MBSE Data Interoperability Specification Report

Evaluation of the SysML v2 Standard

Release 1.0

October 2025

AEROSPACE & DEFENSE PLM ACTION GROUP

Abstract

The Model-Based Systems Engineering (MBSE) project team of the Aerospace & Defense PLM Action Group (AD PAG), an association of aerospace and defense (A&D) companies, completed a year-long investigation of the new SysML v2 standard as a potential solution to enable better MBSE collaboration between Original Equipment Manufacturers (OEMs) and suppliers. This report provides the results of that study.

An overview of SysML v2 highlights the significant changes since SysML v1. The provided benefit and gap analysis includes a verification matrix summarizing the desired capabilities addressed by the new standard. The new SysML v2 framework was applied to different collaboration scenarios to help provide an understanding of different options for implementation within a set of collaborating organizations. Considerations external to the SysML v2 standard that must be planned for as part of a rollout are included. And finally, recommendations for rollout are provided for A&D organizations and MBSE solution providers, and recommendations for future updates to the SysML v2 standard to address the gaps are provided for the Object Management Group (OMG).

This study was encouraging. It is expected that the A&D industry will have great interest in rolling out SysML v2. However, the identified gaps need to be addressed in future SysMLv2.x releases before widespread adoption is expected to occur. It is also expected to take some time before the commonly used frameworks transition to the new standard.

Table of Contents

Table of Figures

Revision Record

Revision	Date	Description
1.0	October 2025	Initial release

MBSE Data Interoperability Specification Report: Evaluation of the SysML v2 Standard

Executive Summary

After completing a year-long investigation, AD PAG member companies remain very interested in Systems Modeling Language (SysML) v2 as an enabler for Model-Based Systems Engineering (MBSE) collaboration. Interest is driven by collaborating internally across functional groups, as well as externally across Original Equipment Manufacturer (OEM) and supplier companies. The thorough investigation resulted in great optimism that SysML v2 will improve the ability for MBSE models to be shared between users with dissimilar SysML tools, thereby mitigating solution provider lock. The fundamental addition of the abstract syntax being accessible through standard application programming interfaces (APIs) is a monumental improvement over SysML v1.x.

Initially, solution providers lacked enthusiasm about SysML 2, but their commitment grew over time. It is encouraging to see the solution provider community excited about SysML v2 adoption. Many new entrants are seated at the SysML "table." This opens the door to integrate tools that currently require custom-made interfaces for communication with each other, requiring an investment that can prevent or hinder integration within the SysML v1 environment.

However, while the SysML v2 value proposition for MBSE is clear, AD PAG MBSE project team members do not see evidence of sufficient technical and business justification to motivate widespread adoption of SysML v2…yet. Major adoption inhibitors include the following:

- Transition Hurdle: Companies maintain a large set of SysML v1 model assets and custom modeling frameworks. Confidence in the ability to translate v1 models to v2 and to realize their modeling frameworks in the v2 environment must be built.
- Maturity of the Standard: Two critical gaps exist in the SysML v2 standard that are key to collaboration. More critical gaps could be found after the standard is released and is exercised using SysML v2 tools. The two identified gaps are as follows:
  - Universally Unique Identifier (UUID) stability. If the UUID changes over time, tools cannot track changes/deletions/additions of model elements.
  - Graphical Layout Stability: Auto-layout is insufficient since the layout impacts model presentation, and diagrams are a key communication mechanism, especially with non-modeling users.
- External Considerations: Companies must define strategies beyond the use of SysML v2 to enable bidirectional system design collaboration. The collaborative SysML v2 information technology (IT) environment is very different than v1 and requires considerations such as change management and data governance to protect intellectual property (IP) between collaborating parties. SysML v2 aids in data sharing, but that doesn't guarantee the data is understood by all parties. Semantics must still be defined and shared to transform the shared model into usable knowledge.

Even in the presence of these inhibitors, AD PAG remains hopeful. The gaps are known by both the Object Management Group (OMG) and the solution providers. The external considerations require thought and effort but are not roadblocks. The question is not “If” but “When” adoption will occur.

The SysML v2 standard was not yet released at the time of writing this report but is expected to be released in 2025.

Introduction

The Aerospace & Defense Product Lifecycle Management (PLM) Action Group (AD PAG) is an association of aerospace Original Equipment Manufacturers (OEMs), aircraft engine manufacturers and Tier 1 supplier manufacturers within CIMdata's globally recognized PLM Community Program, which functions as a PLM advocacy group. The AD PAG sponsored a team of systems engineering domain experts (also known as *subject matter experts*) to evaluate current data interoperability standards intended to enable a Model-Based Systems Engineering (MBSE) conceptual design process. The activity was to assess the feasibility of exchanging digital models, instead of documents, between A&D OEMs and their supply chain partners.

The following four project phases have been completed. This report is the culmination of Phase 5, the most current work.

- **Phase 1** CHARACTERIZE MBSE COLLABORATION PROBLEM: Identified a gap in the capability of commercial SysML-based (Systems Modeling Language) authoring tools offered by the major PLM and MBSE software solution providers. The tools in their out-of-the-box configurations do not support bidirectional MBSE data and model exchange. This means, for example, the tools are not standardized as to **what** and **how** models are exchanged, and configuration management is not ensured.
- **Phase 2** EVALUATE AND RANK SOLUTIONS: Addressed the project team's goal to agree on the most promising strategies and best practice for digital data exchange across the A&D industry. Phase 2 was based on the current maturity level of the most suitable set of MBSE data standards (e.g., SysML, ReqIF, XMI, UMLDI[1]) and related MBSE data authoring tools. The previously published position paper reviews the various solutions considered by the group and provides initial recommendations of the most promising approaches to enable OEM/supply chain design collaboration based on MBSE standards, both short and long term.
- **Phase 3** INVESTIGATE DIRECT EXCHANGE SOLUTIONS: Involved defining the targeted MBSE application use cases that span the systems engineering V-model lifecycle (the WHY) and the key elements of system architecture data that needed to be digitally created and collaboratively exchanged between A&D OEMs and their suppliers/partners (the WHAT). Phase 3 focused on the capabilities within the MBSE tools to directly exchange data with other MBSE tools by method of translation or a shared model format.
- **Phase 4** INVESTIGATE INDIRECT EXCHANGE SOLUTIONS: Continued with a focus on the business challenge of no commercially available system authoring tools using architecture modeling standards that can completely support the direct exchange of digital system architecture models between and among industry OEMs and their global supply chain partners. However, the solution landscape is changing rapidly with third-party indirect solutions. For that reason, the intent of Phase 4 was to answer the question: "What are the current off-the-shelf options (the

[1] ADL – Architecture Description Language, or EADL – Executable Architecture Definition Languages
AADL – Architecture Analysis & Design Language, Carnegie Mellon University, https://wiki.sei.cmu.edu/aadl/index.php/Main_Page
Acme – Carnegie Mellon University, https://www.cs.cmu.edu/~acme/
Arcadia Capella – https://www.obeo.fr/en/capella-professional-offer
ArchiMate – Architecture-Animate, The Open Group, http://www.opengroup.org/aboutus
OPM – Object Process Methodology (ISO 19450), https://en.wikipedia.org/wiki/Object_Process_Methodology
Rapide – https://complexevents.com/stanford/rapide/
SysML – Systems Modeling Language specification, http://www.omgsysml.org/
UML – Unified Modeling Language specification, http://www.omg.org/spec/UML
UMLDI – Unified Modeling Language Diagram Interchange specification, http://www.omg.org/spec/UMLDI

HOW), if any, that enable transition from a document-centric concept design process to a digital and collaborative model-based environment in the near term and to what extent?"

- **Phase 5** INVESTIGATE DIRECT EXCHANGE VIA SYSML V2: This most recent effort was conducted in 2024, and the focus has been to evaluate the potential of the emerging SysML v2 standard for MBSE collaboration between OEMs and suppliers. SysML v2 is a direct exchange solution but is based on a popular standard (i.e., update to a popular standard) versus a proprietary model format. SysML v2 is not just an incremental update. It is a significant overhaul of the modeling language, addressing many of the limitations and challenges encountered while using the SysML language over the past two decades.

The transition to SysML v2 can offer substantial benefits for organizations engaged in complex systems engineering. However, it will also require adaptation and training to leverage the new capabilities most effectively. Without a clear and unified message of urgency and priorities from industry, engineering software solution providers will set their own solution strategies and timelines for incorporating SysML v2 into their software.

The initial release of the SysML v2 standard, SysML v2.0, is expected in 2025 but was not yet released as of this writing. Therefore, it should be noted that all SysML 2.0 standard assessments, conclusions, and recommendations contained in this Phase 5 report are based on paper studies, discussions with software solution providers, and the experience of the subject matter/domain experts within their respective A&D organizations.

Actual hands-on evaluations could not be performed with fully functional MBSE authoring tools that implement the SysML v2 standard because they were not yet available to the AD PAG MBSE project team. Pilot implementations were evaluated; however, they were very limited in functionality and did not represent a full SysML v2 tool.

In this report, SysML v2 refers to the general standard, where SysML v2.0 refers to the initial release of the standard.

A&D Industry Business Drivers

Digital engineering and digitization of the product development process from the early conceptual design stage through the entire product lifecycle is a major business strategy across all industries today. This is especially true in complex A&D systems design and development programs, which are motivated to resolve the design complexities of cyber-physical systems with ever-increasing amounts of embedded software and electronics.

Many economic business drivers, such as the following, are cited for achieving the end-to-end digital and model-based process, often referred to as the *digital thread*:

- Greater market share and profitability via faster time to market (i.e., shorter product design and development cycles)
- Increased system integration leading to more optimized products (i.e., design features directly linked to meeting end-customer functional requirements)
- Increased enterprise engineering productivity and supply chain collaboration by using robust digital models versus documents throughout the product development lifecycle
- Improved collaboration in model-based design, resulting in fewer engineering change orders (ECOs) and fewer physical prototype iterations

- Enhanced product quality and reliability via continuous design validation and integrated requirements (i.e., optimized designs via digital modeling and simulation of performance before hardware)
- Reduced total lifecycle costs, including manufacturing, warranty, and in-service operations (i.e., digital models evolve to meet other domain requirements – digital twins)
- Maintained compliance with global industry regulations (e.g., safety certification, reuse/green)

While optimistic in vision and rigorous in scope, these economic business drivers are influenced by a fundamental problem.

Problem Statement

Specific to the business driver opportunities listed above, A&D OEMs recognize that collaborative systems design and development within a large, globally distributed, supply chain of design and development partners is seriously hindered by reliance on traditional, document-based development processes.

OEMs are addressing this challenge by expanding their use of digital and model-based software solutions that define and manage overall system requirements, associated system architectures, behavior assessments/validation and verification (V&V), product safety, intellectual property (IP) security, and regulatory or contractual obligations. To ensure an inclusive implementation, OEMs are engaging their supply chains to collaborate closely through the exchange of conceptual digital models that communicate the design intent back and forth in a robust and iterative product development process that does not rely on traditional documents, singular artifacts, and/or drawings.

For this purpose, an AD PAG project team of MBSE subject matter/domain experts was formed. The AD PAG MBSE project team members constituted a multi-national cross-section of engineering skills representing member companies of the AD PAG. Their charter was to evaluate current data interoperability standards for enabling an MBSE conceptual design process that relies on “dynamic” digital models for design collaboration instead of “static” technical data packages of documents and drawings.

MBSE Project Team and Its Goals / Objectives

The AD PAG MBSE project team for Phase 5 consisted of industry subject matter/domain experts from Airbus, Boeing, GE Aerospace, Gulfstream, Rolls-Royce, Safran, Spirit AeroSystems, and CIMdata. Participation was voluntary. At least three different companies' participation was required to establish a forum that could make decisions. Virtual working meetings were conducted on a regular basis throughout 2024, and several in-person workshops using the English language were held during calendar years 2023 and 2024.

In October 2023, the project team met in Toulouse, France for a two-and-a-half-day workshop to chart their Phase 5 course of action for 2024. The project **goals** agreed upon at that workshop were to do the following:

- Provide an appraisal of the SysML 2.0 standard within the context of interoperability, and identify where SysML 2.0 solves challenges facing the industry and where gaps remain

- Define and communicate a clear and unified message from industry to the PLM/MBSE solution providers regarding the requirements and foreseen gaps in interoperability capability
- Collaborate with other industry working groups and standards bodies, such as International Council on Systems Engineering (INCOSE), the Object Management Group (OMG), and PDES, Inc. – International Industry Consortium
- Communicate industry needs/priorities as MBSE solution providers begin implementing the SysML 2.0 standard in their next generation of software solutions

Based on those goals, the MBSE project team's key **objectives** were defined as the following:

- Document the AD PAG's assessment of the completeness of the SysML 2.0 standard to meet the AD PAG members' business objectives
- Identify any technical gaps and implementation challenges that remain to be addressed
- Engage with the leading PLM/MBSE solution providers and industry standards organizations to ensure that A&D industry needs are heard clearly and acted upon in future releases of the standard and associated software solutions

All AD PAG MBSE team members who contributed to this evaluation and its subsequent Phase 5 report are not directly involved in the development or deployment of SysML v2, either through the OMG or through solution providers. Therefore, the views that follow are observations made by MBSE subject matter/domain experts on the current application trajectory of SysML v2 from the perspective of industry end users.

No determination is made on whether identified issues are to be resolved in a future release of the standard or in the implementations by solution providers; that determination should be made in consultation between OMG, solution providers, and preferably the AD PAG MBSE project team.

Overview of SysML v2

Developed and published within the OMG, SysML is a standardized systems modeling language used to represent complex systems reliably and efficiently. SysML v1.x extended the matured Unified Modeling Language (UML) to provide a standardized modeling language for expressing the needs of systems engineering through a model-based approach (i.e., system requirements, behaviors, structures and parametric constructs, etc.) for a wide range of systems, including hardware, software, information, processes, personnel, and facilities.

As a mechanism to articulate systems engineering concepts for a diverse set of systems, SysML v1.x is widely used in or evaluated as a software solution for PLM ecosystems deployed within the A&D industry. Despite its utility and, in certain ways, because of the scope of expressibility possible, SysML v1.x has faced challenges related to complexity, ambiguity, and tool interoperability, thereby prompting the development of SysML v2.

What Distinguishes SysML v2 from SysML v1.x?

SysML v2 is envisioned to address most of the aforementioned shortcomings of SysML v1.x, aiming to fundamentally support expressibility but with a more precise, usable, and tool interoperable approach. The AD PAG MBSE project team identified several significant differences—key benefits—of choosing SysML v2 over SysML v1.x. The following overview does not review all changes; materials from OMG should be consulted for a complete understanding of differences between SysML v2 and SysML v1.x.

Foundational Metamodel / Abstract Syntax: UML to KerML

As noted earlier, SysML v1.x[2] is an expansive UML[3] profile, providing stereotypes and sub-profiles to express system engineering constructs. This UML dependency has led to inherent complexities and constraints, in part due to the inherent software engineering expressive nature of UML not aligning perfectly with the needs of multidisciplinary systems engineering topics.

SysML v2[4] is built upon a new foundational metamodel language/abstract syntax known as *KerML* (Kernel Modeling Language).[5] KerML, as a standalone language, was designed specifically for systems modeling, providing a more suitable semantic foundation to overcome the misalignment issues identified through the expansion of UML in SysML v1.x. The *What is KerML?* section below briefly explores this new metamodel in further detail.

Modeling Constructs and Diagrams: Unified and Simplified

An often confusing and daunting learning curve of adopting SysML v1.x has been the use of disparate diagram types with inconsistent notations and semantics (e.g., identifying the specific application scenario for using a Block Definition Diagram, an Internal Block Diagram, or a Parametric Diagram). Each diagram type has its own syntax and individual constraints, causing confusion and opportunities for unnecessary redundancy.

Due to the consistent foundational metamodel/abstract syntax between diagrams, SysML v2 introduces the following:

- Unified set of modeling constructs for defining system elements, relationships, and behaviors based on the common abstract syntax
- Clearer and reduced number of diagram types
- Crucially consistent semantics across different diagram types

This unification of constructs, diagram semantics, and a reduced set of diagrams from which to choose simplifies the language and enhances the understandability of a singular systems engineering construct being represented across different diagrams. It is akin to providing different orientation perspectives within the geometrical modeling field.

Model Representation: Textual and Graphical Notations of the Abstract Syntax

The new abstract syntax is suitable for implementing data repositories but is not directly human readable. Therefore, SysML v2 provides two forms of concrete, human-readable notation: textual and graphical. Both textual and graphical notations are direct representations of the underlying abstract syntax. For example, while editing a model through the textual notation or graphically, the edits are being made to the underlying abstract syntax.

[2] "SysML v1," [Online]. Available: https://www.omg.org/spec/SysML/1.5/PDF
[3] "UML," [Online]. Available: https://www.omg.org/spec/UML
[4] "SysML v2," [Online]. Available: https://www.omg.org/spec/SysML/2.0
[5] "KerML," [Online]. Available: https://www.omg.org/spec/KerML

Within SysML v1.x, the primary interaction/representation of a model was provided through the graphical format (limited textual support available). For those new to the benefits of MBSE, on occasion this has caused confusion between the functionality of SysML editors and generic graphical editing tools. In addition, the lack of consistent semantics across diagram types combined with a primarily visual model representation caused concerns across the industry about the scalability of accessing and understanding complex inferred concepts.

SysML v2 provides a formal textual syntax, in synchronization with the graphical notation, built from the abstract syntax representation. Consequently, unlike in SysML v1.x, the proprietary repository is available for interpretation through both textual and graphical notations. Addressing the misconception between SysML editors and the generic graphics tool, the code is now exposed, and the concepts and usability of the model-based constructs can be more readily understood by a wider audience. Textual notation could help reduce the concern of model scalability. It applies similar approaches used in more traditional high-level programming languages to manage complex and interconnected projects. Finally, a textual notation provides potential to use traditional text-based version control and file collaboration systems. The applicability of SysML v2 with common version control integrations was considered in the evaluation, and further interoperability enablers (in addition to file transfers) are addressed in the *Compatibility and Tool Integration: Standardized API and File Transfer Approaches* section below.

Requirements: Improved Integration and Traceability

Within SysML v1.x, requirements are treated as standalone entities within requirement diagrams and are somewhat isolated from other model elements. Typically, forming traceability between requirements and other model elements is managed through relationships such as Satisfy, Verify, Derive, etc. These links are often implicit and require manual management and navigation across multiple diagrams and/or tables, leading to fragmented models and inconsistent modeling approaches (i.e., outside of those specifically declared within a methodology).

SysML v2 introduces the following improvements in how requirements are modeled and integrated:

- Requirements are modeled with the pattern of definition and usage
- Requirements can be defined with special kinds of constraints using formal, evaluable expressions and attributes that can be bound to other features of the model

Requirements are now tightly integrated into the modeling language and can directly reference components, behaviors, or constraints, enabling more explicit relationships. Additionally, the traceability relationships noted above—Satisfy, Verify, Derive—are inherently defined in the SysML v2 metamodel with the mandated and consistent semantics now providing the opportunity to perform automations and impact analysis based on the conditions defined.

The overall result is that requirements integration in SysML v2 more closely aligns the language with the principles of ISO/IEC/IEEE 15288:2023[6]. The requirements are now not only documented but fully traceable throughout the Design and V&V stages. SysML v2-compatible requirements

[6] "15288:2023," [Online]. Available: https://www.iso.org/standard/81702.html

tools are able to store requirements in a SysML v2 repository that can be referenced by architecture models created by other SysML v2 tools.

Compatibility and Tool Integration: Standardized API and File Transfer Approaches

A primary criticism of SysML v1.x that implementors have had is the interaction of SysML v1.x with other digital engineering capabilities (i.e., beyond interacting with other SysML v1.x editors). Lack of a standardized file syntax has tied interactions with SysML v1.x models to the MBSE solutions being used.

Conceptually, SysML v1.x and v2 models are conceived as part of a federated source of truth and must interact with other digital capabilities, potentially from different solution providers. To date, the interplay between PLM platforms and SysML repositories has been focused on interactions with a specific tool solution provider's representation of a SysML v1.x repository. With SysML v2, there is enthusiasm that tool compatibility will be widespread.

The SysML v2 specification provides a standardized API, along with the textual representation discussed previously, to enable greater seamless data sharing and integration between different platforms. An exciting response is that solution providers have alluded to more tools being able to interface directly with SysML v2 repositories.

What is KerML?

As previously stated, SysML v2 is built on a new foundational metamodel/syntax known as *KerML*. The Kernel Modeling Language is a foundational, domain-neutral modeling language developed to provide a consistent and more complete semantic basis for defining other modeling languages. While today KerML serves primarily as the semantic core for SysML v2, the new metamodel is designed to be flexible and extensible and, therefore, appropriate for producing modeling languages across different domains.

A view of the future is that other Domain-Specific Languages (DSLs) can be produced by leveraging KerML's core constructs and formal semantics to create languages tailored to specific domains or purposes. Indeed, KerML can serve as a potential mechanism for integrating existing modeling standards by mapping constructs to KerML's core elements. Therefore, the core vision is for KerML to facilitate interoperability between modeling languages (i.e., two partners using a different modeling language or semantic methodology).

As SysML v2 is the first DSL to be widely expressed in KerML as its foundational metamodel, the feasibility of realizing true semantic interoperability between modeling languages remains an open question. The domain libraries are intended to provide a vehicle to share ontologies in a standard way between modeling parties. The AD PAG MBSE project team was unable to verify that this solution is robust enough for varied types of semantics because SysML v2 tools were not accessible during this investigation. Robustness for varied types of semantics is an important aspect of the new standard and needs to be exercised in a SysML v2 toolset to assess whether semantics can be unambiguously defined and utilized between modeling parties. The need for semantic interoperability is identified in the *Considerations External to the Standard's Scope* section.

Solution Providers Reporting SysML v2 Support

The introduction of SysML v2 has ignited excitement within the MBSE tool solution provider community. As expected, the established solution providers of SysML v1.x tools are actively planning to incorporate SysML v2 support into their upcoming releases. They recognize the enhanced capabilities and efficiencies the new standard offers.

Notably, solution providers who previously did not support SysML v1.x are now developing SysML v2-compliant solutions, signaling a broader industry shift toward embracing the benefits discussed above. These new entrants are not limiting themselves to traditional architecture modeling. The abstract syntax of SysML v2 opens avenues for integration across the digital thread, connecting diverse data elements, such as requirements, project management tasks, analytical models, Bill of Materials (BOMs), problem reporting systems, Bill of Processes (BOPs), manufacturing plans, work orders and routings, master plans and schedules, etc., and 3D Computer-Aided Design (CAD) tools. In theory, the future would facilitate seamless collaboration and data exchange across different domains, enhancing the efficiency and effectiveness of systems engineering processes.

Furthermore, SysML v2's foundation on the KerML significantly boosts its integration potential with other non-SysML v2 tools. KerML provides a flexible and extensible meta-modeling approach, enabling solution providers to create robust integrations with an array of digital engineering tools and systems. This alignment not only broadens the interoperability of SysML v2 but encourages the development of a more cohesive and interconnected modeling ecosystem, supporting a diverse range of engineering and management activities.

Conclusion of the SysML v2 Overview

The *Overview of SysML v2* section has indicated the positive direction being made with SysML v2 within the OMG. Clearly industrial considerations in the standard have been heard and acted upon. Interestingly though, discussion presented in this report's *Considerations External to the Standard's Scope* section illustrates that interoperable collaboration is still challenging.

While interoperability between partners performing SysML collaboration is now less of a challenge with SysML v2, it is still a significant trial. A holistic consideration about what other principles require being standardized is crucial. The considerations that could hold back SysML v2 deployment, based purely on its merits of supporting extended enterprise collaboration, are likely not within the scope of SysML.

Evaluation of SysML v2 for Collaboration

The AD PAG MBSE team members were generally aware of SysML v2 when the project was initiated, but the team did not have a sufficiently extensive understanding of how it differed from SysML v1.x. Therefore, an investigation was employed with the following three objectives in mind:

1. Identify a SysML v2 "requirements" wish list. The team members jointly defined a capability wish list for SysML v2. That list was used to evaluate the SysML v2 solution.
2. Perform a benefit/gap analysis and evaluate SysML v2 against the wish list capabilities. The MBSE team evaluated SysML v2 against the identified capability wish list. They gained an

understanding of both the benefits related to transitioning to SysML v2 and the potential gaps in the new standard that need to be considered before bringing SysML v2 into the member organizations.

3. Analyze implementation/rollout. The team learned how the new SysML v2 data framework could be applied in four identified collaboration scenarios—unidirectional, bidirectional, joint authorship, and platform co-simulation—so member companies could evaluate how they might roll out SysML v2 in their respective organizations.

Objective 1: Identify SysML v2 "Requirements" Wish List

Following a systems engineering approach, the AD PAG MBSE project team jointly defined a wish list for SysML capabilities. Essentially, the list serves as pseudo "requirements" and provides a foundation that evaluations could be performed against to verify which "requirements" were satisfied by the new SysML v2 standard. It should be noted that not every capability could be verified given the lack of availability of released SysML v2 tools.

An outline of the AD PAG MBSE team's SysML v2 capability wish list is provided below in the form of a numbered index.

1. Better Collaboration
    a. Unidirectional and bidirectional collaboration in exchanging model artifacts
        i. Common file exchange format
    b. Joint collaboration simultaneously within the same model artifacts
        i. Standardized API access to model data in joint collaboration
    c. Model referencing
    d. Model segregation, as needed for organization and intellectual property protection
    e. Common graphical representation within an organization, while allowing for flexibility in how other organizations represent the same model data
2. Synchronization with Other Tooling (must have)
    a. Commercial Off-the-Shelf (COTS) tools
    b. In-house custom tools
    c. Synchronization types: read-only (unidirectional/bidirectional) and read/write (joint collaboration)
3. Ontologies
    a. Ability to implement an ontology within the SysML v2 model
    b. Ability to share ontology within the SysML v2 repository
    c. Ability to utilize ontologies defined outside SysML v2 (e.g., MoSSEC)
4. Model Data Management
    a. Model format that serves as an authoritative source of truth (ASoT)
    b. Model format suitable for change management
        i. Textual Notation or Abstract Syntax
        ii. Graphical notation is not a good format for change management
    c. Graphical model representation is necessary to foster human understanding
    d. Common model database where data can be accessed in a standard way
    e. Change management

   i. Universally Unique Identifiers (UUIDs), necessary to manage model changes across a set of collaborating users
   ii. Branch and merge functionality
   iii. Model differencing across model branches
   iv. Model archive at major baselines with ability to recreate model database

f. Effective part reuse
   i. Type 1: Part instantiation with property specification at both part and instance level in the same model
   ii. Type 2: Reuse of parts from one model in a separate model/view

g. Model query language specification
   i. Nice to have: translation between SysML v2 query language and commonly used external query languages (e.g., SQL, openCypher)

h. Cybersecurity
   i. Nice to have: collaboration between organizations in a common IT environment such as a "Cloud" requires a cybersecurity layer that not only manages user access controls but also helps protect against malicious model access attempts; it would be helpful for the SysML v2 standard to acknowledge this consideration when defining the Representational State Transfer (RESTful) APIs

Objective 2: Perform Benefit / Gap Analysis and Evaluate SysML v2 Against the Wish List Capabilities

The MBSE project team's benefit/gap analysis is organized into several subtopics: metamodel, textual notation, graphical notation, abstract syntax, and APIs. Each subtopic addresses both benefits and gaps. This section also addresses the change management, part reuse, and model query language capabilities of the model data management "requirement." An evaluation status of the SysML v2 capabilities is provided as a table in the *SysML v2 Capability Wish List Verification Matrix* subsection below.

Metamodel

The SysML v1.x metamodel is derived from the Unified Modeling Language (UML) that is used in software engineering. As such, SysML remains with a software bias. As noted earlier, the SysML v2 metamodel is KerML, which is natively defined to describe systems as well as software. Therefore, it is more appropriate for use in system architecture modeling. Where UML leverages diagram languages to describe a model, KerML describes models in either a human-readable textual notation or a machine-readable abstract syntax that is a formal semantic representation.

Benefits

SysML v2 provides a semantic library, which helps to better formalize the modeling language.

Another benefit is known as *declarative behavior modeling*. UML distributes the behavior model over three separate diagram languages and over three subsets of the UML abstract syntax. KerML models behaviors in a uniform and declarative way by doing the following:

- Viewing behaviors as classes that are instantiated by behavior executions/occurrences

- Considering the composition of behaviors (i.e., components of behaviors modeled as steps, akin to components of objects modeled as parts)
- Logically sequencing the steps of a behavior using successions, a special connector that defines temporal precedence constraint[7]

Gaps

SysML v2 semantic descriptions are limited and do not provide full descriptions for all types of architecture models.

Textual Notation

Textual notation is new to SysML v2. It is the AD PAG MBSE team's opinion that it may be useful as an authoritative data source for light models, but it is not sufficient for large models. It is difficult to understand large models in a textual format. While textual models are human-readable, they are difficult to navigate and maintain an understanding for very large system models spanning thousands of pages of text. Some solution providers reported that they do not consider it an ASoT.

Benefits

The primary benefits of textual notation include the human-readable format, the archivable format—it is assumed a textual format can be readable into the foreseeable future (i.e., 50 years+)—and its support of text-based differencing tools, which are readily available for source code analysis.

Gaps

Textual notation lacks UUID capability. Therefore, based on textual notation, change management in a collaborative environment would not be feasible. External parties have attempted to add UUID to the textual format outside of the SysML v2 standard, and in review, the text-based model becomes littered and difficult to read. Textual notation is a friendly format for software engineers who are accustomed to coding, but it is less friendly for systems engineers who serve as the primary users of SysML.

Graphical Notation

Graphical notation is required for human interaction with a large model to build knowledge and understanding. Historically, graphical notation is also the foundation of the SysML standard and continues to be a key component in SysML v2, albeit while now supported by the textual notation and abstract syntax.

It should be noted that the AD PAG MBSE team was not able to compare the SysML v1.x and SysML v2 graphical notation due to lack of availability of released SysML v2 modeling tools during the period of this investigation. A hands-on review is necessary for a comprehensive assessment of the graphical modeling language. The following conclusions are based on pure investigation and solution provider engagement.

[7] G. Wagner, "Understanding KerML and SysML," 14 11 2024. [Online]. Available: https://sim4edu.com/reading/kerml

Benefits

Graphical notation provides the ability for humans to interpret, understand, and author models. These benefits can even be realized when models are created using textual notation, since both notations can share a common abstract notation as the source of truth.

Gaps

Reportedly, graphical notation lacks rigor in deterministically translating textual notation to graphical notation in certain use cases, potentially resulting in differing representations between collaborating parties. This is an area for further study when the AD PAG MBSE project team has access to released SysML v2 modeling tools. Another gap is that the graphical layout is not stored in the data repository in a standard way that is understood by all collaborators. Additionally, the standard relies on tool auto-layout functionality. This means that two users may view the same model but with differing layouts. Layout is important to model understanding. It is typical to lay out components in a specific way to facilitate understanding, such as dependency flow or grouping by similar functionality. While layout is not included in SysML v2.0, an A&D industry organization reported that they believe OMG has reached consensus on how to add graphical layout to future SysML 2.x versions of the standard.

Abstract Syntax

Abstract syntax is new to SysML v2. It is a formal, machine-readable, semantic representation of the model that provides a common database format accessible through RESTful APIs. The AD PAG MBSE team believes it is suitable for use as an authoritative data source for models of any size.

Benefits

A benefit of abstract syntax is that it provides a common format that is agnostic to the representation format for human consumption, enabling tools to effectively access the model data and allowing some flexibility in how the data is represented.

Gaps

Stability of UUID is in question. While UUIDs are included in the abstract syntax, solution providers have indicated that UUIDs are not required to remain stable. That is, the UUID can be changed after it has been created. The AD PAG MBSE team's understanding is that the stability of a UUID is tool dependent. One tool may maintain the UUIDs, but if another tool writes to the shared repository, then the UUIDs could be updated in the model, impacting all collaborating parties. This is a critical flaw for purposes of robust change management.

Application Programming Interfaces

Application programming interfaces were included as a Better Collaboration subtopic of the AD PAG MBSE team's wish list. RESTful APIs provide access to the model data in the repository. The AD PAG MBSE team was unable to exercise the APIs using a released SysML v2 tool; this step is necessary for a comprehensive review. These conclusions are based on API documentation review.

Benefits

The APIs are the critical key to standardizing model collaboration across tools. The APIs enable multiple modelers to collaborate within the same model. Additionally, as a standard definition of an interface to the model, the APIs provide potential for tools from different vendors to access the same model. Avoidance of tool vendor lock is a significant hope within the A&D industry.

Gaps

Identified gaps are described in later sections of this report. In summary, while the following features may not be included in the current scope of SysML v2, the A&D project members identified the importance of (a) joint collaboration support using APIs with an effective query language and (b) support for standardized graphical representation that retains layouts (*Summary of Progress Toward the AD PAG MBSE Team "Requirements"* section below).

Model Data Management - Change Management Capability

The AD PAG MBSE Team's capability wish list included change management as a subtopic of Model Data Management. Change management is critical to any development, especially in a highly collaborative environment. The wish list capability for change management was investigated, and the conclusions follow:

- **UUIDs are necessary to manage model changes** across a set of collaborating users. The UUID is part of the SysML v2 specification. Solution provider engagement seems to indicate that the specification does not require UUIDs to remain stable. That means a UUID can be changed after defining a UUID for a part. As noted in the *Abstract Syntax* section above, this is a critical failure in change management and a huge gap in SysML v2 implementations.
- **Branch and merge functionality** is supported by the SysML v2 standard.
- **Model differencing** across model branches is supported by the SysML v2 standard.
- **Model archive at major baselines with the ability to recreate the model database** is supported by the SysML v2 standard.

Model Data Management - Effective Part Reuse Capability

The AD PAG MBSE Team's capability wish list included part reuse as a subtopic of Model Data Management. Part reuse is critical in A&D system architectures that greatly make use of redundancy to achieve needed system reliability (i.e., availability and integrity). The capability wish list for effective part reuse was investigated, and the conclusions follow:

- Type 1: Part instantiation with property specification at both the part and the instance level in the same model. This requires further investigation. Properties can be defined as metadata or as an attribute. It is believed that metadata is not inherited. Attributes can be inherited. Therefore, it may be appropriate to define attributes on a base part that apply to instances of the part.
- Type 2: Reuse of parts from one model in a separate model/view. Model refencing/usage is supported by the SysML v2 standard.

Model Data Management - Model Query Language Specification Capability

The AD PAG MBSE Team's capability wish list included model query language specification as a subtopic of Model Data Management. While not mandatory, it would be nice to have a mapping with external query languages, such as Object Constraint Language (OCL) and openCypher.

SysML v2 includes a new query language that is used to access model data in the repository. SysML v2 does not provide a mapping to external query languages. Initial reports indicate this new query language is simple in scope, requiring a large number of small query statements to traverse a model, thus driving an inefficient, time consuming set of API interactions.

SysML v2 Capability Wish List Verification Matrix

Continuing with the systems engineering approach, the AD PAG MBSE project team constructed a SysML 2 verification matrix. This matrix, which follows, summarizes the verification status for the SysML v2 capabilities the project team defined in the wish list as pseudo "requirements."

Capability Wish List Verification Matrix			
Wish List Category	**Wish List "Requirement"**	**Verification Status (by analysis)**	**Comments**
Better Collaboration	Unidirectional and bidirectional collaboration in exchanging model artifacts	Verified	
	Common file exchange format	Verified	
	Joint collaboration simultaneously within the same model artifacts	Verified	
	Standardized API access to model data in joint collaboration	Verified	The abstract syntax defines a metamodel that is common across all implementations. The implementation of the abstract syntax in a data repo can be proprietary, and the API service is responsible for providing access across hardware/operating system (OS)/software environments.
	Model referencing	Verified	

Capability Wish List Verification Matrix			
Wish List Category	**Wish List "Requirement"**	**Verification Status (by analysis)**	**Comments**
Better Collaboration (continued)	Model segregation, as needed for organization and IP protection	Fail	Models can be segregated to organize data and group data by user access control groups. However, no data governance functions are defined by the SysML v2 standard; data governance functions must be implemented via external considerations. The lack of standard data governance leads to potential vendor lock. See the *Considerations External to the Standard's Scope* section below.
	Common graphical representation within an organization, while allowing for flexibility in how other organizations represent the same model data	Fail	Graphical layout is not stored in the data repository, at least not in a standard way that is understood by all collaborators. The standard relies on tool auto-layout functionality. This means that two users may view the same model but with differing layouts. Layout is important to model understanding, and it is typical to lay out components in a specific way to facilitate understanding, such as dependency flow or grouping by similar functionality. It should also be noted that further investigation is required to compare the SysML v1.x graphical language with SysML v2.

Capability Wish List Verification Matrix			
Wish List Category	**Wish List "Requirement"**	**Verification Status (by analysis)**	**Comments**
Synchronization with Other Tooling (must have)	COTS tools	Verified	
	In-house custom tools	Verified	
	Synchronization types: read-only (unidirectional/ bidirectional) and read/write (joint collaboration)	Verified	
Ontologies	Ability to implement an ontology within the SysML v2 model	Requires Further Investigation	
	Ability to share ontology within the SysML v2 repository	Requires Further Investigation	
	Ability to utilize ontologies defined outside of SysML v2 (e.g., MoSSEC)	Requires Further Investigation	
Model Data Management	Model format that serves as an ASoT	Verified	
	Model format suitable for change management	Verified	Abstract syntax is suitable for change management, but textual notation is not due to lack of UUID support.
	Graphical model representation is necessary to foster human understanding	Verified	
	Common model database where data can be accessed in a standard way	Verified	

Capability Wish List Verification Matrix			
Wish List Category	**Wish List "Requirement"**	**Verification Status (by analysis)**	**Comments**
Model Data Management (continued)	Change Management	Verified	Basic change management is provided by SysML v2, such as model UUIDs and model branching, but a full solution requires external consideration.
	UUIDs, which are necessary to manage model changes across a set of collaborating users	Verified	UUID is supported only in abstract syntax; it is not supported by the concrete syntax of textual notation and graphical notation.
	Branch and merge functionality	Verified	
	Model differencing across model branches	Verified	
	Model archiving at major baselines with ability to recreate model database	Verified	
	Effective part reuse	Requires Further Investigation	
	Type 1: Part instantiation with property specification at both part and instance level in the same model	Requires Further Investigation	Properties can be defined as metadata or as an attribute. It is believed that metadata is not inherited. Attributes can be inherited, so it may be appropriate to define attributes on a base part that apply to instances of the part.
	Type 2: Reuse of parts from one model in a separate model/view	Verified	
	Model query language specification	Verified	

Capability Wish List Verification Matrix			
Wish List Category	**Wish List "Requirement"**	**Verification Status (by analysis)**	**Comments**
Model Data Management (continued)	Nice-to-have: translation between SysML v2 query language and commonly used external query languages (e.g., SQL, openCypher, SparQL)	Fail	Translation between the SysML v2 query language and external query languages is not defined.
	Cybersecurity	Fail	CyberSecurity is important but not part of SysML v2 and must be addressed as part of the external considerations. See the *Establishing IT Infrastructure to Support SysML v2 Collaboration* subsection in the *Considerations External to the Standard's Scope* section below.

Objective 3: Analyze Implementation/Rollout

SysML v2 employs a remarkably different data framework than SysML v1.x, which is constrained to graphical modeling. SysML v2 formalizes the model in a machine-readable format (i.e., abstract syntax) and stores it in a data repository accessible through RESTful APIs. The abstract syntax can be represented in a graphical or textual format for human understanding. Therefore, the roll out of SysML v2 looks very different than that of SysML v1.x.

To better understand the rollout options, the SysML v2 data model framework was applied to the four collaboration scenarios–unidirectional, bidirectional, joint authorship, and platform co-simulation. These are not represented as a comprehensive set of possible SysML v2 applications in collaboration environments. Rather, they are intended to help organizations think about different options and facilitate discussion.

SysML v2 Implementation Framework for Use with Collaboration Scenarios

As shown in the following figure, a SysML v2 implementation framework was defined and used as the foundation for applying SysML v2 to the collaboration scenarios.

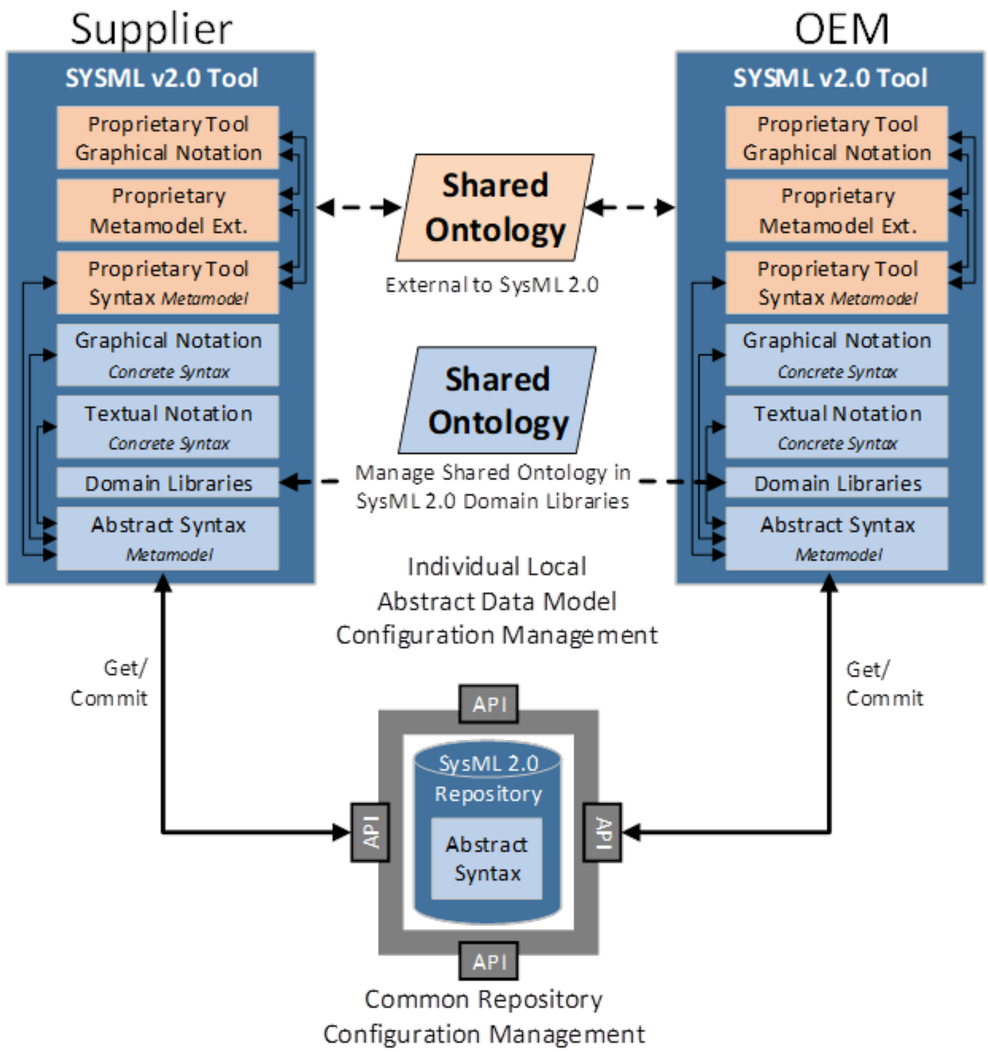

Figure 1 - SysML v2 Implementation Framework

The SysML v2 tool accesses a remote data repository using a standard set of RESTful APIs. Solution providers currently employ their own proprietary data formats. The AD PAG MBSE team and solution provider engagements in this investigation led the team to believe solution providers will continue supporting those formats so they do not abandon their current customer base, at least for the foreseeable future. There was agreement that this is a good strategy to maintain current model assets. Therefore, the SysML v2 tool will likely (hopefully) provide translation between the proprietary model formats (non-SysML) and SysML v2 formats. This is shown in the implementation framework above through the inclusion of blue SysML v2 modules, as well as orange proprietary data formats.

A shared ontology is still required so that collaborating parties can understand the modeling data. While SysML v2 provides a standard way to access model data, it does not provide a standard way to understand the data.

The ontology is shared separately from the model data in SysML v1.x modeling environments. SysML v2 includes the concept of a domain library, which the AD PAG MBSE team believes could be potentially used to codify an ontology and be shared in the data repository. However, this needs to be exercised in SysML v2 tools to validate the concept. The figure above shows the traditional ontology (orange), which is shared for use in proprietary model formats, as well as the ontology shared within a SysML v2 domain library (blue). The goal is to fully describe the ontology in the blue SysML v2 domain, such that the orange ontology is only required for collaboration with legacy non-SysML v2 tools.

The implementation framework was applied to the four collaboration scenarios to explore how SysML v2 might be rolled out.

MBSE Collaboration Scenarios and Workflows

As noted earlier, the AD PAG MBSE project team evaluated SysML v2 in the context of four different scenarios for MBSE collaboration: unidirectional, bidirectional, joint authorship, and platform co-simulation. These following collaboration descriptions and images of Figures 2, 5, 8, and 11 are derived from experimental frameworks of AD PAG member companies Airbus and Boeing.[8]

Unidirectional Collaboration Scenario

One modeling party is responsible for the authorship of the model, and the other collaborating party/parties provide comments for model updates so the model author can make changes. This scenario provides the most control of the model by minimizing the number of parties that can make changes, but it is more cumbersome for modeling parties to collaborate.

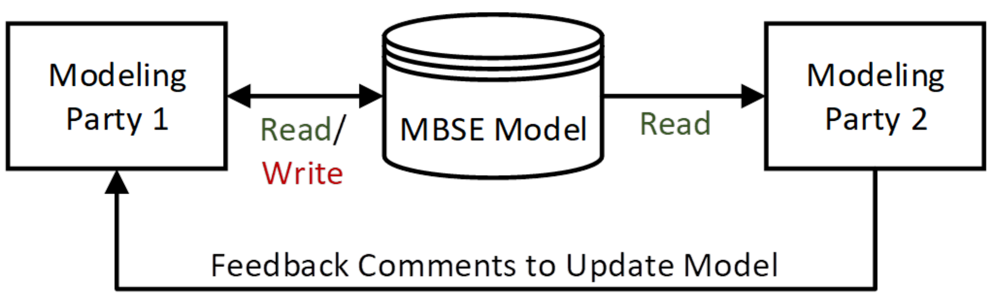

Figure 2 - Unidirectional Collaboration

[8] Ref. Juan Carlos Mendo, Mark Williams, "Boeing's Supplier Engagement Framework (SEF) and Technical Data Package (TDP): The Need for Industry Alignment", Global Product Data Interoperability Summit, 2022

Unidirectional Collaboration Workflows

Two SysML v2 workflows were identified for use in unidirectional collaboration:

- **Project Interchange File(s).** A SysML v2 project interchange file (.kpar) can be used to archive the SysML v2 data repository to be shared with a reviewer modeling party. The reviewer uses the project interchange file to create a copy of the model data repository in their own environment. The reviewer does not edit the model but can provide feedback comments so the model owner may make all changes and share another baseline of the model in another project interchange file. This does not require a shared IT infrastructure.

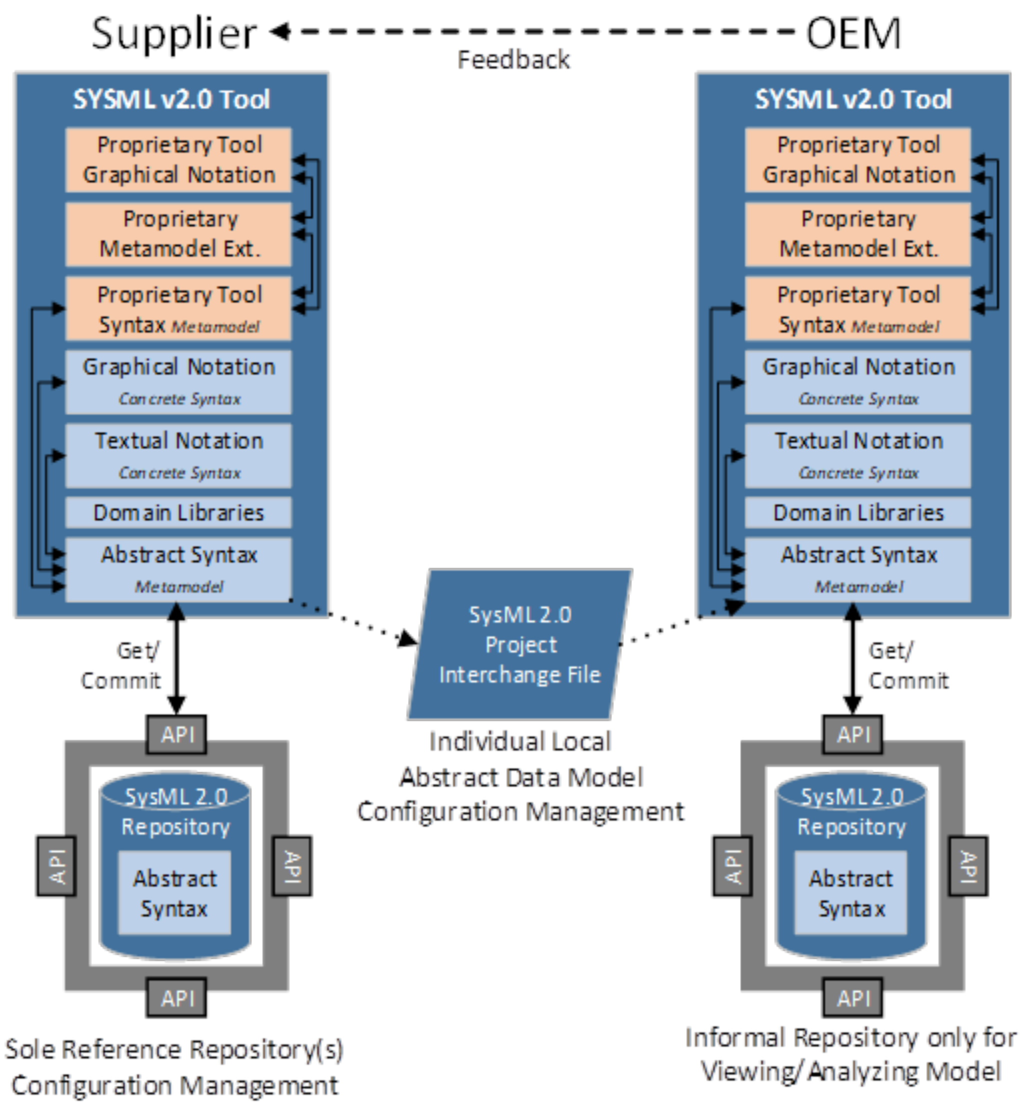

Figure 3 - Unidirectional: Project Interchange File(s) Workflow

- **Shared Model Repository/Repositories.** A shared model repository can also facilitate unidirectional collaboration where only a single modeling party makes changes to the model. This can be an unenforced, informal agreement or strictly enforced through user access controls implemented around the SysML v2 repository. This workflow is technically preferred, as compared to project interchange files, because it provides live access to the most recent model data. However, it requires a shared IT storage infrastructure, which is not easy to setup between separate organizations given cybersecurity constraints, as well as management of contractual agreements of data access and data/IP ownership.

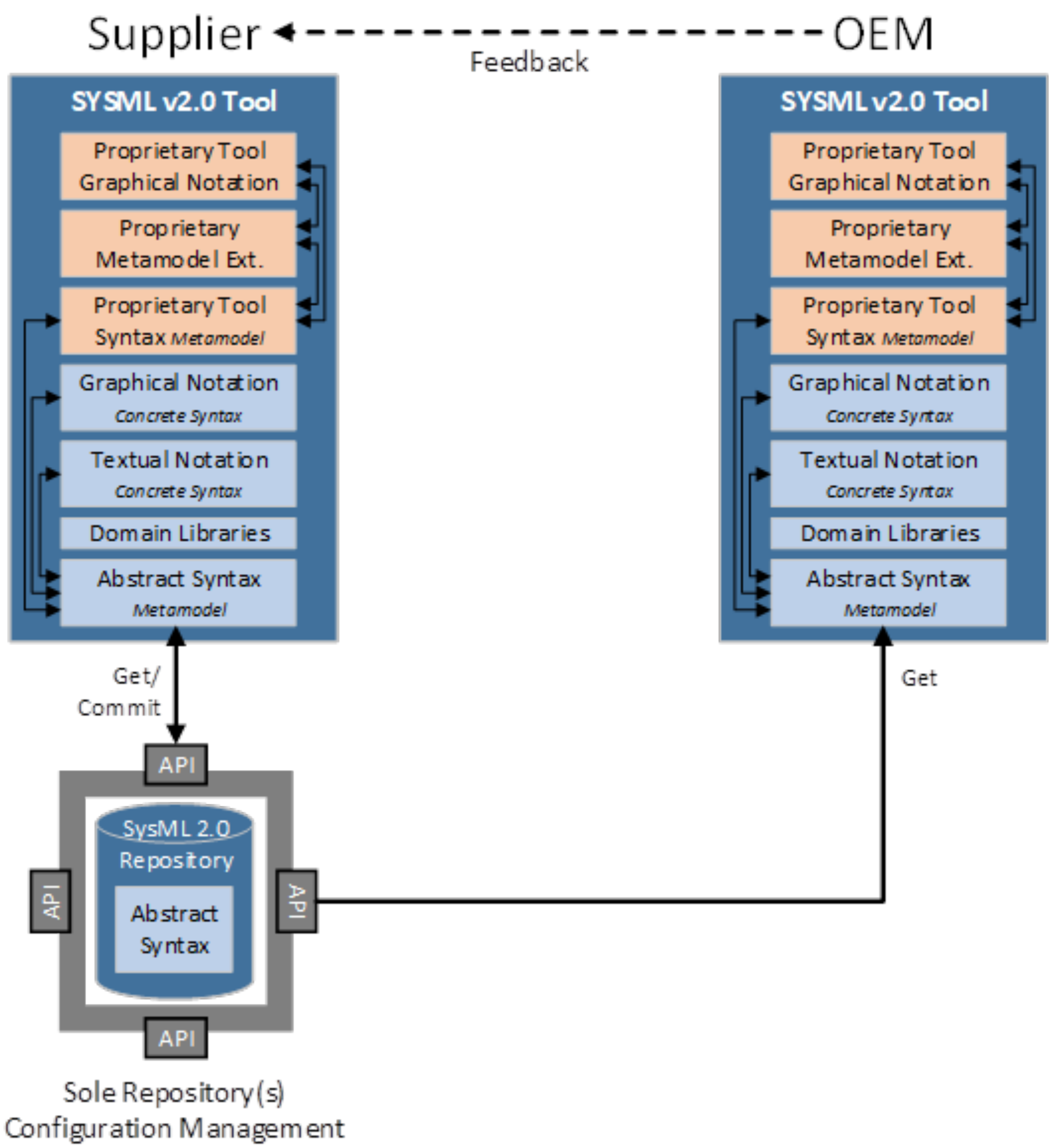

Figure 4 - Unidirectional: Shared Model Repository/Repositories Workflow

Bidirectional Collaboration Scenario

Multiple modeling parties make changes to separate copies of the model and exchange updated model(s). This scenario enables multiple parties to directly edit the model, but it is a challenging and tedious environment for change management.

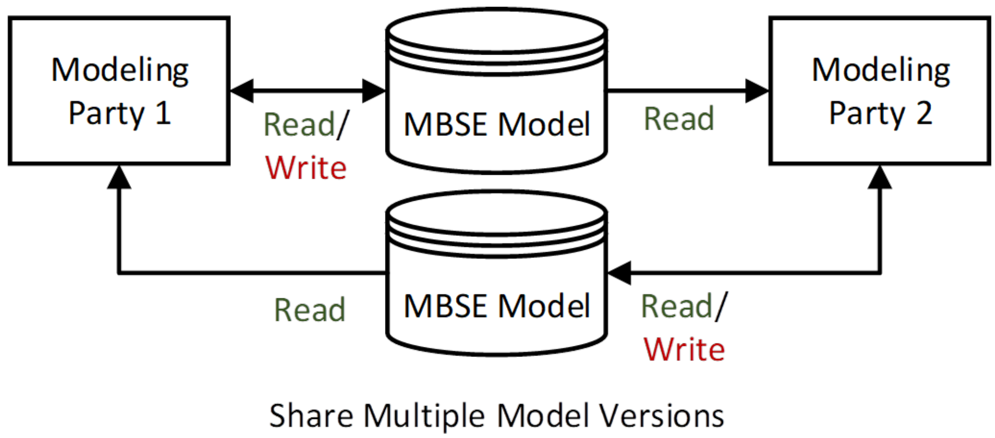

Figure 5 - Bidirectional Collaboration

Bidirectional Collaboration Workflows

This scenario is based on a bidirectional exchange of model data, which presumes only a single party is modifying the model data that is being exchanged. After exchange, another party could modify model data prior to yet another exchange. Two SysML v2 workflows were identified for use in bidirectional collaboration.

- **Project Interchange File(s).** A SysML v2 project interchange file can be used to archive the SysML v2 data repository to be shared with another modeling party. The receiving party can make changes to the model and then reshare the modified model as a project interchange file with the other party. This enables all modeling parties to make direct changes in the model without requiring a shared IT infrastructure. However, because modeling parties are working on separate copies of a model, the changes from all collaborating parties must be rolled into a "golden" copy of the model, which also does not require a shared IT infrastructure.

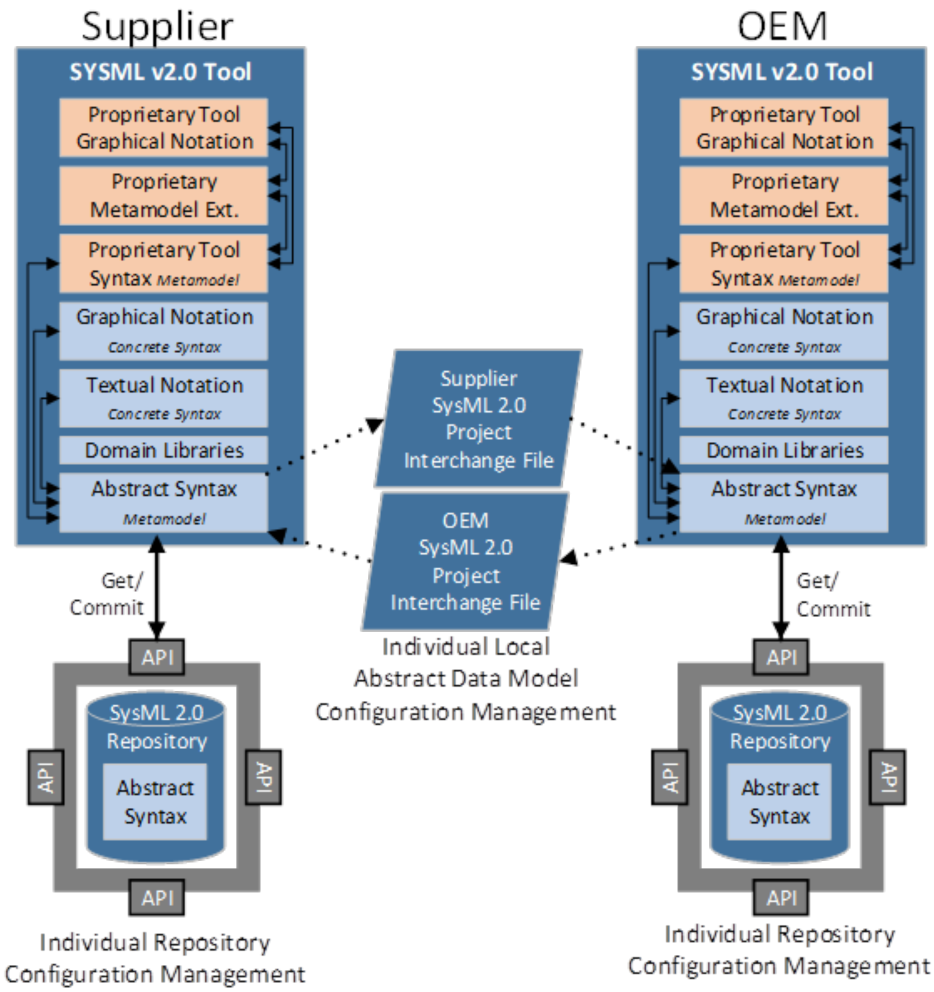

Figure 6 - Bidirectional: Project Interchange File(s) Workflow

- **Shared Model Repository/Repositories.** A shared model repository can also facilitate bidirectional collaboration where only a single modeling party makes changes to a model version. This can be through use of model branches for each modeling party, followed by a branch merge process. This workflow is technically preferred, as compared to project interchange files, because it provides live access to the most recent model data. However, it requires a shared IT storage infrastructure, which is not easy to set up between separate organizations given cybersecurity constraints, as well as management of contractual agreements of data access and data/IP ownership.

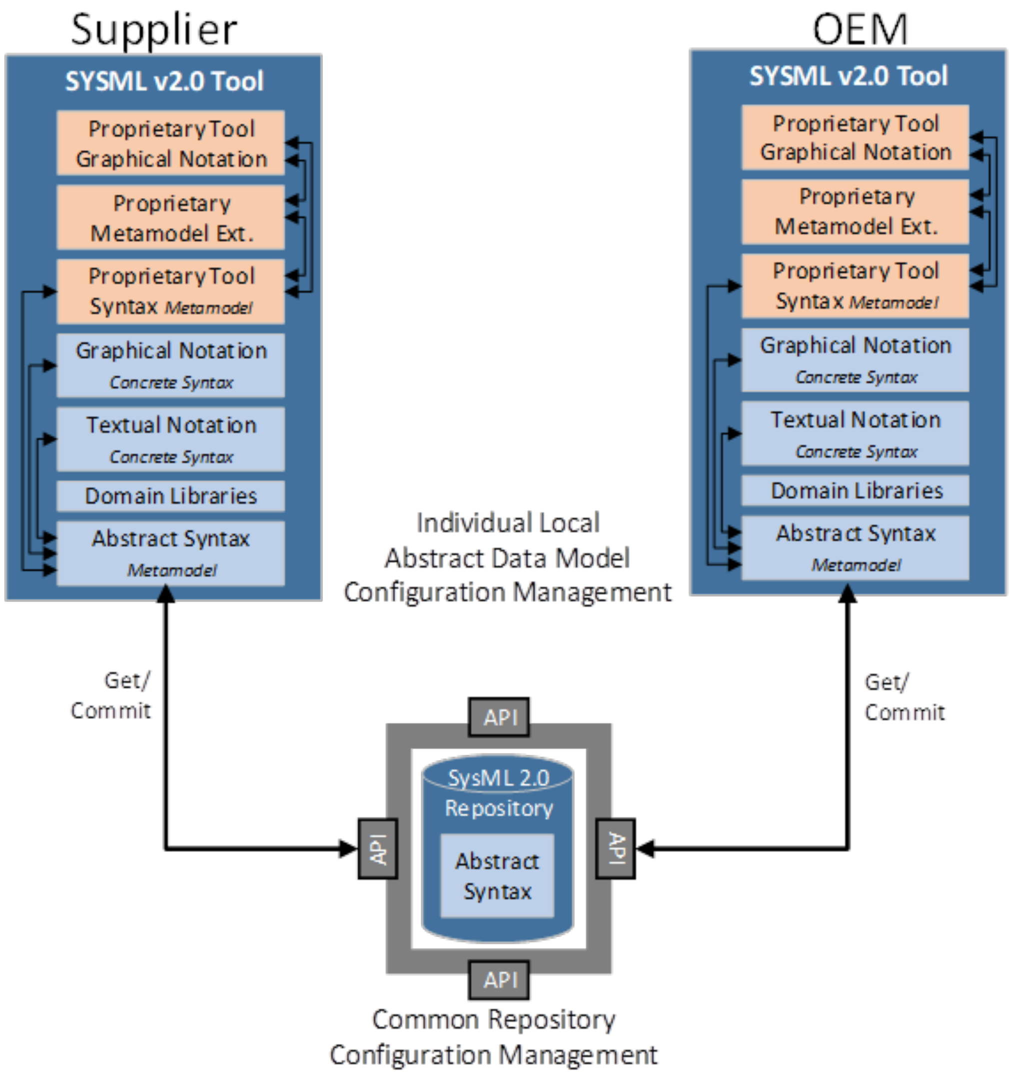

Figure 7 - Bidirectional: Shared Model Repository/Repositories Workflow

Joint Authorship Collaboration Scenario

Multiple parties directly edit the same model. This is a rich (i.e., focused, active, and productive) form of collaboration and requires a shared IT data storage environment with user access controls for modeling parties.

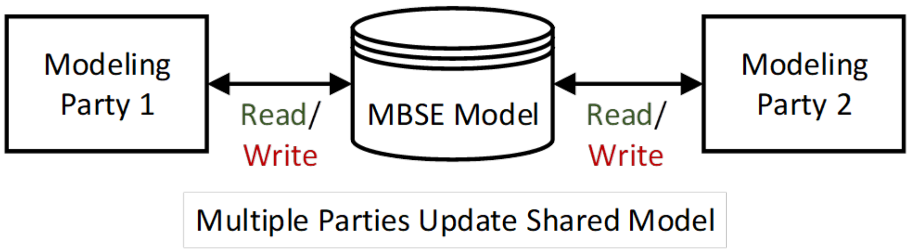

Figure 8 - Joint Authorship Collaboration

Joint Authorship Collaboration Workflows

Two SysML v2 workflows were identified for use in joint authorship collaboration. This scenario is based on a joint management of model data, which presumes more than one party is modifying the model data. This is a technically desirable scenario given the live access to model data, but both workflows require a shared IT storage infrastructure, which is not easy to set up between separate organizations given cybersecurity constraints.

- **Shared Model Repository/Repositories.** This provides the most straightforward solution for multiple parties to jointly collaborate. However, it doesn't protect IP as it allows all collaborating parties to access all model data unless a user access control solution is implemented around the data repository.

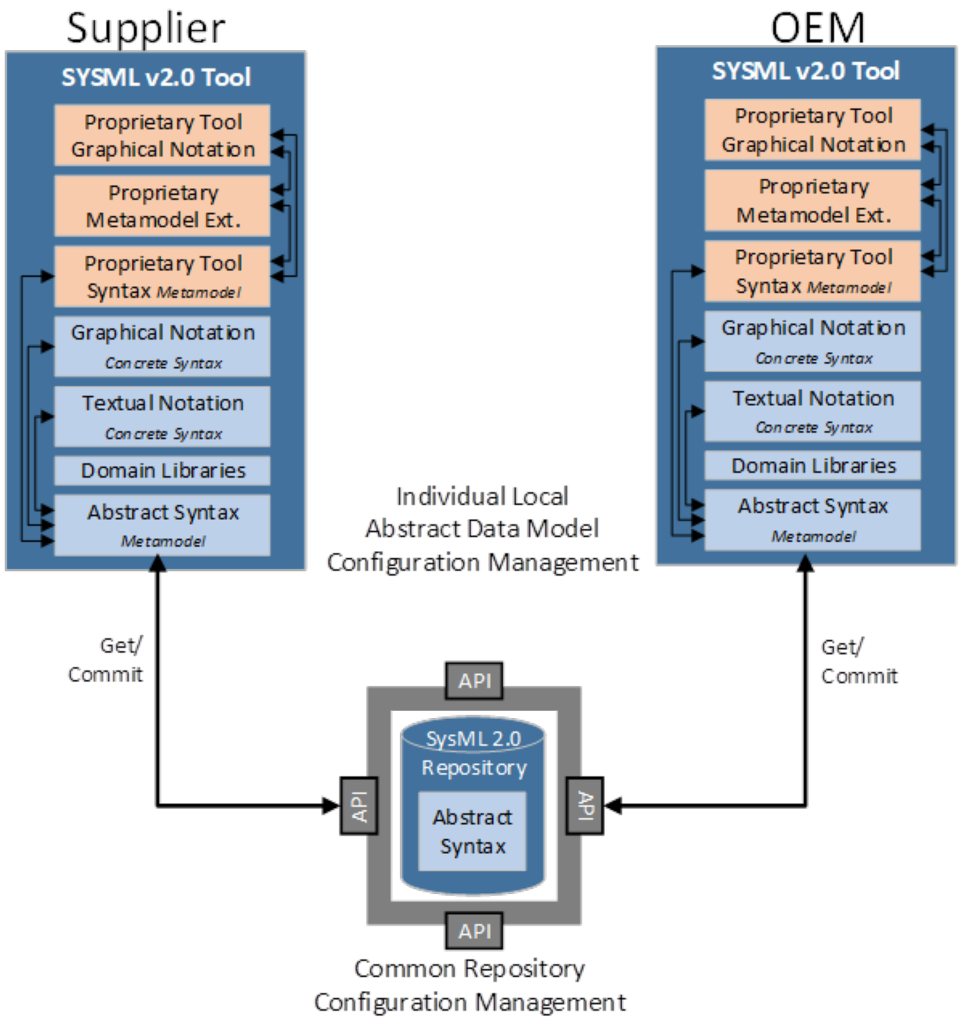

Figure 9 - Joint Collaboration: Shared Model Repository/Repositories Workflow

- **Referenced Model Repository/Repositories.** A project usage can be set up in a model repository that provides access to another repository. This workflow enables a level of data protection where a modeling party can access and use a model from another party without sharing their own model data.

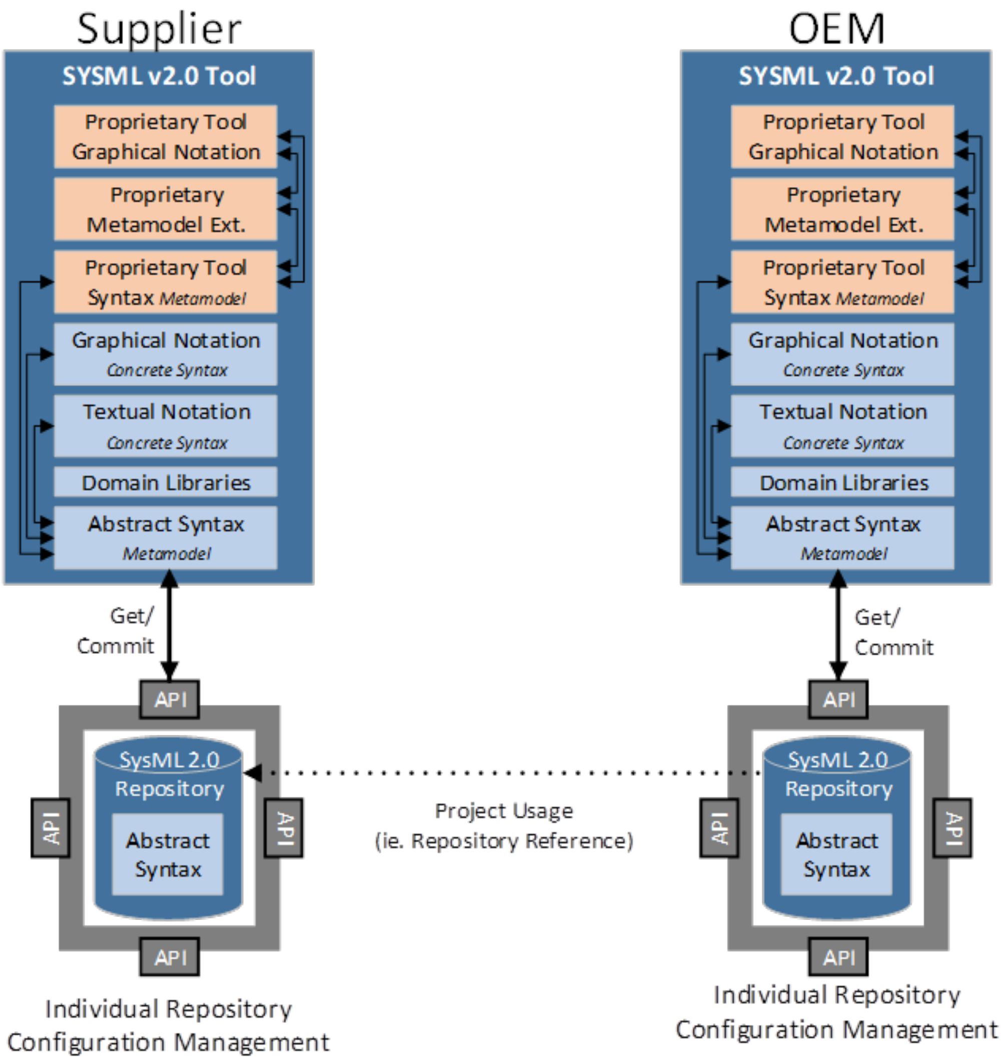

Figure 10 - Joint Collaboration: Referenced Model Repository/Repositories Workflow

Platform Co-Simulation Collaboration Scenario

This is the same as the Joint Collaboration scenario, except that it also employs an integrated digital environment between modeling parties to ensure a low-latency, high-performance connected experience to support live simulation.

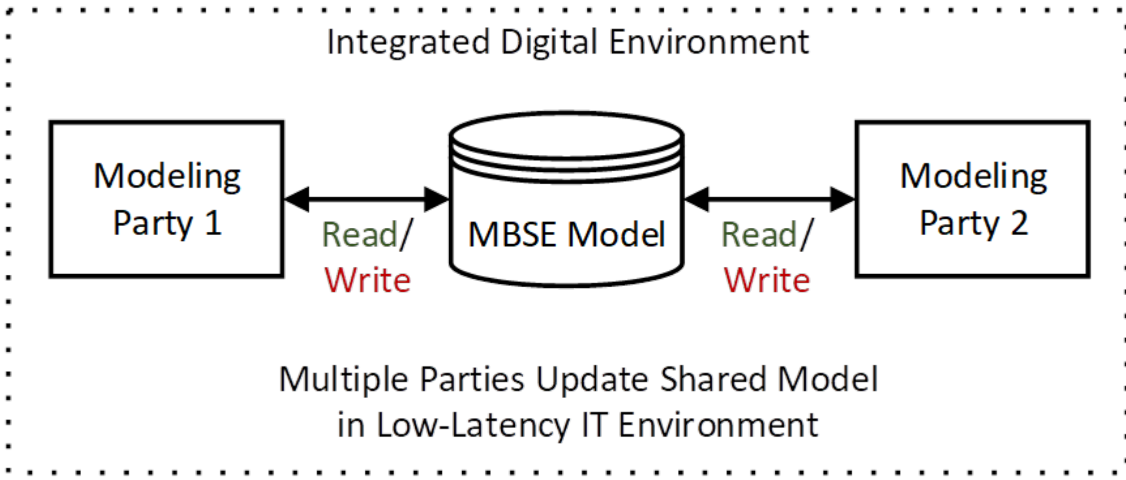

Figure 11 - Platform Co-Simulation Collaboration

Platform Co-Simulation Workflows

As noted above, the platform co-simulation scenario is similar to the joint collaboration scenario but adds an integrated digital environment. This scenario requires the most demanding IT infrastructure with low latency. The following two SysML v2 workflows were identified for use in platform co-simulation.

- **Shared Model Repository/Repositories.** Behavioral models can be created and managed by their respective owner organizations and then integrated in a higher-level model. The set of models can support co-simulation in the integrated digital environment.

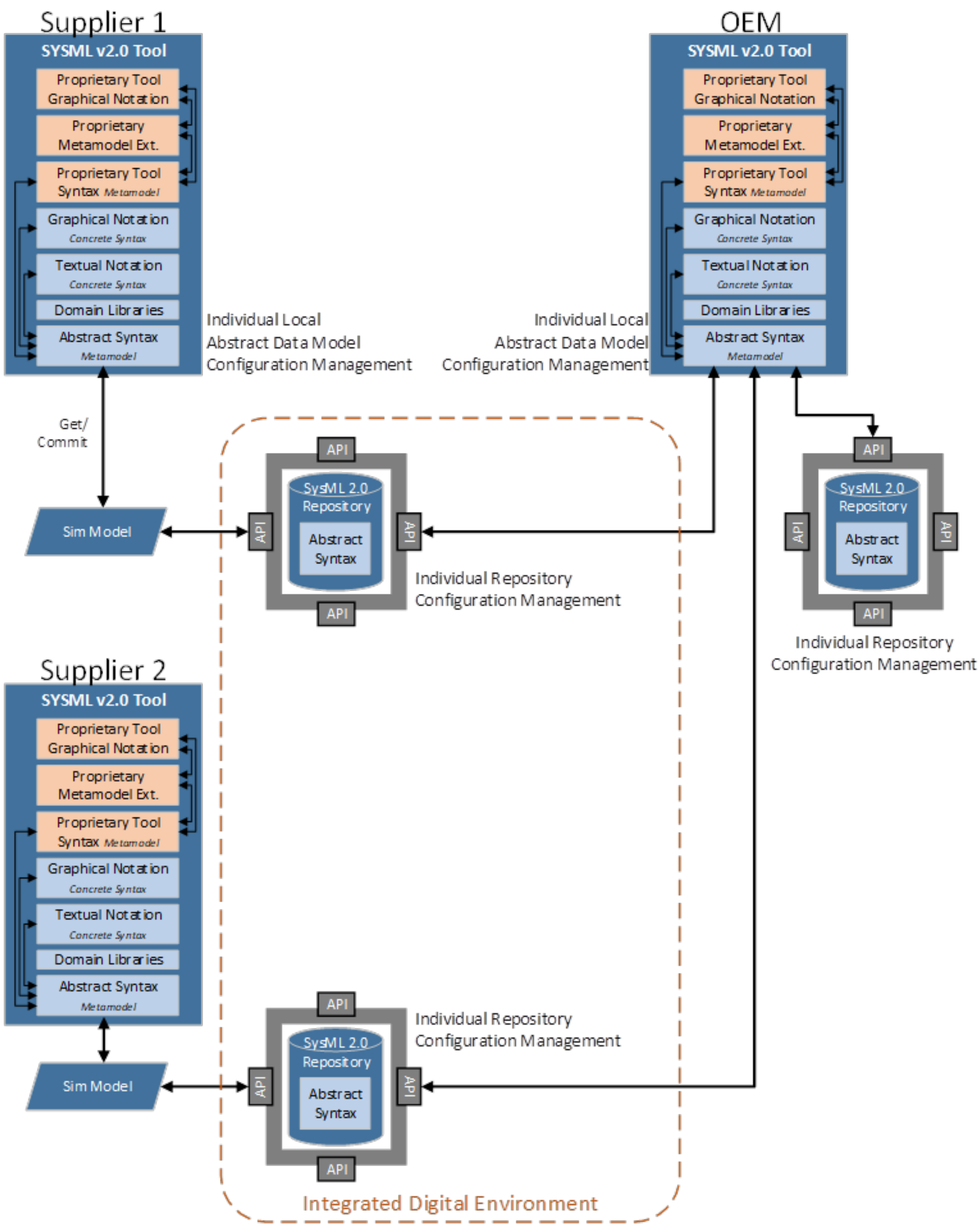

Figure 12 - Co-Simulation: Shared Model Repository/Repositories Workflow

- **Referenced Model Repository/Repositories**. The integrated model interfaces to separate system models through use of model references. This potentially enables the simulation to be managed through a single repository, which is referenced to dependent repositories.

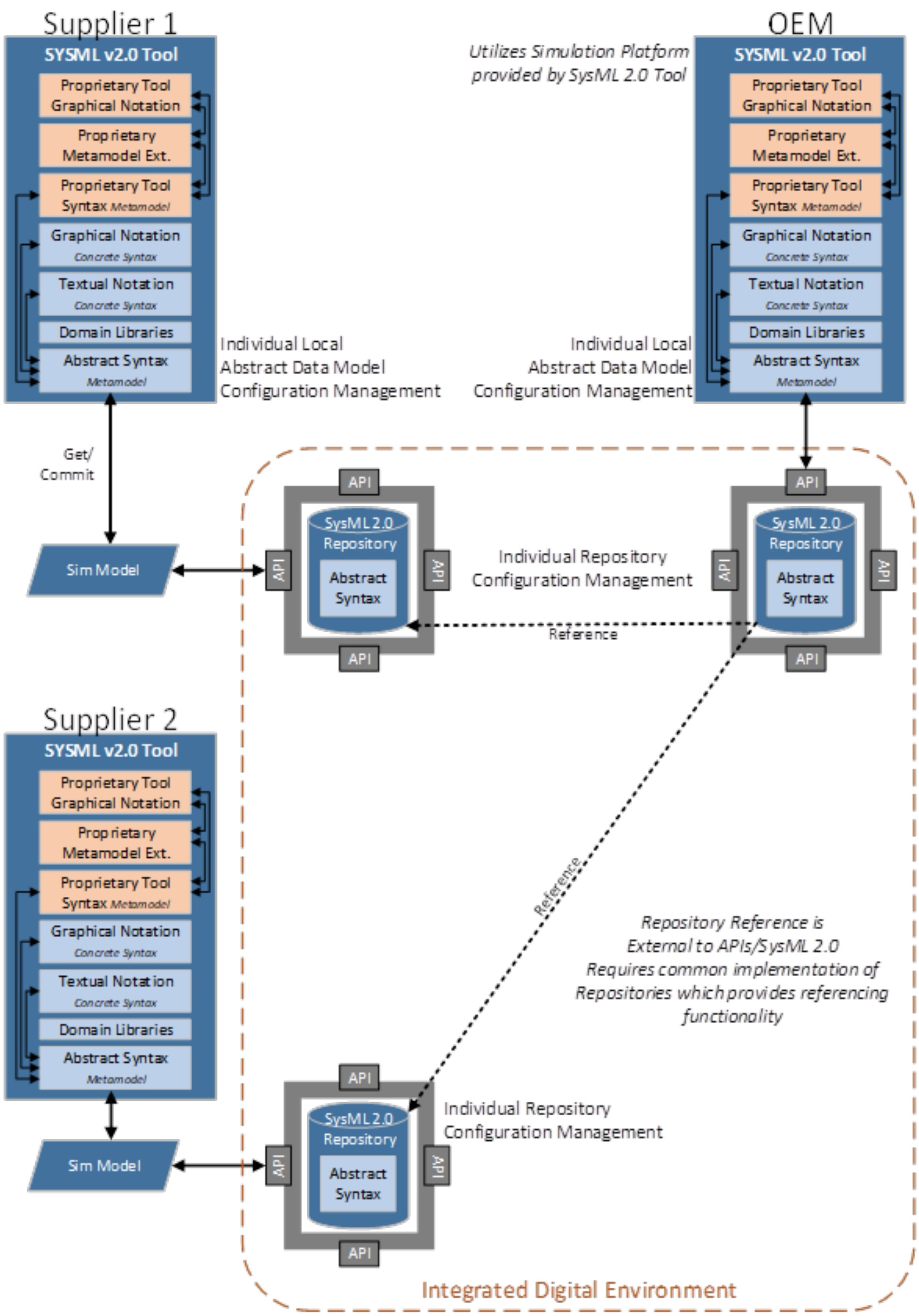

Figure 13 - Co-Simulation: Referenced Model Repository/Repositories Workflow

Team-Preferred and Most Likely Rollout Workflow/Scenario

The workflows that utilize shared model repositories are the technically preferred solution. However, the AD PAG MBSE project team members know it is very difficult to establish a shared IT infrastructure across all internal and external modeling parties. This is due to the aggressive cybersecurity constraints imposed to protect company IP and classified data in the case of defense projects.

Consequently, the team is confident that the most likely initial rollout of SysML v2 will leverage the project interchange files. While this does not provide the richest form of collaboration, it is better aligned with cybersecurity constraints for available IT infrastructure.

In the future, the AD PAG MBSE team members would like to see transition to the joint authorship collaboration scenario, a rich form of collaboration.

Transition to SysML v2 – Challenges for A&D

Adoption of SysML v2 by the A&D industry will almost certainly occur as the tools become available. However, transitioning to this new standard will likely be delayed to future 2.x releases because SysML v2.0 currently presents numerous challenges and considerations. This section explores the challenges and outlines the implications for businesses adopting this specific SysML v2.0 release.

Rolling Out SysML v2 and Upgrading Models

Airframe manufacturers are unlikely to convert existing artifacts or legacy in-service projects to SysML 2.0. The rationale for this approach lies in the scale and complexity of existing projects. Aircraft development generates an immense volume of models and artifacts, often spanning thousands of interconnected components and systems. Transitioning these into SysML 2.0 would incur significant costs and effort, with limited immediate benefits. These costs include retraining teams, revalidating models, and resolving compatibility issues within already established projects. Consequently, businesses will most likely reserve SysML v2 adoption for new product introductions (NPIs), leveraging the new standard's capabilities from the outset while leaving existing projects unchanged.

Maintaining Coexistence of SysML v1.x and v2

The aerospace sector's long product lifecycles demand the simultaneous support of SysML v1.x and v2. Aircraft often remain in service for decades, and projects tied to these platforms will continue to rely on SysML v1.x. At the same time, new projects will adopt SysML v2, capitalizing on its improvements in expressiveness, precision, and collaboration capabilities. This dual-support strategy necessitates maintaining expertise and infrastructure for both standards, including separate modeling environments, training programs, and governance frameworks. Solution providers will need to provide backward-compatible solutions and facilitate integration between SysML v1.x and v2.0 models, ensuring businesses can manage this coexistence effectively without significant disruptions.

Leveraging Legacy Artifacts

Aircraft design rarely starts from a blank slate. New projects often draw heavily on the models, data, and artifacts of previous products, incorporating lessons learned, proven designs, and shared architectural elements. Even when a project adopts SysML v2, referencing or reusing elements of earlier SysML v1.x-based models will remain essential. This interdependency highlights the need for robust methods to integrate and reference legacy artifacts within SysML v2 environments. Without such capabilities, businesses risk duplication of effort, loss of valuable design insights, and inconsistencies between related projects. Although a seamless bridge between SysML v1.x and v2 may not be possible due to differing metamodels, community methodologies to manage migrations are clearly necessary to enable businesses to maximize the value.

Collaborating During Transition

Collaboration between airframe OEMs, partners, and suppliers introduces additional complexities. Organizations will not adopt SysML v2 together at the same time, and often they are tied to toolchains tailored to their specific needs. Consequently, even if an OEM transitions to SysML v2, its suppliers or partners may still rely on SysML v1.x. Ideally, models could be translated bidirectionally between SysML v1.x and v2; however, this MBSE project team investigation has concluded this is not a feasible solution due to the fundamental differences between the standards and associated data models.

An alternative is to consider SysML v1.x and v2 models as separate languages that are connected by a third-party digital thread solution, similar to how Arcadia/Capella models are connected to SysML v1 models today. Effective model interoperability will be critical to ensure seamless collaboration across the supply chain during the transitionary period. Furthermore, businesses will need to establish clear communication protocols and standards for sharing models in mixed SysML v1.x and v2 environments to minimize the risks of misinterpretations or errors.

Enabling Product Lines and Common Architectures

The aerospace sector's emphasis on product lines, platforms, and common architectures further complicates the transition. Businesses are building reusable models to streamline the development of future products. These models often represent core components, subsystems, or design patterns used across multiple projects. Migrating these foundational models to SysML 2.0 is vital to maintaining their relevance and utility.

The process of transitioning product line models requires careful planning to ensure that key relationships, constraints, and dependencies are preserved. To facilitate this process, the industry requires tools that automate model conversion, ensuring consistency and accuracy while minimizing manual effort.

To mitigate risks associated with uncoordinated migrations, a standardized approach to transitioning from SysML v1.x to v2 is necessary. The OMG has already provided guidance as part of the SysML v2 standard. However, it is imperative for the A&D industry to collaborate on refining and adopting these guidelines. Without a unified approach, businesses may pursue disparate migration strategies, potentially undermining interoperability and reusability across the industry.

Considerations External to the Standard's Scope

While SysML v2 addresses key interoperability challenges inherent in SysML v1.x, such as improved tool compatibility through its API and textual syntax, there remain significant considerations outside the scope of the standard that could impact its deployment. This section explores several key external factors: absence of a universally recognized metamodel for systems architecture (i.e., bridging organizational methodologies), lack of standardized metadata/contractual frameworks for collaboration and model management, questionable IT infrastructure support for SysML v2 collaboration, and limited support (two of four) of the pillars required for seamless interoperability.

Bridging Organizational Methodologies

A significant challenge in collaborating in any systems model language—and one that continues in deploying SysML v2—is the missing link between diverse organizational methodologies and IT infrastructures. Different organizations often adopt distinct frameworks for systems architecture modeling, such as the Unified Architecture Framework (UAF), Arcadia framework, or even a custom in-house methodology and associated metamodel.

Additionally, each organization or tool deployed by the organization determines how to create shared data environments, enforce data governance measures, and organize the data repository. Although SysML v2 models developed using these frameworks are technically compatible (thanks to the standardization of syntax and communication protocols), the underlying semantics can vary significantly.

For instance, a SysML v2 UAF model developed in a web-based tool with "Cloud" synchronization might be syntactically interoperable with a SysML v2 model created using local repositories, different version control mechanisms, and different architecture frameworks. However, the metamodel incompatibility, semantic differences, and divergent IT infrastructures, mean that tools or processes tailored to one framework cannot automatically comprehend or utilize models from the other without substantial remapping. This situation necessitates manual efforts to align and translate between metamodels, semantics, and infrastructure differences undermining the benefits of interoperability that SysML v2, as a step change toward seamless interoperability, aims to provide.

The crux of this issue is the absence of a universally recognized metamodel–semantic alignment approach for systems architecture and recognized IT infrastructure principles that encapsulate industry commonalities. Such a metamodel and its approach to semantic recognition (e.g., use of ISO/IEC 81346 RDS) would serve as an agnostic foundation free from organizational or proprietary biases and ideally be both standardized and widely available. Without this common semantic grounding, organizations face persistent hurdles justifying the migration from SysML v1 to v2, as the promises of improved interoperability appear to be unfulfilled in practice.

Standardizing Metadata and Collaborative Contracts

Another significant external consideration is the lack of universally agreed-upon frameworks for exchanging systems architecture models and associated metadata. The current landscape is fragmented, with organizations using a variety of modeling tools ranging from SysML v1.x tools with proprietary syntax to non-SysML-compliant tools such as MathWorks Systems Composer, Capella, etc. This diversity extends to digital assets management, which often is handled on a case-by-case basis, complicating collaboration both within and between organizations.

In an ideal scenario, standardized metadata would be created according to a digital metadata metamodel, such as the MoSSEC ISO 10303-4443 standard. A metadata metamodel provides a universally agreed-upon syntax for handling digital objects, specifying access rights, defining ownership, and providing guidelines for how elements should be managed by partner organizations. Embedding such metadata and contractual information within the models is crucial for enabling secure and seamless model management across organizational boundaries, irrespective of the specific toolsets in use.

The absence of standardized metadata and collaborative contracts hampers the ability to achieve true interoperability. Even if technical compatibility is achieved through standardized syntax and communication protocols, the lack of agreed-upon metadata standards leads to inefficiencies and potential security risks because organizations must negotiate and manage these aspects individually.

Establishing IT Infrastructure to Support SysML v2 Collaboration

SysML v2 collaboration can be achieved only if the IT infrastructure is established to provide and govern access by all collaborating parties. Access control is needed for collaborators internal to an organization, and the control becomes even more challenging when the environment is opened to external organizations. Cybersecurity protection objectives are necessary, but they conflict with collaboration objectives. The non-comprehensive list of IT infrastructure topics listed below are to help guide SysML v2 rollout initiatives:

- Shared environment – shared data repository storage and access to its RESTful APIs: This can range from a locally-hosted storage space to a remote "Cloud" solution. Collaboration infers non-centrally located parties; therefore, solutions need to be established to open access (e.g., Virtual Private Networks (VPNs), public "Cloud" with secure authentication). Data read/write performance must support responsive, real-time access to model data as it is being interrogated.
- Data governance: User access controls are likely necessary because it is typical that access to all data is not granted to all collaborating parties. A layer of data governance beyond basic user authentication needs to be applied to the collaboration environment. Policies need to be established to govern what organizations have access to which portions of the shared data. Those policies then need to be enacted by the appropriate infrastructure solutions, which can range from separated shared environments where all users given access to an environment can access all data to common shared environments where users can only access a subset of data in the environment to which they have been granted.
- Change management: While the SysML v2 data repository provides limited capabilities for change management through branches, it is anticipated that an external change management solution is required to manage major "golden" baselines used for significant development and production milestones. Additionally, if a unidirectional or bidirectional exchange of project interchange format is used instead of a shared repository, those exchanges need to be managed outside of the data repositories.

- Change control: Changes are controlled to a varied level throughout the development lifecycle, and this must be planned and implemented using a change control solution. For example, it is common to allow unapproved changes in early development, but later in development, changes are subjected to approval prior to implementation.

Supporting the Four Pillars Required for Seamless Interoperability

For any organization to achieve seamless interoperability in accessing knowledge within their digital system, the following four pillars of standardization are considered useful:

1. Standardized semantics: A common understanding of the meaning and relationships of model elements across different frameworks and organizations
2. Standardized syntax (within the scope of SysML v2): A uniform language structure, which SysML v2 addresses through its standardized syntax requirements and underlying KerML foundation
3. Standardized metadata: Agreed-upon metadata tags/descriptors and management information for digital assets, enabling consistent handling and security measures
4. Standardized communication (within the scope of SysML v2): Protocols and mechanisms for exchanging models and data, facilitated by SysML v2 through its APIs and export capabilities

SysML v2 addresses the needs for standardized syntax and communication and semantics for model execution. However, where concurrent agreement in the use of standardized semantics and metadata (e.g., SysML usage to describe systems architecture) is absent, the value proposition of migrating to SysML v2 purely for interoperability benefits is questionable.

The challenges outlined in Pillars 1 and 2 are external to the SysML v2 standard but are critical for organizations seeking to justify the transition from SysML v1.x or other modeling tools based on SysML v2's interoperability benefits.

Conclusion of External Considerations

Differences in ways of working—both between organizations seeking to collaborate and in the approaches of different software solution providers—present significant challenges to interoperability. SysML v2 represents a significant step forward in addressing interoperability challenges through standardized syntax and communication mechanisms. However, external considerations pose substantial hurdles to the standard's effective adoption.

The ability of organizations to achieve true semantic and syntactic interoperability is limited due to the lack of a universally recognized metamodel and IT infrastructure alignment for systems architecture and the absence of standardized metadata/contractual frameworks. SysML v2 solution providers are encouraged to help organizations identify solutions for establishing an IT environment that supports collaboration between different solution provider tools. Pre-integrating third-party solutions for data governance and change management/control would make it easier for organizations to adopt the solution provider's SysML v2 tools and potentially offer a market advantage. However, a wide variety of needs exists across different organizations, so the tools must not require any specific solutions.

Addressing the external factors requires a collaborative effort beyond the scope of the SysML v2 standard. Industry stakeholders, including the SysML v2 developers, have a vested interest in promoting the development and adoption of standardized semantics and metadata frameworks. Only through a holistic approach that encompasses all four pillars of standardization can the full interoperability benefits of SysML v2 be realized to support the argument of migrating from existing tools and methodologies to the new standard.

Summary, Conclusions, and Recommendations

SysML v2 presents significant opportunities for the A&D sector, especially regarding enabling model-based collaboration. However, while SysML v2.0 offers promising advancements, it must address critical gaps to achieve widespread adoption in future v2.x updates. Some of the gaps will need to be addressed directly in the SysML v2 specification/standard, while other challenges require solutions provided by the solution providers. A collaborative approach among industry stakeholders and solution providers will be essential to realize the SysML v2 standard's full potential and to pave the way for the next iteration–SysML v2.1. The AD PAG MBSE team hopes that iteration will become the benchmark for MBSE collaboration. That will require OMG and solution providers to move more quickly as work is already underway for SysML v2.1.

Summary of Progress Toward the AD PAG MBSE Team "Requirements"

This section summarizes the headway made (as well as the gaps in progress) toward fulfillment of primary AD PAG MBSE project team "requirements."

Collaboration Using MBSE

SysML 2.0 demonstrates considerable advancements in enabling collaboration through features such as unidirectional and bidirectional model exchange, common exchange formats, and API capabilities. It appears to support all four collaboration scenarios outlined earlier in the paper, with the new API being a significant enabler for shared repositories.

However, the analysis also highlighted significant IT challenges around sharing repositories. Therefore, sharing models via project interchange files will likely continue to be common practice and reinforces the need for common file formats. Further gaps remain, such as the lack of a standardized graphical representation that retains diagram layouts critical to model comprehension. This presents a hurdle for achieving a common understanding when collaborating. Similarly, when it comes to data management, SysML 2.0 provides some useful features, such as the ability to segregate models for protection of intellectual property; yet, these features are not sufficient on their own to enable collaboration. Additional solutions will be required by the tools to ensure robust data governance.

MBSE Model Data Synchronization

The definition of a standard API that provides access to SysML v2 models enables many more opportunities for interacting with model data. The API effectively opens the door to synchronization with both COTS tools and in-house custom tools (unidirectional and bidirectional synchronization). However, it should be noted that a SysML 2 conformant tool is not required to support the standard API. The members of this project team recommend that solution providers support the API in order to enable effective collaboration with other SysML v2 tools which also implement the API. While direct collaboration is preferred, IT policy challenges can continue to inhibit a shared IT environment for data repositories. Therefore, the alternate method of hosting duplicate repositories in separate IT environments, shared via the SysML v2 File Interchange Format, remains an important synchronization use case.

Effective Model Data Management

SysML v2 meets many requirements for model data management, including serving as an ASoT, supporting change management, and enabling model differencing and branching. However, the lack of UUID support in textual representations limits the effectiveness of this representation for use as an exchange format and therefore collaboration. Additionally, it will be necessary for comprehensive change management capabilities to be available to support collaboration. This topic goes beyond the scope of the SysML v2 standard and needs to be addressed by complementary software tools provided by MBSE solution providers collaborating with industrial users or by other enterprise software solution providers such as the PLM companies.

Key Conclusions for A&D

The A&D industry will take great interest in the eventual release of SysML v2 and the initial release of supporting tools. Over the next few years, it will be vital to pilot the tools and the new language to fully understand the capabilities and gaps. SysML v2.0 is not released yet, and the current expected release date is sometime in 2025.

The A&D industry uses multiple frameworks that are required to support defense contracts. These frameworks will need to be updated to work with the new standard. Until that point, defense contractors (and likely the rest of the industry) will continue to use older versions of the frameworks and therefore SysML 1.x. The frameworks will begin to be transitioned only after the SysML v2.0 specification is stable and has been released. It will also take a while for the various commonly used frameworks to transition over to the new standard.

SysML v2.0 takes substantial steps toward meeting the AD PAG MBSE project team “requirements” for enabling collaboration. New features, such as the API, offer new avenues for streamlined workflows. However, unresolved gaps in the standard and external considerations related to collaboration limit the new standard’s ability to drive widespread adoption. It’s believed that the next iteration–SysML v2.1–is where these gaps will be addressed, and only then will this new language become the preferred standard for collaboration.

It is very encouraging that substantial solution provider support exists for SysML v2. That, along with interest from new market entrants, indicates growing confidence in the standard’s potential.

Recommendations

This section summarizes the AD PAG MBSE project team’s recommendations for the aerospace industry, solution providers, and SysML v2.1 developers.

Recommendations for the Aerospace Industry

- Adopt a phased rollout. Focus SysML v2 adoption on new projects while maintaining SysML 1.x for legacy initiatives. This phased approach reduces risks and ensures that ongoing projects are not disrupted. Focus on pilot projects and specific use cases to assess the standard’s effectiveness without disrupting ongoing workflows.

- Develop dual-standard capabilities. Establish internal capabilities to operate across both standards, including training teams, updating workflows, and integrating tools to support a mixed environment.
- Promote cross-organization collaboration. Engage early with suppliers and partners to align transition timelines and agree on standards for model exchange and interoperability.
- Standardize migration practices. Work with industry groups, such as OMG, to refine and adopt consistent methodologies for transitioning models. This coordination will ensure uniformity across the sector and reduce inefficiencies.
- Plan for long-term support. Invest in maintaining SysML 1.x expertise and infrastructure for as long as legacy projects remain active, ensuring continuity and reliability.
- Align with industry standards. Collaborate with industry groups and solution providers to ensure a unified approach to transitioning models and addressing identified gaps.
- Invest in training. Build internal expertise in SysML v2 to prepare for future adoption and transition, while maintaining capabilities for SysML 1.x to support legacy projects.

Recommendations for Solution Providers

By addressing the following considerations, airframe manufacturers, suppliers, and solution providers can effectively navigate the transition to SysML v2, unlocking its full potential for enhanced collaboration and innovation.

- Enable seamless interoperability. Develop features that allow SysML 1.x models to be integrated, referenced, or converted into SysML v2 environments with minimal loss of fidelity. Open SysML v2 “Cloud” environments so that they can be accessed by other SysML v2 tool vendors.
- Provide automation tools for migration. Create tools that automate the process of migrating SysML 1.x models to SysML v2, ensuring accuracy and consistency while reducing manual effort.
- Support bidirectional transformation. Ensure that models can be transitioned both forward and backward between SysML 1.x and v2, enabling businesses to adapt to mixed environments. It is understood there may be loss of functionality and there likely won't be full automation, requiring a human in the loop.
- Offer training and support. Provide comprehensive training resources and technical support to help businesses navigate the transition and leverage the full potential of SysML v2.

Recommendations for SysML v2.1

- Address graphical representation gaps. Ensure a standardized approach to storing and sharing graphical layouts and preserving critical visual context across collaborators.
- Improve change management. Extend UUID support to textual and graphical representations to facilitate collaboration with more means for robust model versioning and differencing.
- Standardize migration tools. Develop standardized tools and methodologies to transition models between SysML 1.x and v2, ensuring consistency across the industry.

About A&D PLM Action Group

The Aerospace & Defense PLM Action Group (www.ad-pag.com) is an association of aerospace and defense companies within CIMdata's globally recognized PLM Community Program, which functions as a ***PLM advocacy group*** to:

- Set the direction for the aerospace & defense industry on PLM-related topics that matter to members (*including promoting, not duplicating, the work of standards bodies*)
- Promote common industry PLM processes and practices
- Define requirements for common interest PLM-related capabilities
- Communicate with a unified voice to PLM solution providers
- Sponsor collaborative PLM research on prioritized industry and technology topics

CIMdata administers Group operations, coordinates research, and manages the progression of policy formulation.

About CIMdata

CIMdata, a global strategic management consulting firm, provides services designed to maximize an enterprise's ability to design, deliver, and support innovative products and services. For more than forty years, CIMdata has provided industrial organizations, providers of digital technologies and services, and investment firms with world-class insight, expertise, and best-practice methods on a broad set of product lifecycle management (PLM) topics and the digital transformation they enable. CIMdata also offers research, subscription services, publications, and education through certificate programs and international conferences. To learn more, visit www.CIMdata.com or email info@CIMdata.com.

