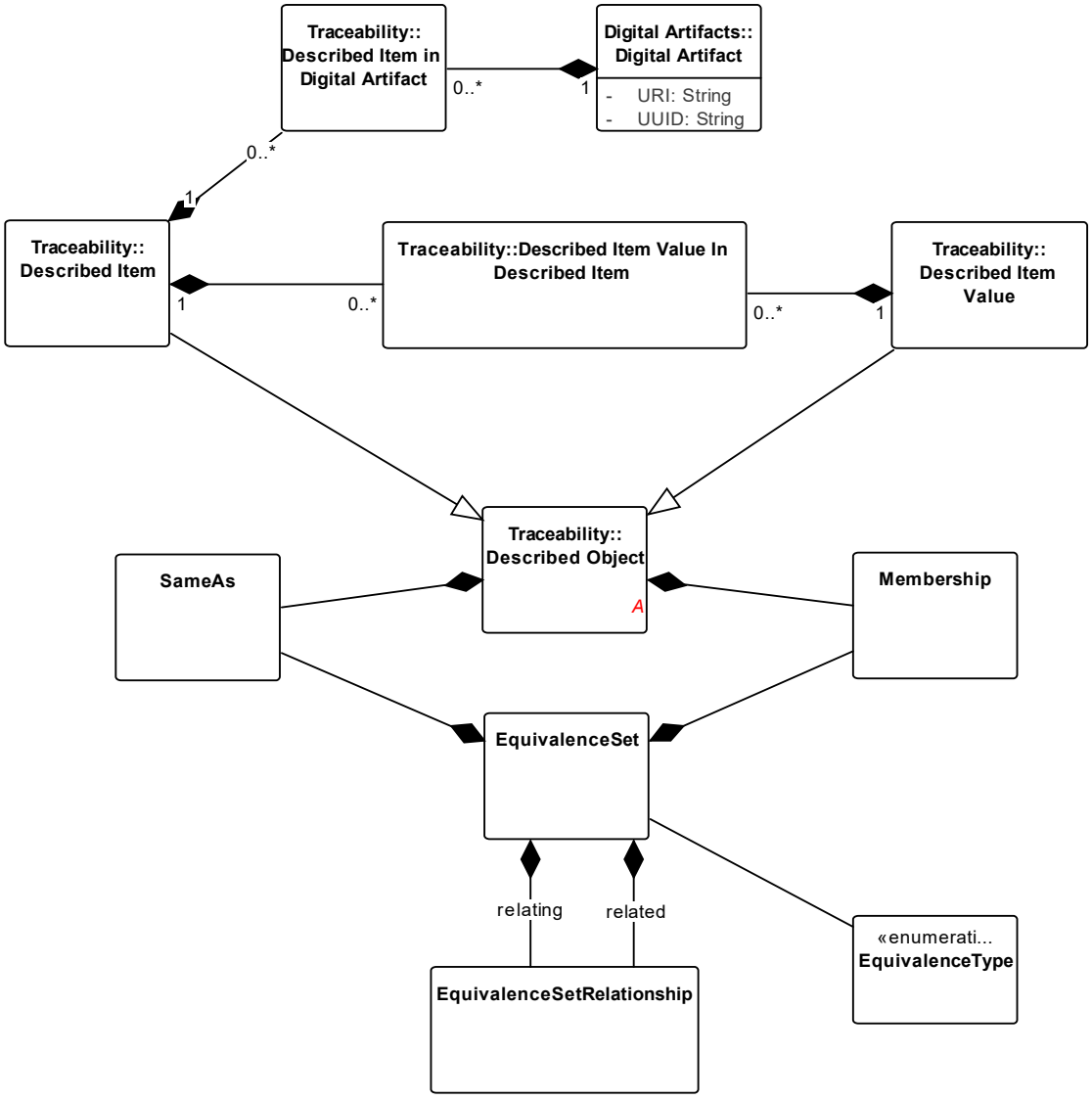

Figure 16: Equivalence

Equivalence::EquivalenceSet

For future work, it may be possible to reuse the simpler traceability relationship concept.

Equivalence::Membership

A DescribedObject's membership in an equivalence set.

Equivalence::SameAs

A DescribeObject's equivalence to an EquivalenceSet.

Version Control

Package in package 'Digital Artifacts'

This package describes terminology for tracking digital artifacts over time. The terminology in this package is largely based on existing version control systems such as git.

Version Control diagram

Class diagram in package 'Version Control'

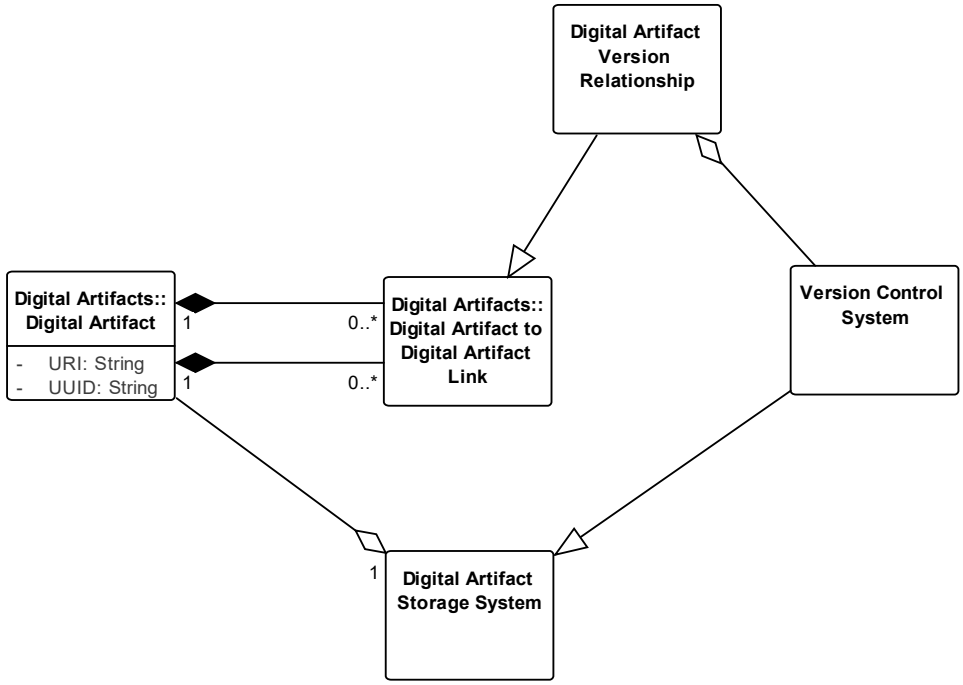

Figure 17: Version Control

Version Control::Digital Artifact Storage System

A digital artifact storage system is any system capable of storing digital artifacts. A network share is an example of a digital artifact storage system.

Version Control::Digital Artifact Version Relationship

A Digital Artifact Version Relationship describes the relationship between two digital artifacts, such as the change that occurred between two revisions of a model.

Version Control::Version Control System

A version control system is a tool used to track digital artifacts over time. Git and Subversion are common examples.

Glossary

Package in package 'ASoT Ontology'

Tracking terms and definitions is a capability that may be required of an ASoT. This package defines the terms used to describe a glossary.

Glossary diagram

Class diagram in package 'Glossary'

An ASoT should have its own glossary or ontology.

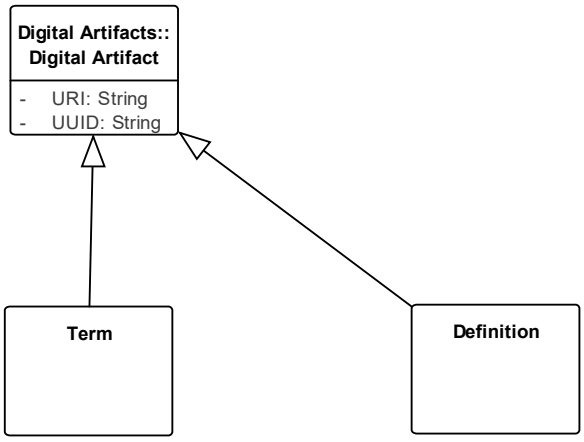

Figure 18: Glossary

Infrastructure

Package in package 'ASoT Ontology'

This package describes the support infrastructure of the ASoT required to meet non-functional requirements.

Infrastructure diagram

Class diagram in package 'Infrastructure'

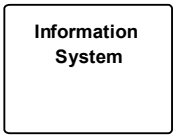

Figure 19: Infrastructure

Infrastructure::Information System

As defined by the CNSS Glossary:

A discrete set of information resources organized for the collection, processing, maintenance, use, sharing, dissemination, or disposition of information.
Source: 44 U.S.C. Sec 3502
Note: Information systems also include specialized systems such as industrial/process controls systems, telephone switching and private branch exchange (PBX) systems, and environmental control systems.

Issue Tracking

Package in package 'ASoT Ontology'

This package describes issue tracking capabilities. Its contents are based on existing industry tools such as Github, Redmin, Trac, and Jira.

Issue Tracking diagram

Class diagram in package 'Issue Tracking'

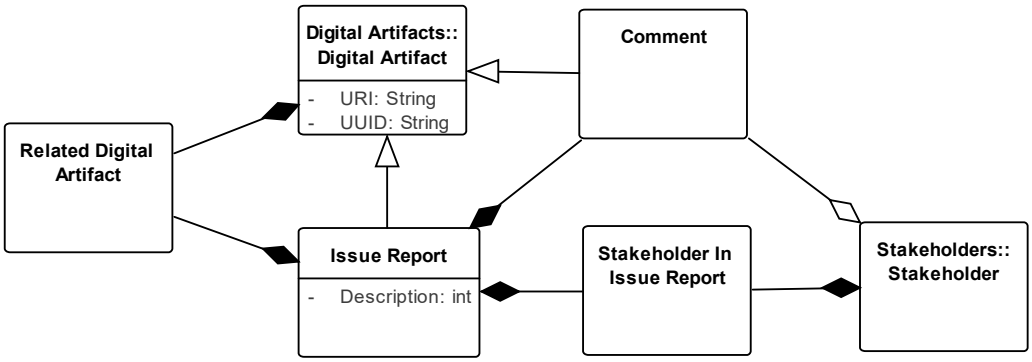

Figure 20: Issue Tracking

Management

Package in package 'ASoT Ontology'

This package describes terminology for accessing and modifying data in an ASoT.

Management diagram

Class diagram in package 'Management'

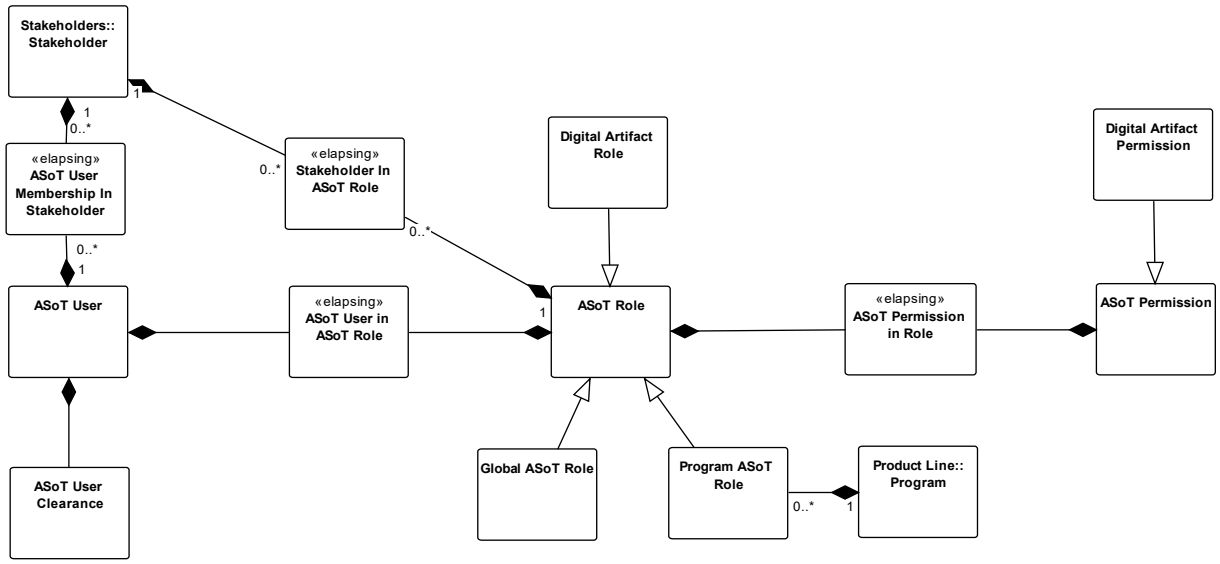

Figure 21: Management

Management::ASoT Permission

An ASoT Permission is a description of a single action on the ASoT or on digital artifacts managed by the ASoT.

Management::ASoT Role

An ASoT Role is a Stakeholder or ASoT user's tasking in a given context. A single ASoT User or Stakeholder may have multiple ASoT Roles.

An ASoT Role may be specific to a given program, or it may be global (applying to all programs managed by the ASoT).

An ASoT Role may be associated with certain ASoT permissions, with ASoT Business Processes, or other ASoT features.

From National Information Assurance (IA) Glossary

https://www.hsdl.org/?view&did=7447

Role: "A group attribute that ties membership to function. When an entity assumes a role, the entity is given certain rights that belong to that role. When the entity leaves the role, those rights are removed. The rights given are consistent with the functionality that the entity needs to perform the expected tasks."

Management::ASoT User

Analogous to common usage of 'user' in an information system. Unlike a stakeholder, an ASoT user refers to a specific individual.

The acronym "ASoT" is added to avoid collisions with other uses of "user."

From the CNSS Glossary:

1. Individual, or (system) process acting on behalf of an individual, authorized to access an information system.
Source: NIST SP 800-53 Rev 4
2. An individual who is required to use COMSEC material in the performance of his/her official duties and who is responsible for safeguarding that COMSEC material. See hand receipt holder and local element.
Source: CNSSI No. 4005 (COMSEC); NSA/CSS Manual Number 3-16 (COMSEC)

Management::ASoT User Clearance

An ASoT User's clearance refers to the attributes of the user that indicate what information the user is or is not allowed to view.

Management::ASoT User Membership In Stakeholder

An ASoT user can be a member of one or more stakeholders.

Management::Digital Artifact Role

A Digital Artifact Role is a specialization of ASoT Role referring to the relationship between an ASoT User or Stakeholder and a digital artifact.

Digital Artifact Roles such as "Owner" or "Author" may have standard sets of ASoT Permissions.

Management::Global ASoT Role

Global ASoT Roles apply across all programs managed by the ASoT. Global ASoT Roles are expected to be reserved for ASoT Users or Stakeholders that own or manage the ASoT infrastructure.

Management::Program ASoT Role

A Program ASoT Role is an ASoT Role specific to a single program. A given Stakeholder may have different ASoT Roles on different Programs.

Management::Stakeholder In ASoT Role

A given stakeholder may fill multiple ASoT Roles concurrently or over time.

Management::Use in Model

From INCOSE:

Models
When models are used in product development, they can be developed as supersets and instrumented with variation points for the product line. For example, system design or architecture models using SysML or UML can be instrumented through variation points, which apply to structural elements such as processes, objects, classes, states, use-cases, packages, and others.

Meta

Package in package 'ASoT Ontology'

This package contains reference information about the ontology itself.

Meta diagram

Class diagram in package 'Meta'

This model describes the concept of an Authoritative Source of Truth.

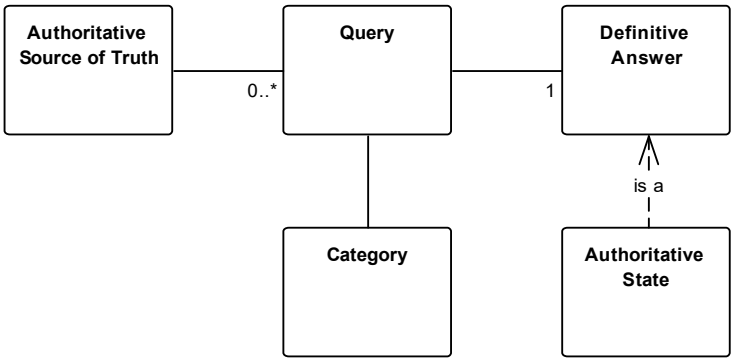

Figure 22: Meta

Ontology diagram

Class diagram in package 'Meta'

This diagram lays out the facets of an ontology.

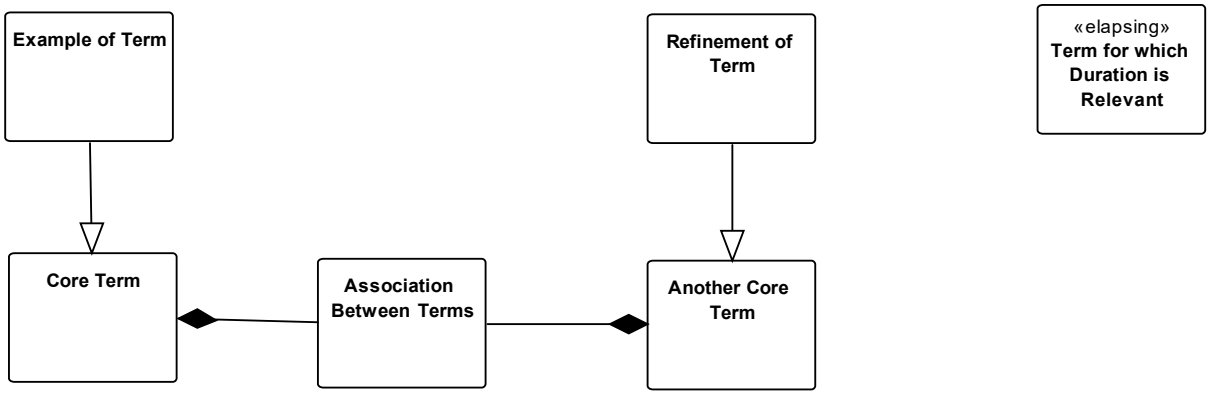

Figure 23: Ontology

Meta::Authoritative Source of Truth

An Authoritative Source of Truth is a capability that gives definitive answers to queries about a collection of systems.

Reference:
https://www.msco.mil/MSReferences/Glossary/TermsDefinitionsA-B.aspx

From OMG Wiki:

The authoritative source of truth for a digital artifact serves as the primary means of ensuring the credibility and coherence of the digital artifact that its creators share with a variety of stakeholders. It gives stakeholders from diverse organizations and distributed locations the authorization to access, analyze, and use valid digital artifacts from an authoritative source. The owners of digital environments or the community for digital engineering ecosystems provides stakeholders with an authoritative source of truth that assures confidence in the quality of the digital artifact across disciplines, domains, and life cycle phases.

https://www.omgwiki.org/MBSE/doku.php?id=mbse:authoritative_source_of_truth

Meta::Authoritative State

The collection of digital artifacts and relationships between those artifacts currently approved for a system of interest.

Example: the currently approved configuration for a component. Previously approved but replaced configurations are no longer part of the Authoritative State.

Think of this state like the HEAD for the master branch (git).

Meta::Category

Categories are functional groupings of queries based on typical tool and process organization.

Meta::Core Term

A Core Term is important and commonly used in a problem domain.

Meta::Definitive Answer

Definitive answers are answers governed by organizational processes.

Meta::Example of Term

Terminology examples are critical for validation of an ontology.

Meta::Query

Queries are questions about the past, present, or future state of the design or implementation of a collection of systems.

Product Line

Package in package 'ASoT Ontology'

The product line package describes the organization of programs, projects, and products. It is based largely on the INCOSE product line management guidelines.

Product Line diagram

Class diagram in package 'Product Line'

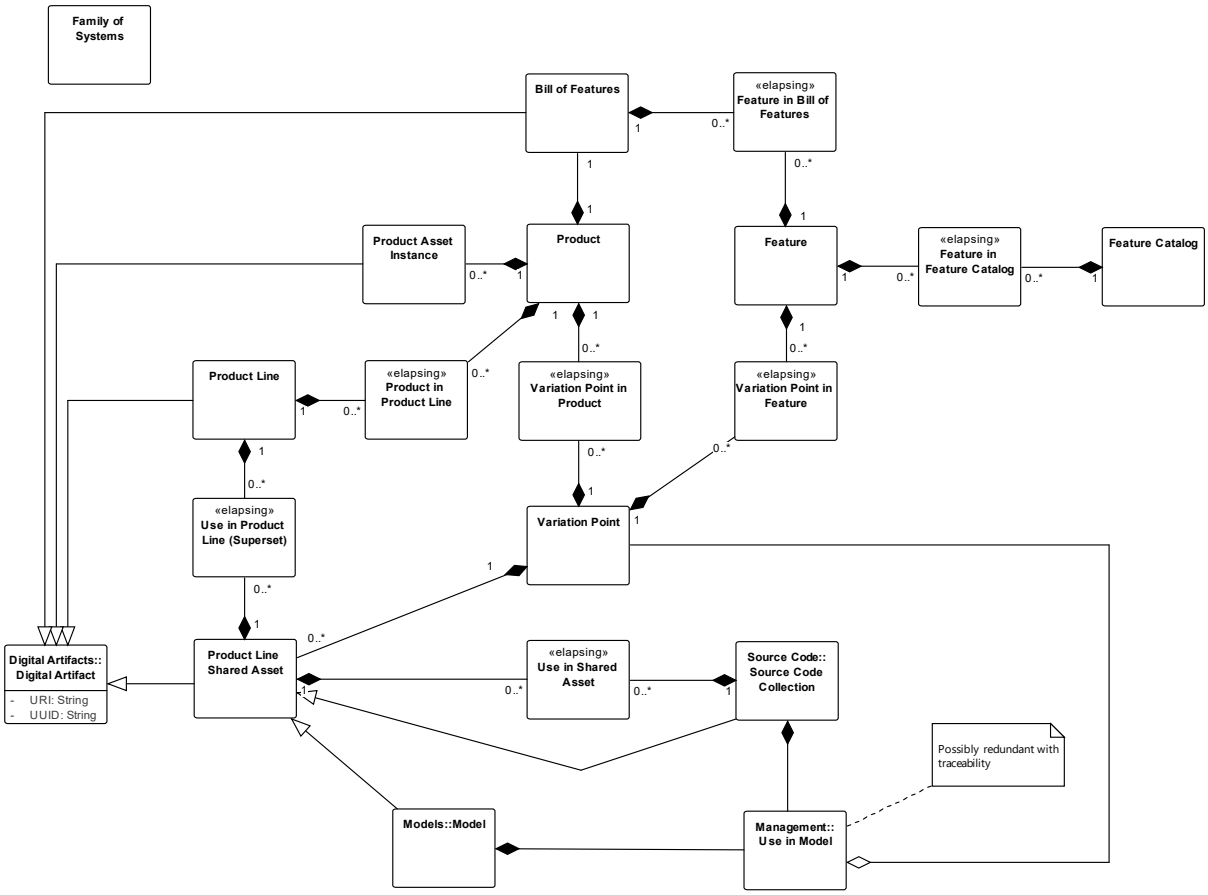

Figure 24: Product Line

Program Management diagram

Class diagram in package 'Product Line'

This diagram describes the relationship between a program and a project.

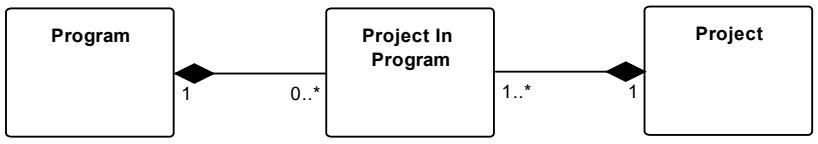

Figure 25: Program Management

Testing diagram

Class diagram in package 'Product Line'

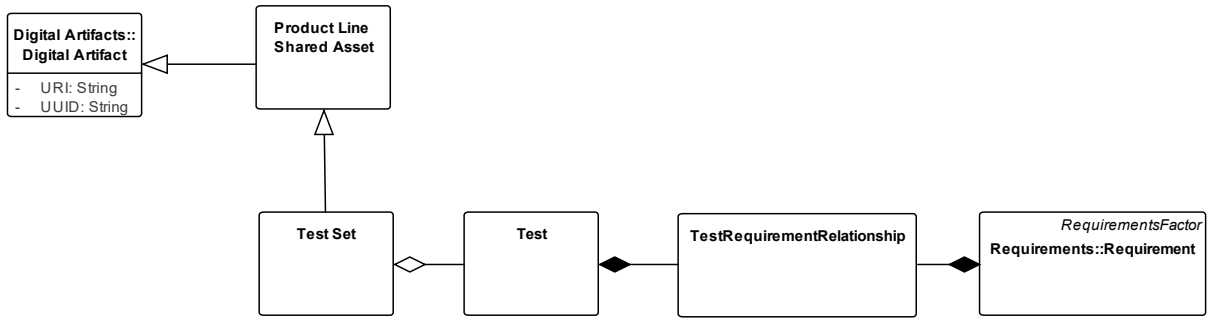

Figure 26: Testing

Product Line::Bill of Features

From INCOSE:

The features selected for each product in the product line are specified in the Bill-of-Features for that Product.

Product Line::Feature

A feature is a capability or component that can be included in a given product.

From INCOSE:

A feature is a distinguishing characteristic that describes how the members of the product line differ from each other.

Product Line::Feature Catalog

From INCOSE PLE Primer:

The Feature Catalogue captures the features that are available for each product to select.

Product Line::Product

A product is a not-yet-customized description of an item to be created. The definition used here differs from the DAU definition of Product in that it distinguishes the general product (e.g., Ford F150) from a specific Product Asset Instance (e.g., John's Ford F150 configured with his specific paint color and trim options).

Product Line::Product Asset Instance

From INCOSE:

The PLE Factory Configurator is an automated software tool that applies a Bill-of-Features to all of the shared assets. It evaluates each variation point to determine if that variation point's content should be included or not.

The PLE Factory produces as output Product Asset Instances, each one containing only the shared asset content suited for one of the products in the product line. Together, they constitute the artifact set for one of the products in the product line.

"Product Asset Instance" is analogous to "Product" as defined by the DAU:

The DAU Definition of Product is

Product
1.) The result of Research, Development, Test, and Evaluation (RDT&E) in terms of hardware or software being produced (manufactured). Also known as an end item. 2.) The item stipulated in a contract to be delivered under the contract (i.e., service, study, or hardware).

Product Line::Product Line

Family of Systems is a Synonym.

Product Line::Product Line Shared Asset

A shared asset is a single digital artifact that contains variation points. In practice a shared asset may be a digital artifact that is composed of other digital artifacts.

From INCOSE:

Shared assets are artifacts that support the creation, design, implementation, deployment, and operation of products. They can be represented digitally and configured, and are shared across the product line.

Each shared asset is a superset that contains variation points, which are pieces of content that should be included in or omitted from a product based on the features selected for that product.

Product Line::Program

As used in the ASoT ontology, "Program" refers to a top-level organizational concept, typically with its own funding and staffing. The ASoT ontology does not assume all programs to be DoD programs.

From the DAU Glossary:

Program
1.) A DoD acquisition program. 2.) As a verb, program means to schedule funds to meet requirements and plans. 3.) A major, independent part of a software system. 4.) A combination of Program Elements (PEs) designed to express the accomplishment of a definite objective or plan.

Product Line::Project

A project is a means for organization of digital artifacts and business processes within a program.

Product Line::Project In Program

A program does not need to have projects, but a project must be in at least one program.

Product Line::Use in Shared Asset

From INCOSE:

In software systems, shared assets can include the source code, the models used to generate the source code, build files, and automated unit tests. Source code can be configured in several ways including individual blocks of code, files, or build files. A modular software architecture may assist greatly with the feature instrumentation but is not necessarily required. Features might align closely with the system's software, electrical, and/or mechanical modularity, or they might be cross-cutting where a feature affects several different modules.

Product Line::Variation Point

From INCOSE:

The shared asset supersets contain variation points, which are places in the asset that denote content that is configured according to the feature choices of the product being built. A statement of the product's distinguishing characteristics — its features — is applied to "exercise" these variation points (that is, cause the content associated with each variation point to be configured to meet the needs of the product).

Requirements

Package in package 'ASoT Ontology'

The terminology used in this package is based largely on the capability of IBM's DOORS.

Requirements diagram

Class diagram in package 'Requirements'

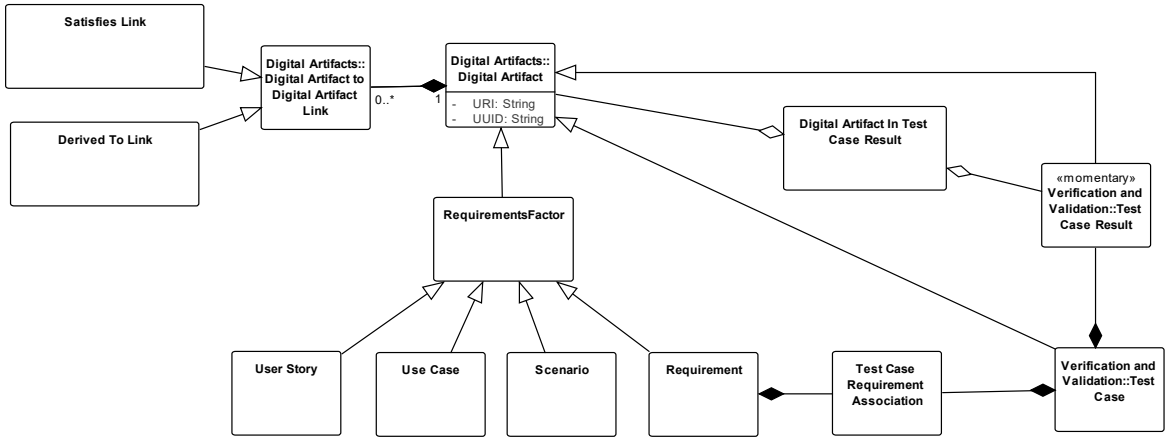

Figure 27: Requirements

Test and Digital Artifact diagram

Class diagram in package 'Requirements'

A subset of the requirements package showing the relationship between a test and a digital artifact.

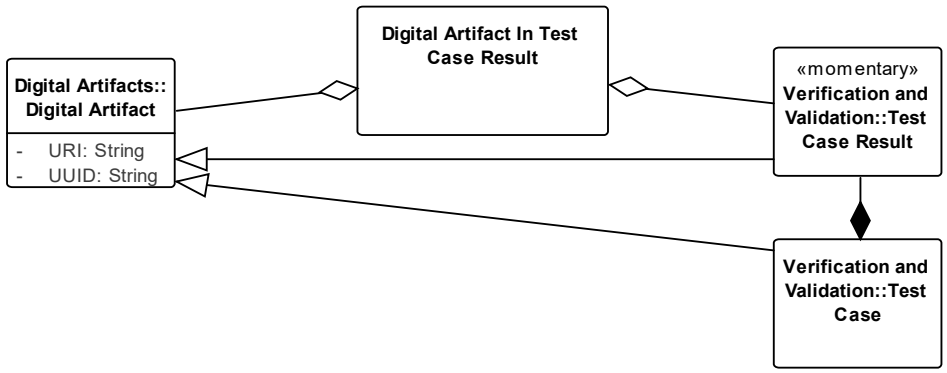

Figure 28: Test and Digital Artifact

Requirements::Digital Artifact In Test Case Result

A test case result can be associated with zero or more digital artifacts that contributed to the result.

Requirements::Requirement

From https://sebokwiki.org/wiki/Requirement_(glossary)

Statement that identifies a product* or process operational, functional, or design characteristic or constraint, which is unambiguous, testable or measurable, and necessary for product or process acceptability. (ISO/IEC 2007)

Safety

Package in package 'ASoT Ontology'

This section of the ontology is a reference to the core artifacts of safety analysis as defined by MIL-STD-882E.

Safety diagram

Class diagram in package 'Safety'

Execution of a Safety Plan should be associated with Described Objects in Digital Artifacts.

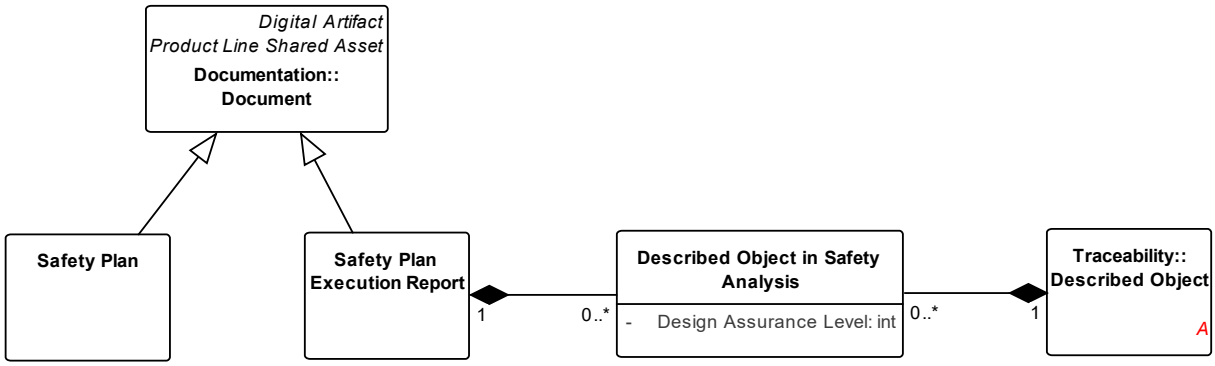

Figure 29: Safety

Safety::Safety Plan

A Safety Plan describes the strategy to be used to assure system safety.

Safety is defined in MIL-STD-882E:

3.2.30 Safety. Freedom from conditions that can cause death, injury, occupational illness, damage to or loss of equipment or property, or damage to the environment.

Stakeholders

Package in package 'ASoT Ontology'

A variety of entities have a vested interest in an ASoT. This package describes them and their relationships.

Stakeholders diagram

Class diagram in package 'Stakeholders'

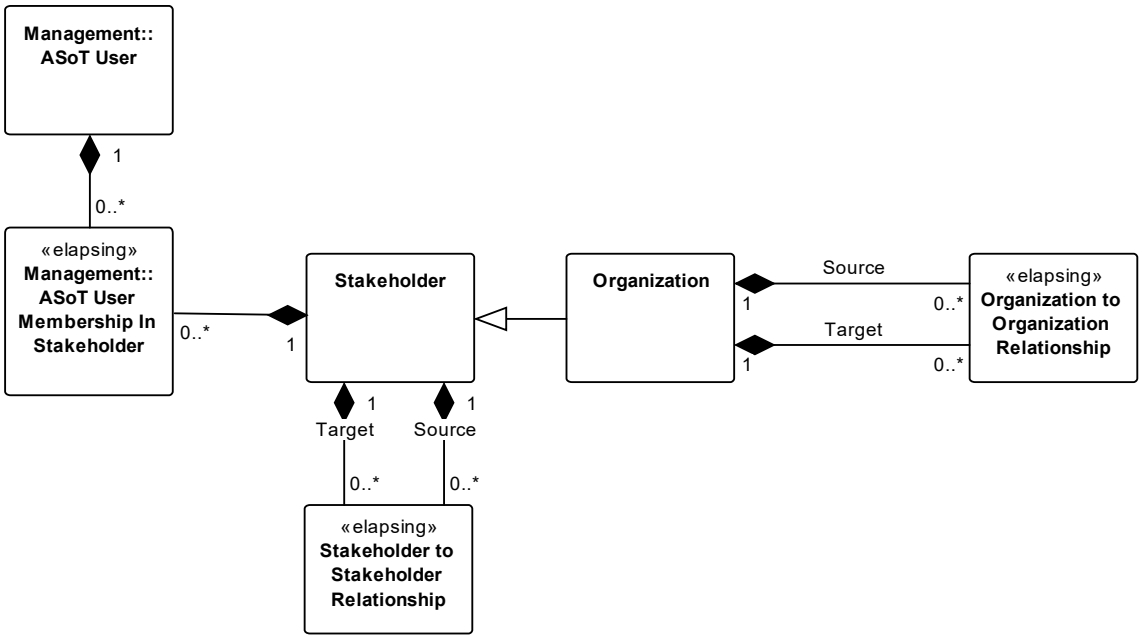

Figure 30: Stakeholders

Stakeholders and Programs diagram

Class diagram in package 'Stakeholders'

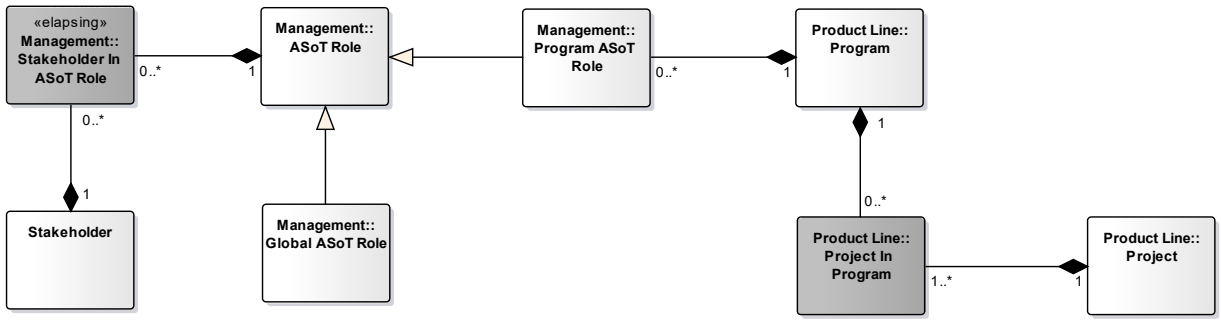

Figure 31: Stakeholders and Programs

Stakeholders::Organization

An organization is a type of stakeholder.

From the CNSS Glossary

An entity of any size, complexity, or positioning within an organizational structure (e.g., a federal agency, or, as appropriate, any of its operational elements). See enterprise.
Source: NIST SP 800-37 Rev 1

From https://sebokwiki.org/wiki/Organization_(glossary)

A group of people and facilities with an arrangement of responsibilities, authorities and relationships. (INCOSE 2011)

Stakeholders::Organization to Organization Relationship

Organizations can be related to either, for example as parent/child organizations.

Stakeholders::Stakeholder

A stakeholder is an organization associated in some way with the project managed by the ASoT. An ASoT User can be part of multiple stakeholders.

From https://www.sebokwiki.org/wiki/Stakeholder_(glossary)

(1) Individual or organization having a right, share, claim, or interest in a system or in its possession of characteristics that meet their needs and expectations (ISO/IEC/IEEE 2015)
(2) Individual or organization having a right, share, claim, or interest in a system or in its possession of characteristics that meet their needs and expectations; N.B. Stakeholders include, but are not limited to end users, end user organizations, supporters, developers, producers, trainers, maintainers, disposers, acquirers, customers, operators, supplier organizations and regulatory bodies. (ISO/IEC June 2010)
(3) An individual, team, or organization (or classes thereof) with interests in, or concerns relative to, a system. (ISO/IEC 2007)
(4) A stakeholder in an organisation is (by definition) any group or individual who can affect or is affected by the achievement of the organisation's objectives. (Freeman 1984)

Stakeholders::Stakeholder to Stakeholder Relationship

Stakeholders may have a variety of relationships including teammates, partners, sub-contractors, etc.

Verification and Validation

Package in package 'ASoT Ontology'

This package provides terminology for verification and validation processes.

Verification and Validation diagram

Class diagram in package 'Verification and Validation'

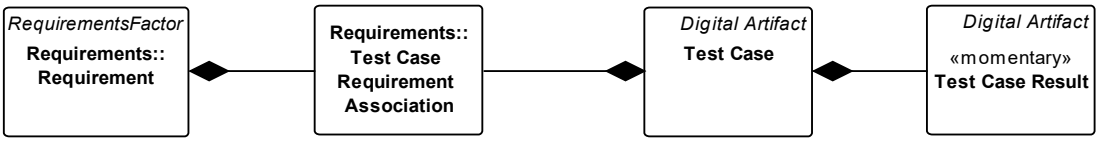

Figure 32: Verification and Validation

References

[1] N. Aizenbud-Reshef, B. T. Nolan, J. Rubin, and Y. Shaham-Gafni. Model traceability. *IBM Systems Journal*, 45(3):515–526, 2006.

[2] Jim Amsden. *OSLC Architecture Management Specification 2.1. Part 1: Specification*, http://docs.oasis-open.org/oslc-domains/oslc-am/v2.1/cs01/part1-architecture-management-spec/oslc-am-v2.1-cs01-part1-architecture-management-spec.html edition.

[3] Jim Amsden. *OSLC Architecture Management Specification 2.1. Part 2: Vocabulary*, http://docs.oasis-open.org/oslc-domains/oslc-am/v2.1/oslc-am-v2.1-part2-architecture-management-vocab.html edition.

[4] Jim Amsden. *OSLC Core Version 3.0. Part 1: Overview*, http://docs.oasis-open.org/oslc-core/oslc-core/v3.0/oslc-core-v3.0-part1-overview.html edition.

[5] Jim Amsden. *OSLC Core Version 3.0. Part 2: Discovery*. OASIS, http://docs.oasis-open.org/oslc-core/oslc-core/v3.0/oslc-core-v3.0-part2-discovery.html edition.

[6] Jim Amsden. *OSLC Core Version 3.0. Part 6: Resource Shape*. OASIS, http://docs.oasis-open.org/oslc-core/oslc-core/v3.0/oslc-core-v3.0-part6-resource-shape.html edition.

[7] US Army Combat Capabilities Development Command (DEVCOM) Aviation and Missle Center (A&MC) Aviation Engineering Directorate (AED). Army military airworthiness certification criteria (amacc). Technical report, US Army, Redstone Arsenal, March 2019.

[8] US Army Combat Capabilities Development Command (DEVCOM) Aviation and Missile Center (AvMC). Comprehensive architecture strategy. august 2018.

[9] Tim Berners-Lee. Linked data. 2006.

[10] The FACE Consortium. The open group face(tm) contract guide: Guidance in writing solicitations and proposals with face requirements, version 2.0. September 2018.

[11] Steve Speicher Dave Johnson. *OSLC Core Specification 2.0. Part 2: Vocabulary*, http://open-services.net/bin/view/Main/OslcCoreSpecification edition.

[12] Jad El-khoury. An analyis of the oasis oslc integration standard for a cross-disciplinary integrated development environment. http://kth.diva-portal.org/smash/record.jsf?pid=diva2%3A1427580&dswid=-5551, April 2020.

[13] The Open Group. Guidance in writing solicitations and proposals with FACE^{TM} requirements, version 2.0. Technical report, The Open Group, 2018.

[14] INCOSE. Incose overview. web. https://www.incose.org/about-systems-engineering/system-and-se-definition/system-and-se-definitions.

[15] Software Engineering Institute. Architecture-centric virtual integration process (acvip) handbooks — acquisition and management handbook with the architecture analysis and design language (aadl). Technical report, Carnegie Mellon University, 2018.

[16] Mike Kerstetter and Kurt Woodham. Savi behavior model consistency analysis. In *Global Product Data Interoperability Summit*, 2016.

[17] No Magic. Oslc api.

[18] Tom McDermott, Paul Collopy, Chris Paredis, and Molly Nadolski. Enterprise system-of-systems model for digital-thread enabled acquisition. Technical Report SERC-2018-TR-109, Systems Engineering Research Center, July 2018.

[19] Modeling and Simulation Coordination Office. M&S Glossary. web, Accessed 10/29/2019. https://www.msco.mil/MSReferences/Glossary/MSGlossary.aspx.

[20] Baiju Muthukadan. *Selenium with Python*. https://selenium-python.readthedocs.io/, accessed 28 January 2021, 2018.

[21] Department of Defense. Department of defense instruction: Dod modeling and simulation (m&s) verification, validation, and accreditation. December 2009.

[22] Department of Defense. Electromagnetic spectrum data sharing. *DoD CIO*, August 2011.

[23] Department of Defense. Sharing data, information, and information technology (it) services in the department of defense. *DoD CIO*, August 2013.

[24] Department of Defense Open Systems Architecture Data Rights Team. Dod open systems architecture (osa) contract guidebook for program managers. Technical report, U.S. Department of Defense, June 2013.

[25] Office of the Deputy Assistant Secretary of Defense for Systems Engineering. Department of defense digital engineering strategy. June 2018.

[26] Office of the Under Secretary of Defense for Acquisition and Sustainment (OUSD(A&S)). Contracting considerations for agile solutions, version 1.0. Technical report, US Department of Defense, November 2019.

[27] OMG. Authoritative source of truth. web.

[28] OMG. Digital artifact. web.

[29] Committee on National Security Systems. National information assurance (ia) glossary. web, April 2010. https://www.dni.gov/files/NCSC/documents/nittf/CNSSI-4009_National_Information_Assurance.pdf.

[30] Committee on National Security Systems. National information assurance (ia) glossary. April 2010. https://www.hsdl.org/?view&did=7447.

[31] OUSD(A&S). Contracting considerations for agile solutions. Nov 2019.

[32] Greg Pollari and Nigel Shaw. Mbse multi-model, multi-domain interoperability. In *Global Product Data Interoperability Summit*, 2017.

[33] Kenneth Reitz. *Requests-OAuthlib: OAuth for Humans*. https://requests-oauthlib.readthedocs.io/en/latest/, accessed 28 January 2021, 2014.

[34] Mark Schulte and Jad El-khoury. *OSLC Requirements Management Version 2.1. Part 2: Vocabulary*. OASIS, http://docs.oasis-open.org/oslc-domains/oslc-rm/v2.1/oslc-rm-v2.1-part2-requirements-management-vocab.html edition.

[35] Tyler Smith, Rand Whillock, Robert Edman, Bruce Lewis, and Steve Vestal. Lessons learned in inter-organization virtual integration. In *Aerospace Systems and Technology Conference*, 2018.

[36] Steve Speicher et al. Oslc primer. May 2019.

[37] Steve Vestal and Hazel Shackleton. Sysml and aadl profile and modeling guidelines, 2020.

[38] W3C. Owl 2 web ontology language document overview, December 2013.

Acronyms

AAAF Army Aviation Architecture Framework

AADL Architecture Analysis and Design Language

ACVIP Architecture-Centric Virtual Integration Process

AIPD Architecture Implementation Process Demonstrations

ALM Application Lifecycle Management

API Application Programming Interface

ASoT Authoritative Source of Truth

ATO Authority to Operate

AvMC Aviation & Missile Center

BAA Broad Area Announcement

CAMET Curated Access to Model-based Engineering Tools

CAS Comprehensive Architecture Strategy

CCB Change Control Board

DEVCOM Combat Capabilities Development Command

CCRPP Civilian Commercialization Readiness Pilot Program

CDRL Contract Data Requirements List

COTS Commercial Off The Shelf

CRUD Create Read Update Delete

CS Computer Software

CSD Computer Software Documentation

CSV Comma Separated Value

CVIT Continuous Virtual Integration Toolkit

DEVCOM U.S. Army Combat Capabilities Development Command

DFARS Defense Federal Acquisition Regulations Supplement

DID Data Item Description

DOORS Dynamic Object-Oriented Requirements System

DSDM Domain Specific Data Model

DTIC Defense Technical Information Center

DoD Department of Defense

DoDAF Department of Defense Architecture Framework

DSI Design Space Investigator

FACE Future Airborne Capability Environment

FAF FVL Architecture Framework

FRA Functional Reference Architecture

GE General Electric

GFE Government Furnished Equipment

GFI Government Furnished Information

GPR Government Purpose Rights

HTML HyperText Markup Language

HTTP Hypertext Transport Protocol

IBM International Business Machines

INCOSE International Council on Systems Engineering

IRI Internationalized Resource Identifiers

IT Information Technology

JMR Joint Multi-Role

JMR-TD Joint Multi-Role - Technology Demonstrator

JSON JavaScript Object Notation

MBE Model Based Engineering

MBEE Model Based Engineering Environment

MBSE Model Based Systems Engineering

MMS Model Management System

MOSA Modular Open Systems Approach

MSAD Mission Systems Architecture Demonstration

MSI Mission System Integrator

NASA National Aeronautics and Space Administration

NG Next Generation

OMG Object Management Group

OSLC Open Services for Lifecycle Collaboration

OWL Web Ontology Language

PEO Program Executive Office

PLM Product Lifecycle Management

PWS Performance Work Statement

RDF Resource Description Framework

RDFS Resource Description Framework Schema

REST Representational State Transfer

RFP Request for Proposal

RLM Reprise License Manager

SHACL Shapes Constraint Language

SEMP System Engineering Management Plan

SEP System Engineering Plan

SIL System Integration Lab

SME subject matter expert

SMUG Software and Model User Guide

SOW Statement of Work

SPARQL SPARQL Protocol and RDF Query Language

SSoT Single Source of Truth

SysML System Modeling Language

TDP Technical Data Package

TPOC Technical Point of Contact

UAH University of Alabama at Huntsville

UML Universal Modeling Language

URL Uniform Reference Locator

URI Uniform Resource Identifier

USM UoP Supplied Model

USM Unit of Portability Supplied Model

UUID Universal Unique IDentifer

VE View Editor

VM Virtual Machine

XML Extensible Markup Language