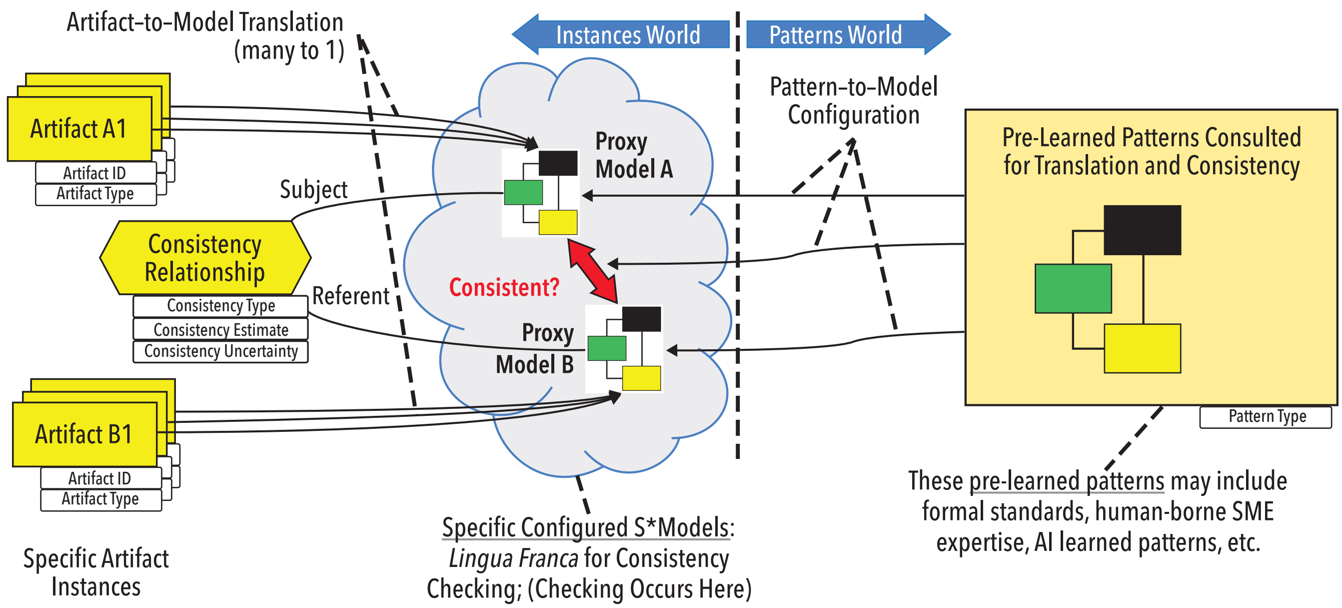
INSIGHT

This Issue's Feature:

Digital Engineering



Checking consistency of artifacts by translating them to proxy models that are consistency checked

Illustration credit: from the article *From Fragmentation to Federation: A Proposed Strategy for Advancing Digital Engineering Standards Development* by Celia S. Tseng, Joseph W. Marvin, William D. Schindel, Juan Carlos Mendo, and Terri W. Chan (page 11)

OCTOBER 2025
VOLUME 28 / ISSUE 5

INCOSE

A PUBLICATION OF THE INTERNATIONAL COUNCIL ON SYSTEMS ENGINEERING

INSIGHT

A PUBLICATION OF THE INTERNATIONAL COUNCIL ON SYSTEMS ENGINEERING

OCTOBER 2025 VOLUME 28/ ISSUE 5

Inside this issue

About This Publication

INFORMATION ABOUT INCOSE

INCOSE's membership extends to over 27,000 members and CAB associates and more than 200 corporations, government entities, and academic institutions. Its mission is to share, promote, and advance the best of systems engineering from across the globe for the benefit of humanity and the planet. INCOSE charters chapters worldwide, includes a corporate advisory board, and is led by elected officers and directors.

For more information, click here:
The International Council on Systems Engineering
(www.incose.org)

OVERVIEW

INSIGHT is the magazine of the International Council on Systems Engineering. It is published six times per year and features informative articles dedicated to advancing the state of practice in systems engineering and to close the gap with the state of the art. ***INSIGHT*** delivers practical information on current hot topics, implementations, and best practices, written in applications-driven style. There is an emphasis on practical applications, tutorials, guides, and case studies that result in successful outcomes. Explicitly identified opinion pieces, book reviews, and technology roadmapping complement articles to stimulate advancing the state of practice. ***INSIGHT*** is dedicated to advancing the INCOSE objectives of impactful products and accelerating the transformation of systems engineering to a model-based discipline.
Topics to be covered include resilient systems, model-based systems engineering, commercial-driven transformational systems engineering, digital engineering, artificial intelligence, natural systems, agile security, systems of systems, and cyber-physical systems across disciplines and domains of interest to the constituent groups in the systems engineering community: industry, government, and academia. Advances in practice often come from lateral connections of information dissemination across disciplines and domains. ***INSIGHT*** will track advances in the state of the art with follow-up, practically written articles to more rapidly disseminate knowledge to stimulate practice throughout the community.

PERMISSIONS

ARTICLE SUBMISSION insight@incose.net

Publication Schedule. ***INSIGHT*** is published six times per year. Issue and article submission deadlines are as follows:

- December 2025 issue – 1 September 2025
- February 2026 issue – 1 November 2025
- April 2026 issue – 2 January 2026
- June 2026 issue – 1 March 2026
- August 2026 issue – 1 May 2026
- October 2026 issue – 1 July 2026

For further information on submissions and issue themes, visit the INCOSE website: www.incose.org

 ISSN 2156-485X; (print) ISSN 2156-4868 (online)

ADVERTISE

Readership
INSIGHT reaches over 27,000 members and CAB associates and uncounted employees and students of more than 130 CAB organizations worldwide. Readership includes engineers, manufacturers/purchasers, scientists, research and development professionals, presidents and chief executive officers, students, and other professionals in systems engineering.

Issuance	Circulation
2025, Vol 28, 6 Issues	100% Paid

Contact us for Advertising and Corporate Sales Services
We have a complete range of advertising and publishing solutions professionally managed within our global team. From traditional print-based solutions to cutting-edge online technology the Wiley-Blackwell corporate sales service is your connection to minds that matter. For an overview of all our services please browse our site which is located under the Resources section. Contact our corporate sales team today to discuss the range of services available:

- Print advertising for non-US journals
- Email Table of Contents Sponsorship
- Reprints
- Supplement and sponsorship opportunities
- Books
- Custom Projects
- Online advertising

Click on the option below to email your enquiry to your nearest office:

- Asia and Australia corporatesalesaustralia@wiley.com
- Europe, Middle East and Africa (EMEA) corporatesaleseurope@wiley.com
- Japan corporatesalesjapan@wiley.com
- Korea corporatesaleskorea@wiley.com

USA (also Canada, and South/Central America):

- Healthcare Advertising corporatesalesusa@wiley.com
- Science Advertising Ads_sciences@wiley.com
- Reprints Commercialreprints@wiley.com
- Supplements, Sponsorship, Books and Custom Projects busdev@wiley.com

Or please contact: Marcom@incose.net

CONTACT

Questions or comments concerning:

Submissions, Editorial Policy, or Publication Management
Please contact: William Miller, Editor-in-Chief
insight@incose.net

Advertising—*please contact:* Marcom@incose.net

Member Services – ***please contact:*** info@incose.org

ADVERTISER INDEX October Volume 28-5

CORPORATE ADVISORY BOARD — MEMBER COMPANIES

3DSE Management Consultants
Advanced Systems Engineering, LLC
Aerospace Corporation, The
Airbus
Albers Aerospace
AM General LLC
Analog Devices, Inc.
ANSYS, Inc
Arcfield
Auburn University
Australian National University
AVIAGE SYSTEMS
Aviation Industry Corporation of China, LTD
BAE Systems
Bechtel
Becton Dickinson
Belcan Engineering Group LLC
BMT Canada
Boeing Company, The
Booz Allen Hamilton Inc.
Boston Scientific Corporation
BTS Software Solutions
California State University Dominguez Hills
Capgemini Engineering
Carnegie Mellon University Software Engineering Institute
Change Vision, Inc.
Colorado State University Systems Engineering Programs
Cornell University
Cranfield University
C.S. Draper Laboratory, Inc.
Cubic Corporation
Cummins, Inc.
Dassault Systèmes
Defense Acquisition University
Deloitte Consulting, LLC
Denso Create Inc
DENTSU SOKEN INC
Drexel University
Eaton
EMBRAER
FAMU – FSU College of Engineering
Federal Aviation Administration (U.S.)
Florida Institute of Technology
Ford Motor Company
GE Aerospace
General Dynamics
General Motors
George Mason University
Georgia Institute of Technology
Hitachi Energy
Honeywell Aerospace Technologies
Huawei Technologies Co. Ltd
Idaho National Laboratory
IQNOX, LLC
ISAE – Supaero
ISDEFE
IVECO Group
Jama Software
Jet Propulsion Laboratory
John Deere & Company
Johns Hopkins University
KBR, Inc.
KEIO University
L3Harris Technologies
Lawrence Livermore National Laboratory
Leidos
LEONARDO
Lockheed Martin Corporation
Los Alamos National Laboratory
Loyola Marymount University
Magna
ManTech International Corporation
Marquette University
Massachusetts Institute of Technology
MBDA (UK) Ltd
Medtronic
MetaTech Consulting Inc.
Missouri University of Science & Technology
MITRE Corporation, The
Mitsubishi Electric Corporation
Mitsubishi Heavy Industries, Ltd
Modern Technology Solutions Inc
National Aeronautics and Space Administration (NASA)
National Reconnaissance Office (NRO)
Naval Postgraduate School
Nissan Motor Co, Ltd
Northrop Grumman Corporation
Pacific Northwest National Laboratory
PARAMETRIC TECHNOLOGY GMBH PTC
Pennsylvania State University
Petronas International Corporation Limited
Prime Solutions Group, Inc
Project Performance International (PPI)
Purdue University
RealmOne
Redwire Space
Rolls-Royce
RTX
Saab AB
SAFRAN
SAIC
Sandia National Laboratories
Saudi Railway Company
SENSEONICS
Shanghai Formal – Tech Information Technology Co., Ltd
Shell
Siemens
Sierra Nevada Corporation
Singapore Institute of Technology
Southern Methodist University
SPEC Innovations
Stevens Institute of Technology
Strategic Technical Services LLC
Studio SE, Ltd.
Swedish Defence Materiel Administration (FMV)
Systems Planning and Analysis
Taiwan Space Agency
Tata Consultancy Services
Thales
The George Washington University
The University of Arizona
The University of Texas at Arlington
The University of Utah
Torch Technologies
TOSHIBA Corporation
Trane Technologies
Tsinghua University
UK MoD
UNCOMN
Universidade Federal De Minas Gerais
University of Alabama in Huntsville
University of Arkansas
University of California San Diego
University of Connecticut
University of Maryland
University of Maryland, Baltimore County
University of Maryland Global Campus
University of Michigan, Ann Arbor
University of New South Wales, The, Canberra
University of South Alabama
University of South-Eastern Norway (USN)
University of Texas at Austin
University of Texas at El Paso (UTEP)
US Department of Defense
Veoneer US Safety Systems, LLC
Virginia Tech
Volvo Cars Corporation
Volvo Construction Equipment
Wabtec Corporation
Wayne State University
Weber State University
Wichita State University College of Engineering
Woodward Inc
Worcester Polytechnic Institute (WPI)
Woven by Toyota, Inc.
Zuken, Inc

FROM THE EDITOR-IN-CHIEF

William Miller, insight@incose.net

We are pleased to announce the October 2025 *INSIGHT* issue published cooperatively with John Wiley & Sons as the systems engineering practitioners' magazine. The *INSIGHT* mission is to provide informative articles on advancing the practice of systems engineering and to close the gap between practice and the state of the art as advanced by *Systems Engineering*, the Journal of INCOSE also published by Wiley. The theme of this issue is digital engineering and follows the March 2022 *INSIGHT* also themed on digital engineering. We thank theme editor Frank Salvatore and the authors for their contributions; Frank served as one of the theme editors in the March 2022 *INSIGHT*.

We lead off with "The Digital Transformation of Standards: Why OEMs Must Act Now" by Leslie McKay. Standards remain stuck in the 20th century published as static PDF documents that are increasingly inadequate for modern digital workflows. They're still "documents written by experts, for experts," human-readable text that requires manual discovery, interpretation, transformation, and implementation. Original equipment manufacturers (OEMs) have made it clear that standards must become machine-readable and machine-interpretable, delivered as data that can integrate seamlessly into design and product lifecycle management (PLM) systems. Research by SAE International reveals the transformative benefits that digital standards would deliver—improved discoverability, traceability, efficiency, reusability, accessibility, and quality. The aerospace industry has taken the lead in forming the Digital Standards Alliance (DSA) as an industry-led membership group focused exclusively on accelerating the migration to digital standards.

"From Fragmentation to Federation: A Proposed Strategy for Advancing Digital Engineering Standards Development" is authored by Celia Tseng, Joe Marvin, Bill Schindel, and Juan Carlos Mendo. No comprehensive framework exists to guide digital engineering (DE) across the full ISO/IEC/IEEE 15288 system lifecycle process. Data and models are fragmented across engineering tools, enterprise systems, and lifecycle processes. This fragmentation limits organizations' ability to integrate, scale, and deliver consistent value across increasingly complex ecosystems. The authors call for INCOSE to adopt a strategic standardization role—working with other standards development organizations (SDOs) to define interoperability use cases, align lifecycle models, and model-based systems engineering (MBSE) standards, and develop a shared reference framework for interoperability. They recommend the INCOSE agile systems engineering life cycle management (ASELCM) pattern as a foundation for aligning lifecycle processes with model-based technical exchanges through a federated digital thread to enable seamless, cross-enterprise interoperability—and for empowering organizations to realize the speed, scale, and economic advantage promised by digital transformation.

"Digital Engineering: Transforming the Research into the Business Roadmap" by Salvatore Bruno, Dr. Carol Woody, Steve Henry, and Celia Tseng follow-up the article from the March 2022 *INSIGHT* "Digital Engineering Measures: Research and Guidance" by Tom McDermott, Kaitlin Henderson, Alejandro Salado, and Joseph Bradley. Organizations are embedding digital engineering and model-based engineering into their systems engineering lifecycles to achieve cost savings, higher product quality, and earlier delivery. However, understanding the success of their digital transformation investment, has been lacking from a business perspective. To address this requested need, government, university, and industry experts have collaborated with the INCOSE Measurement Working Group to enhance the issuance of the initial Digital Engineering Measurement Framework (v1.1) to include measures that track, and report results related to digital engineering business operations and environmental benefits. This article describes the approach used to mature the guidance document, the benefits of the new measures, and recommendations for the sequential release.

"System Thinking in the Design of Enterprises with Support of SEREA (Systems Engineering Reference Enterprise Architecture)" by Hugo Ormo describes the development of SEREA and its tailoring. One of the biggest challenges in designing complex systems is to design and specify the components in such a way that they contribute to the overall functionality of the system. Specialization, cost efficiency, or other reasons require that these components be co-developed, verified, and delivered by various suppliers, and then integrated, verified, and validated by the integrator. This makes the approaches of systems engineering crucial for integrators (OEMs). As the complexity of the systems to be managed increases, so does the complexity of designing, maintaining, and further developing these approaches. The transition from document-based to model-based operational architectures presents a comparable challenge. SEREA is listed to be designated as an INCOSE technical product, offering a model-based reference enterprise architecture based on the unified architecture framework (UAF). With SEREA as a basis, a customized enterprise architecture can be created. Since SEREA is based on ISO 15288, it is particularly suitable for companies dealing with the life cycle management of complex systems.

"PBSE Data Initialization Framework and Practice by Using LLM" by Degang Liang and Baoyu Dong" received a best paper award at the 35th Annual INCOSE International Symposium, Ottawa, Canada, 26-31 July 2025. Their paper explores the application of artificial intelligence for systems engineering (AI4SE) in real-world engineering projects by leveraging large language models (LLMs) to develop a methodology that reduces the deployment threshold of pattern-based systems engineering (PBSE) for enterprises and enhances the efficiency of instance generation. PBSE builds upon MBSE by utilizing engineering patterns, validated in advance, to enhance the efficiency and quality of data production. However, the authors observed certain challenges performing PBSE, such as the barriers to initializing the product S* model and the low efficiency in generating instances.

"Solving the Octopus Problem in Digital Engineering – Towards Reusable Asset Specification 3.0" is authored by Matthew Hause, Sriram Krishnan, Mark Petrotta, Michael Shearin, and Tomas Vileiniskis. Engineering projects often face the plight that insights remain locked in individual models and documents, never reaching others who could benefit subsequently. Teams repeatedly reinvent the wheel – recreating models, consuming precious time, and compounding technical debt. To break this cycle, digital engineering needs curated, discoverable assets. This article outlines updating the OMG Reusable Asset Specification (RAS 3.0) to enable better discovery through structured metadata, searchable asset catalogs, and curation services, and accelerating reuse, collaboration, and scalability.

"The Decision Analysis Data Model" by Greg Parnell, Bob Kenley, Devon Clark, Jared Smith, Frank Salvatore, Chiemeke Nwobodo, and Sheena Davis describes a decision analysis data model (DADM) developed in model-based systems engineering software to provide the process, methods, models, and data to support decision management. DADM can support digital engineering for waterfall, spiral, and agile development processes. Their paper describes the decision management processes and provides the definition of the data elements. DADM is based on ISO/IEC/IEEE 15288, the INCOSE *Systems Engineering Handbook*, the Guide to the Systems Engineering Body of Knowledge, the Data Management Body of Knowledge, systems engineering textbooks, and journal articles. The DADM was developed to establish a decision management process and data definitions that organizations and programs can tailor for their system life cycles and processes. The DADM can also be used to assess organizational processes and decision quality.

"An Implementation of DADM Using Semantic Interoperability and Visualization" by Thomas Hagedorn, Daniel Dunbar, Joshua Bernstein, and Mark Blackburn describes a DADM implementation called the armaments interoperability and integration framework (IoIF), a configurable software framework that supports engineering analysis and decision-making. IoIF uses linked data to facilitate data interoperability across mission, system, and discipline-specific models. At its core is a semantic representation of a system and a formalized model of the system analysis process. This can be applied to decision analysis as described by DADM. Using IoIF and linked front ends, a user can incorporate engineering analyses into analytic workflows to aid in decision making.

"The Need for a Shared Vocabulary of Digital Engineering" by Joe Gregory, Clarence (Moe) Moreland, James S. Wheaton, and Celia Tseng addresses some of the key terminology challenges facing DE practitioners and describes how a machine-readable ontology can help to create a shared understanding of DE and enable more effective implementation of DE practices.

"Embedding Digital Engineering into the Classroom" by Joe Gregory and Alejandro Salado applies integrating engineering data in a digital thread in engineering education to expose students to authentic, connected workflows across multiple domains. The digital engineering factory (DEF), developed at the University of Arizona, is a web-based platform designed to support systems and software engineering students by providing integrated access to tools for project management, requirements, modeling, analysis, verification, and test planning. In a classroom deployment, students work in role-defined teams using the DEF to manage their project data and participate in model-based design and review. The DEF provides students with a clear view of the full engineering lifecycle and enables automated grading based on traceable, semantically validated data.

"Accelerating Digital Engineering Adoption: A Comprehensive Example Using MBSE and Digital Twin with a Portable Robotic Arm" by Saulius Pavalkis and Mariah Otte distills a years-long applied research effort at Dassault Systèmes into a portable showcase: a five-axis Arduino-based robotic arm modelled, simulated, manufactured, and verified through a single digital thread. Using SysML-based MBSE, Modelica multiphysics, robotic simulation, FMI co-simulation, and MQTT-enabled hardware-in-the-loop, demonstrates how requirements trace directly to architecture, mechanics, electronics, and code. The result is a replicable template for universities and industry teams seeking to adopt DE with minimal cost and maximum pedagogical impact.

"Helping Organizations Adopt Digital Engineering in a Mature and Sustainable Way with DE CMAF" by Michael Shearin, Valerie Sitterle, and Owen Eslinger introduces a DE capability maturity and assessment framework (CMAF) to help organizations identify desired capabilities, required maturity levels on a specified timeline, and any gaps across all five DE capability areas: DE environments and infrastructure, workforce development and skills, workflows, DE practice, and time.

We hope you find *INSIGHT*, the practitioners' magazine for systems engineers, informative and relevant. Feedback from readers is critical to *INSIGHT*'s quality. We encourage letters to the editor at insight@incose.net. Please include "letter to the editor" in the subject line. *INSIGHT* also continues to solicit special features, standalone articles, book reviews, and op-eds. For information about *INSIGHT*, including upcoming issues, see *https://www.incose.org/publications/insight.* For information about sponsoring *INSIGHT*, please contact the INCOSE marketing and communications director at marcom@incose.net. ■


Accelerate Smart Product Innovation with MBSE for Electronics
DASSAULT SYSTEMES

The Digital Transformation of Standards: Why OEMs Must Act Now

Leslie McKay, leslie.mckay@sae.org

INTRODUCTION

Standards have remained largely unchanged since the early industrial revolution. While our world has transformed digitally, standards remain stuck in the 20th century published as static PDF documents that are increasingly inadequate for modern digital workflows. They're still "documents written by experts, for experts," human-readable text that requires manual discovery, interpretation, transformation, and implementation. Meanwhile, the world around us has gone digital:

- Financial transactions happen through seamless digital data exchange
- Software updates occur automatically in the background
- Information is consumed in bite-sized, interconnected formats

Yet standards remain as lengthy, dense documents that place the entire burden of extraction and implementation on end users. This creates a growing gap between how we consume information today and how standards are delivered.

THE DIGITAL IMPERATIVE

Original equipment manufacturers (OEMs) have made it clear: standards must become machine-readable and machine-interpretable, delivered as data that can integrate seamlessly into design and product lifecycle management (PLM) systems. Current PDF formats simply cannot support the data needs of modern digital workflows.

Research by SAE International reveals the transformative benefits that digital standards would deliver:

Discoverability – Properly structured and classified standards become searchable and identifiable, making navigation far easier.

Traceability – Connecting implementation and compliance with standards throughout the product development life cycle improves quality and performance.

Efficiency – Instead of each team manually extracting data from PDFs, shared repositories of interoperable standards information bring consistency across organizations.

Reusability – Digital repositories create powerful, up-to-date libraries of objects for organizations to build upon.

Accessibility – Moving from document-centric to object-based models allows required components to be accessed and connected more openly.

Quality – Digital approaches help identify gaps, errors, and duplication within standards while improving end products through easier implementation.

WHY CHANGE IS HARD

The standards industry faces unique challenges that make digital transformation difficult:

- **Fragmentation**: The ecosystem includes many different types of organizations with varying scopes, approaches, and resources.
- **Complexity**: Standards range from simple specifications to 100+ page documents with interconnected families of drafts, current versions, and amendments.
- **Diversity**: Standards support every industry and every need, making universal solutions challenging.
- **Dependency**: Standards development organizations (SDOs) rely on voluntary expert committees for content creation and validation.

These factors explain why individual SDOs struggle to make progress alone.

THE URGENCY FOR CHANGE

Without digital transformation, the standards industry risks obsolescence. Current document-based formats are becoming

Table 1. What makes the DSA unique

Industry-Driven	Single Purpose	Trusted	Globally Focused
Created based on user specifications and demands	Digital standards transformation is its sole mission	Built on SAE ITC's proven model of pre-competitive consortia	Geographically and organizationally agnostic

less fit-for-purpose as industry demands machine-readable, interoperable data that integrates seamlessly into modern digital workflows.

The benefits of digital standards—improved discoverability, traceability, efficiency, reusability, accessibility, and quality—are too significant to delay. The aerospace industry has taken the lead in forming the **Digital Standards Alliance (DSA)** because the need for change is urgent and the benefits are clear.

The DSA represents a new approach to solving this challenge. Founded by Boeing, Lockheed Martin, the Aerospace Industries Association (AIA), and SAE International, the DSA is an industry-led membership group focused exclusively on accelerating the migration to digital standards. Recent members include DIN Solutions, ASTM International, AFNOR Group, BSI Standards Ltd, UL Standards, and the Department of Defense (DoD) Standardization Program Office (See Table 1).

TAKING ACTION: STEPS FOR ORGANIZATIONS

For Companies Using Standards:

1. **Map your standards usage** – Identify who uses standards in your organization and how different roles interact with different types of standards.
2. **Assess current processes** – Understand which standards are most critical and how much time is spent extracting and converting standards information.
3. **Identify pain points** – Determine what causes the most frustration and inefficiency in standards use.
4. **Consider your digital future** – Understand what systems you'll use in 5-10 years and how standards need to interoperate with them.
5. **Define your vision** – Capture how you'd want standards delivered to be more useful and user-friendly.

For Standards Development Organizations:

1. **Know your portfolio** – Understand which standards in your collection are most suited to digitalization and transformation.
2. **Stay informed** – Keep aware of global progress, including ISO/IEC SMART programs and CEN CENELEC initiatives.
3. **Listen to users** – Understand the biggest frustrations with discovering, interpreting, transforming, and implementing your standards.
4. **Prioritize impact** – Focus on solving problems that would make the biggest difference for your customer base.
5. **Build on existing progress** – Identify what components of your digital ecosystem could contribute to unified industry solutions.

JOIN US!

Standards have been the invisible infrastructure holding modern systems together for over a century. Now, to remain relevant for the next century, they must transform to meet the digital age. The question isn't whether this transformation will happen, but how quickly the industry can make it reality. If you are in the aerospace and defense sector and wish to accelerate progress, it is critical that you join the DSA. ■

For more information about the Digital Standards Alliance, visit https://www.sae-itc.com/programs/dsa *or contact* DigitalStandardsAlliance@sae-itc.org.

ABOUT THE AUTHOR

Leslie McKay is the director of digital standards development at SAE International and the vice-chair of the Digital Standards Alliance. Leslie has over 20 years experience managing software and hardware products. Much of her career has focused on developing successful products that leverage artificial intelligence and machine learning, specifically voice recognition and natural language processing solutions. She won two user experience awards from Honeywell for leading a team to develop a voice-recognition system used to direct warehouse workers in distribution centers. Leslie is dedicated to establishing best practices for the authoring and use of digital standards in the product development lifecycle.

From Fragmentation to Federation: A Proposed Strategy for Advancing Digital Engineering Standards Development

Celia S Tseng, celia.tseng@incose.net; **Joseph W. Marvin,** joe.marvin@incose.net; **William D. Schindel,** schindel@ictt.com; **Juan Carlos Mendo,** juan.mendo@incose.net; and **Terri W. Chan,** terri.chan@incose.net

ABSTRACT

The future of systems engineering depends on solving a pressing challenge: the fragmentation of data and models across engineering tools, enterprise systems, and lifecycle processes. While existing standards from the International Organization for Standardization (ISO), the Object Management Group (OMG), and other standard development organizations (SDOs) address specific domains and data exchange formats, no comprehensive framework exists to guide digital engineering (DE) across the full ISO/IEC/IEEE 15288 system lifecycle process—from technical and management processes to enabling and agreement activities. This disconnect limits organizations' ability to integrate, scale, and deliver consistent value across increasingly complex ecosystems.

This article examines the current gaps in interoperability standardization efforts and outlines opportunities for the International Council on Systems Engineering (INCOSE) to shape the future of digital transformation. It introduces the INCOSE agile systems engineering life cycle management (ASELCM) pattern as a foundation for aligning lifecycle processes with model-based technical exchanges through a federated digital thread. The authors call for INCOSE to adopt a strategic standardization role—working with other SDOs to define interoperability use cases, align lifecycle models, and model-based systems engineering (MBSE) standards, and develop a shared reference framework for interoperability. This collaboration is essential for enabling seamless, cross-enterprise interoperability—and for empowering organizations to realize the speed, scale, and economic advantage promised by digital transformation.

KEYWORDS: digital engineering, interoperability, ontology, semantics, standards

INTRODUCTION

Achieving effective interoperability across the systems lifecycle remains a major challenge identified in the *Systems Engineering Vision 2035* (INCOSE 2021). Modern engineering spans multiple organizations, tools, and disciplines—each within its own evolving digital ecosystem—making the federation of data, processes, and tools essential for managing distributed, heterogeneous systems-of-systems. Despite numerous data standards such as SysML v2, FMI 3.0, STEP, and QIF, meaningful interaction across lifecycle processes remains fragmented (Powell 2025, Fischer et al. 2025).

Figure 1, excerpted from the *Systems Engineering Vision 2035*, highlights today's DE pain points. While the DE toolsets continues to grow, it is constrained by IT limitations and deeper system realities. Engineering tools—whether cloud or on-premise—typically access only the specific data they consume or produce, often in proprietary formats unique to each vendor. These tools perform critical functions but were not designed for the broad, interop-

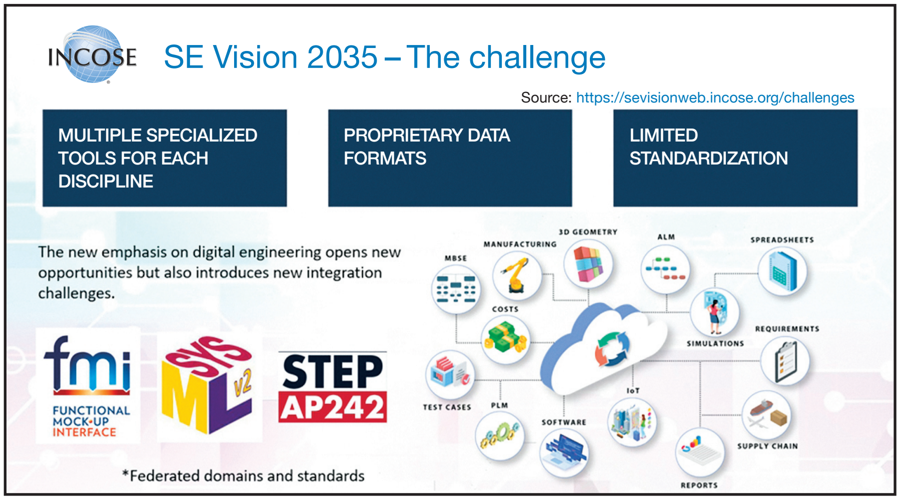


Figure 1. Systems Engineering Vision 2035 challenge

erable data access required for true DE. Standards to support this level of interoperability are only now beginning to emerge.

To achieve true interoperability, both syntactic and semantic interoperability must be addressed. Table 1 provides an overview of commonly used system engineering standards and their support for syntactic and semantic interoperability. Syntactic interoperability ensures that data formats, exchange protocols, and system interfaces are compatible, while semantic interoperability enables systems, machines, and stakeholders to interpret data meaningfully and consistently. This becomes especially important in model-based engineering, digital threads, and artificial intelligence (AI)-supported decision environments.

Each standard is assessed as having strong, partial, or indirect support in these two dimensions. A rating of strong indicates clearly defined data structures or formal logic-based semantics, enabling automated processing and integration. Partial support means the standard offers some structure or semantic clarity but lacks full formalization or consistency in application. Indirect suggests that while

Table 1. Systems engineering standards and interoperability support

Standard	Primary Domain	Syntactic Interoperability	Semantic Interoperability
ISO 10303 (Standard for Electronic Exchange of Product Data–STEP)	Product lifecycle, CAD/PLM	Strong	Partial
ISO 15926 (Industrial Automation Systems and Integration)	Process plants, lifecycle integration	Strong	Strong
ISO 15288/INCOSE Systems Engineering Handbook	Systems engineering lifecycle	Indirect	Indirect
OMG SysML v1.x	System modeling	Partial	Partial
OMG SysML v2.0	System modeling	Strong	Strong
ISO 42010	Architecture description	Indirect	Indirect
ArchiMate (TOGAF)	Enterprise architecture	Strong	Partial
DoDAF/MODAF/NAF/UAF	Enterprise architecture	Strong	Partial
OSLC (Open Services for Lifecycle Collaboration)	Cross-tool model traceability	Strong	Strong
W3C Semantic Web (OWL, RDF, SPARQL)	Cross-domain knowledge integration	Weak	Strong
ISO/IEC 11179	Metadata registries and vocabularies	Partial	Partial
LOTAR (EN/NAS 9300)	Product data archiving	Strong	Partial
ISO 14721 (OAIS)	General Digital Archiving	Strong	Partial
ProStep Digital Data Package (DDP)	Product data exchange	Strong	Partial
CASCaDE	Cross-disciplinary model-based collaboration	Strong	Strong
FMI	Simulation model exchange	Strong	Partial
QIF	Quality data exchange across manufacturing lifecycle	Strong	Partial

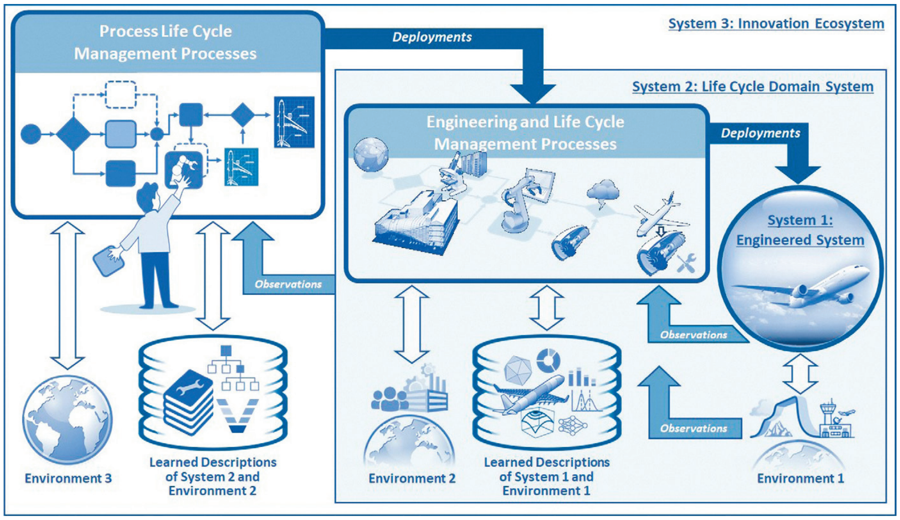


Figure 2. INCOSE ASELCM level 1 reference model

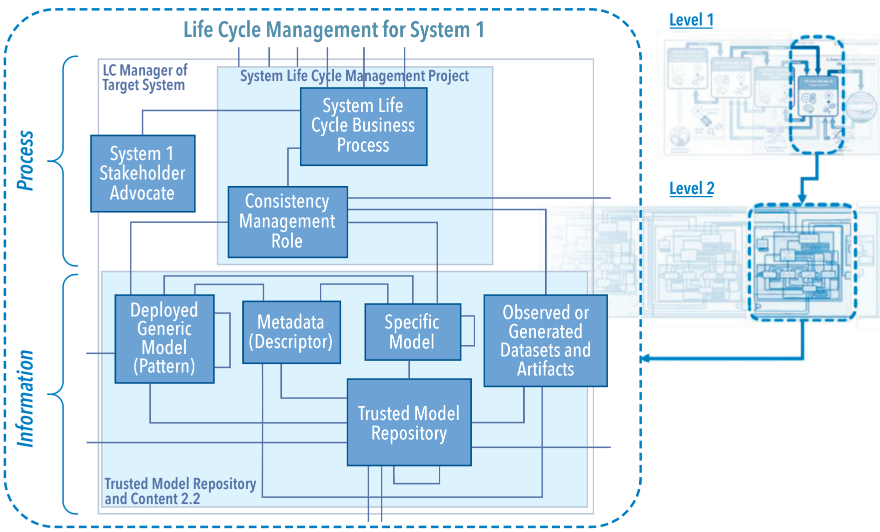


Figure 3. INCOSE ASELCM level 2 reference model

the standard may not directly address syntax or semantics, it can contribute when combined with other frameworks.

These assessments reflect both the intended purpose of each standard and how it is applied in practice, highlighting current strengths and gaps in achieving full lifecycle interoperability for model-based and digital engineering. While many standards support data exchange at the technical level, they often fall short in enabling semantically consistent communication across systems and stakeholders. The core issue is semantic heterogeneity—the lack of shared meaning across tools, models, and organizational contexts. As a result, information may be exchanged syntactically but misinterpreted semantically. Standards like ISO 15288 and the INCOSE *Systems Engineering Handbook* define what information is produced and consumed at each lifecycle stage, but they do not offer a theoretical foundation to ensure that these exchanges lead to aligned, collaborative outcomes.

The ASELCM Pattern (Figure 2) — particularly level 2 shown in Figure 3 — provides a system-theoretic framework for identifying where interoperability is essential across technical, technical management, agreement, and enabling processes (AIAA 2023). This framework helps analyze real-world use cases in which effective collaboration depends on aligned semantics and shared models. Interoperability is a prerequisite for consistency management—the ability to detect, assess, and reconcile discrepancies between models across disciplines (Schindel 2021). For instance, interoperability enables a change in a system requirement to automatically flag consistency issues in linked architecture models, test plans, or configuration data, helping teams maintain alignment across stakeholders and tools. Without this, organizations face late discovery of issues, miscommunication, redundant work, costly rework, and regulatory noncompliance—challenges that cannot be solved with isolated internal data alone. By establishing robust interoperability, organizations can maintain a traceable, version-controlled digital thread that supports faster decisions, stronger collaboration, and more predictable system outcomes (AIAA 2023).

HOW SPECIFIC MUST AN INTEROPERABILITY FRAMEWORK BE?

Interoperability frameworks must strike a balance between generality and domain specificity. In ontology hierarchies, general-to-specific layering is well understood, but expressing multiple models—such as subsystem specifications, industrial or internal reference architectures, or system-level designs—within the same framework does not ensure we can evaluate whether they (1) contradict each other or (2) align with known facts about their domain. The key question is: Is the framework specific enough to support the kind of consistency checking the stakeholder requires?

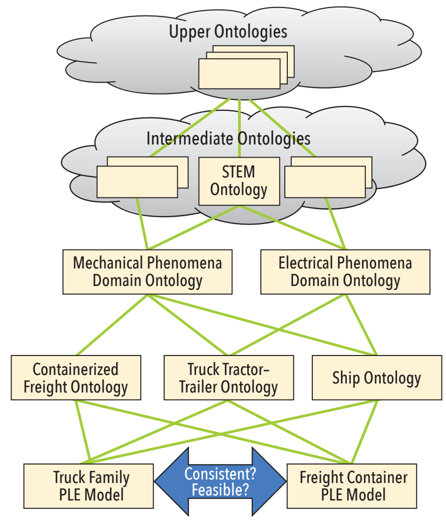


Figure 4: How specific must an ontology be?

For example, consider MBSE models of a semi-trailer truck family that must now accommodate containerized freight lifted by cranes for intermodal shipping (Figure 4). Even if both truck and container models are valid in SysML, their interoperability alone doesn't reveal whether their assumptions about crane, trailer, and ship interactions are logically or physically consistent. This isn't a failure of SysML—it's a matter of whether the reference ontology is specific and expressive enough to capture the semantics of those interactions.

Figure 4 also underscores a deeper issue: engineering ontologies should be grounded in the physical sciences. Generalized STEM principles, such as Hamilton's Principle, underlie all technical domains (Schindel 2024). Omitting these foundations from digital engineering ontologies risks weakening their ability to support real-world reasoning and innovation.

EXAMPLE STANDARDIZATION APPROACHES TO ADDRESS INTEROPERABILITY CHALLENGES:

Addressing interoperability in digital engineering requires strategies that handle both the structure and the meaning of exchanged information. Traditional standardization has emphasized shared syntactic structures and fixed terminologies—effective in static, bounded domains but insufficient for today's distributed, evolving, and heterogeneous ecosystems. As enterprise-scale systems grow more dynamic and cross-disciplinary, more agile approaches are emerging to support interoperability standardizations at scale. Three such approaches are outlined below:

1. Shared Terminology Approach (Common Meta-Standards)

   The most common interoperability strategy is to define a shared terminology in a standard—such as STEP, SysML, or QIF—to which all participating systems must conform. When applications use different internal vocabularies, human specialists create transformation mappings to align them (Bittner 2006).

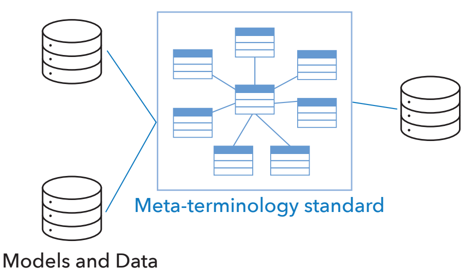


Figure 5. Shared terminology standards approach

   This approach has advanced syntactic interoperability but relies on static semantics that are rigid and brittle amid domain diversity and evolving technologies (Bittner et al. 2006). Because semantics are often implicit or narrowly defined, mismatches must be resolved manually, limiting scalability and hindering dynamic cross-domain collaboration or rapid tool integration.

2. Logic-Based Ontology Approach (Semantic Interlingua with Refence Ontology)

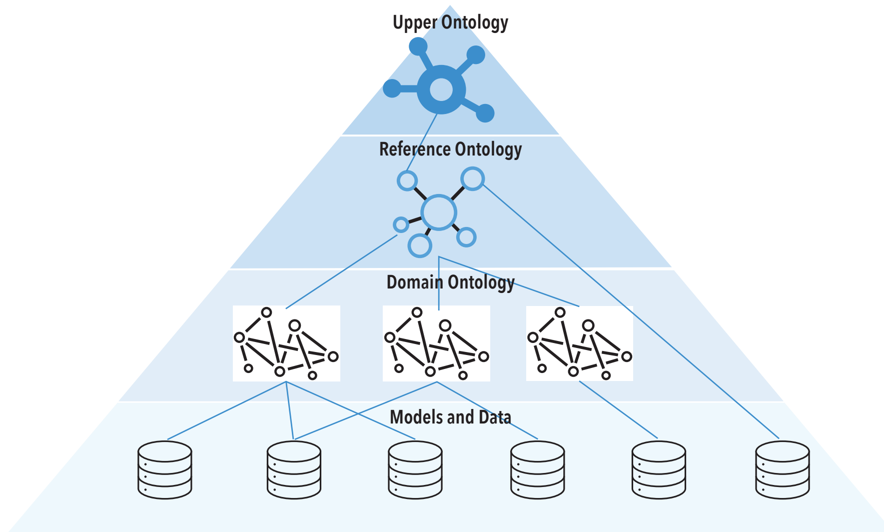


Figure 6. Logic-based ontology standards approach

A more agile and scalable strategy uses logic-based ontologies to formally define the semantics of each application's terminology. Systems retain their internal ontologies but map them to a shared reference ontology—an interlingua defined in formal logic (Bittner et al. 2006). This enables automated transformation via inference engines, preserving meaning across tools with different coordinate systems, measurement conventions, or modeling assumptions. Standardized upper-level ontologies such as the basic formal ontology (BFO) ISO/IEC 21838-2 (ISO 2021) and common core ontologies (CCO) (Rudnicki 2019) are available to serve as these references for aligning disparate domain ontologies in systems engineering—though they currently lack a STEM-focused ontology foundational to all engineering disciplines. This approach is applied by the INCOSE Digital Information Exchange Working Group (DEIX WG) to develop a digital engineering ontology (Gregory et al. 2025). In the manufacturing domain, the Industrial Ontologies Foundry (IOF) core ontology—grounded in BFO—enables interoperability across manufacturing, supply chain, and MBSE collaborations (Drobnjakovic et al. 2023).

3. Adaptive Multi-Semantics Approach to Consistency Management

   In large programs, artifacts and data sources frequently employ different semantics, and true uniformity may be impractical. Ongoing system evolution and new phenomena drive paradigm shifts that continually alter semantic frameworks (Schindel 2020). Rather than enforcing a single standard, this approach tolerates multiple coexisting semantics, using explicit consistency

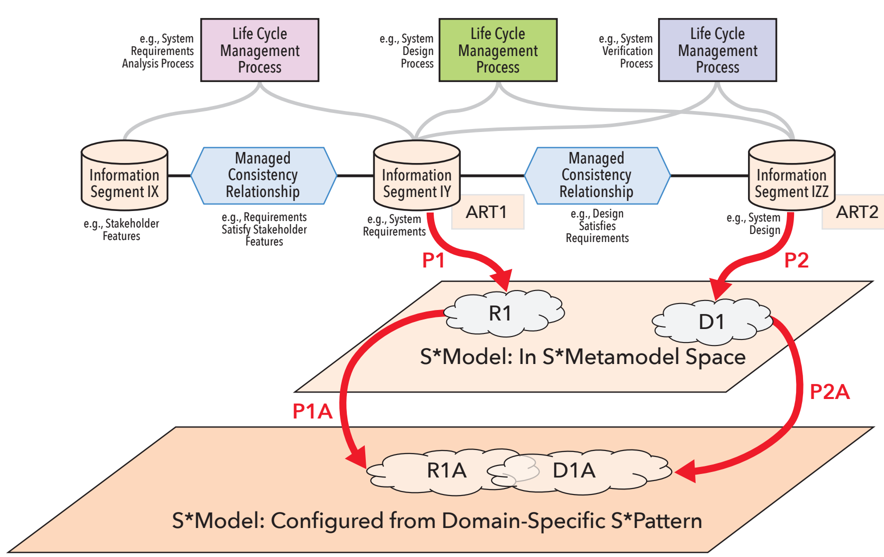


Figure 7. Understanding automated conversion as projections in vector sub-spaces

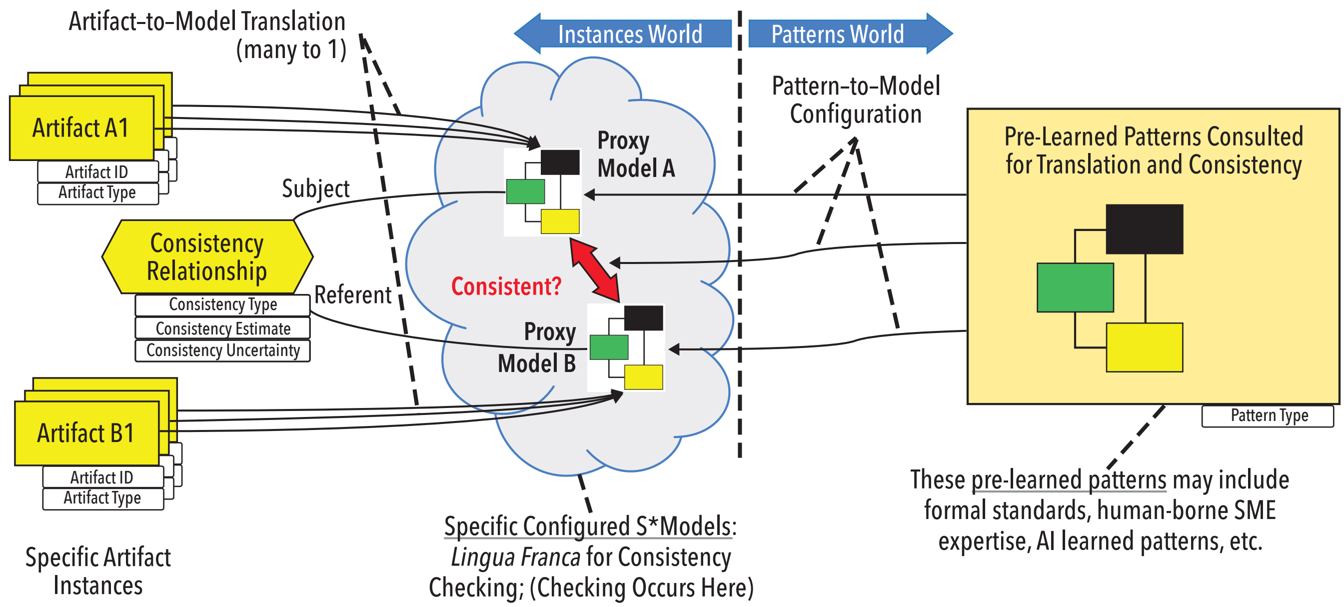


Figure 8. Checking consistency of artifacts by translating them to proxy models that are consistency checked

management to align them (Schindel 2021). The INCOSE MBSE Patterns Working Group is developing methods for machine-assisted translation of artifacts into standardized MBSE "proxy models" (Schindel 2025), based on the S*Metamodel as a STEM-based ontology (Schindel 2016 Patterns WG 2025). These proxies represent the original artifacts for cross-framework consistency checks, enabling dynamic interoperability across heterogeneous ecosystems (Figures 7 and 8).

Each of the three approaches—shared terminology, logic-based ontology, and adaptive multi-semantics—offers unique strengths depending on the use case. Shared terminology provides broad syntactic interoperability but lacks flexibility in rapidly evolving domains. Logic-based ontologies enable precise, automated semantic interoperability but require investment in semantic modeling. Adaptive multi-semantics accepts real-world variability, enabling pragmatic consistency management across diverse frameworks. No single approach is universally best; effective strategies often blend these methods to suit specific technical, stakeholder, and lifecycle needs. INCOSE and partner SDOs have an opportunity to combine these approaches to bridge technical standards and lifecycle models, advancing toward a federated, semantically robust ecosystem aligned with the *Systems Engineering Vision 2035*.

PROPOSED INTEGRATION STACK FOR INTEROPERABILITY STANDARDS

To support both syntactic and semantic integration, we propose a layered approach: an integration stack for interoperability standards (see Table 2 and Figure 9). This framework defines six layers: (1) infrastructure, (2) syntactic exchange, (3) semantic annotation, (4) domain ontology, (5) reference ontology, and (6) upper ontology. Each layer serves a specific role in enabling meaningful, structured, and scalable information exchange across systems and organizations. For systems

Table 2. Integration stack for interoperability standards

Layer	Purpose	Standard Categories	Example Standards
6. Upper Ontology Layer	Foundation for shared logic and cross-domain alignment	Upper-level, formal logic ontologies	BFO, DOLCE, SUMO, ISO/IEC 21838
5. Reference Ontology Layer	Provides shared interlingua for mapping domain models	Mid-level ontologies derived from upper ontologies	Common Core Ontologies (CCO), Industrial Ontologies Foundry (IOF) Core
4. Domain Ontology Layer	Encodes domain-specific semantics	Formal or semi-formal ontologies for engineering, manufacturing, etc.	CASCaDE Ontology, DEIX Ontology
3. Semantic Annotation Layer	Provides contextual meaning for data/model elements	Standards for annotation, semantic linking	RDF, OWL, SKOS, OSLC Resource Shapes, DEIX DVM concepts
2. Syntactic Exchange Layer	Structures how data is exchanged across tools and services	File formats, data schemas, exchange protocols	XML, JSON-LD, STEP (AP242/239), ReqIF, XMI, FMI, QIF, DEIX DVM
1. Infrastructure Layer	Transport and messaging	Network, transport, and API protocols	REST, HTTP, MQTT, OSLC Services, GraphQL

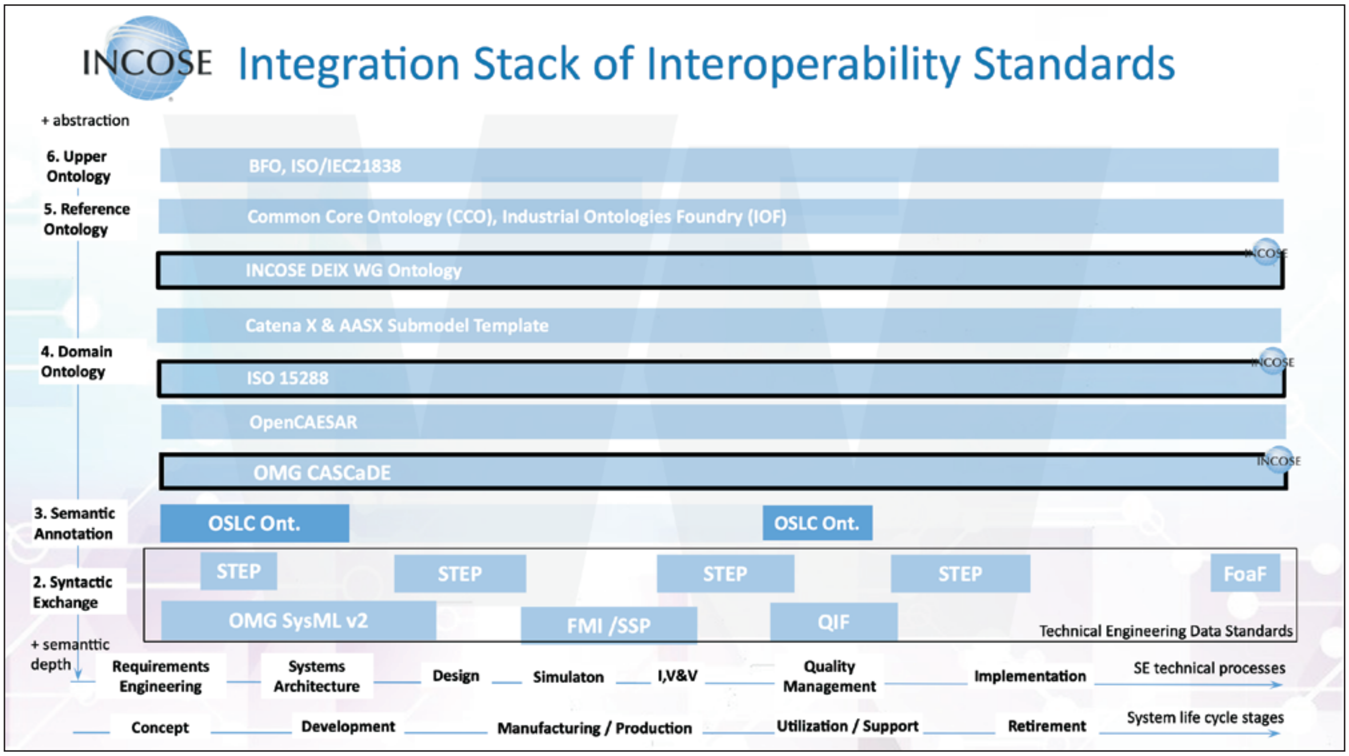


Figure 9. Integration stack of interoperability standards across system technical and management processes

engineers, it preserves data structure and meaning—supporting consistency, analysis, and decision-making. For tool providers, it offers a modular integration path that supports innovation within individual layers while aligning to a broader digital engineering ecosystem. This integration stack approach creates a flexible, scalable foundation for connecting diverse tools and standards across technical and management processes.

Figure 9 illustrates the layered integration stack for interoperability standards against the systems engineering technical and management processes, reflecting syntactic and domain-specific standards across all lifecycle stages. The standards shown in Figure 9 represent commonly used and developing systems engineering standards today. In the future, this mapping will continue to evolve and refined in alignment with INCOSE's standard development strategy to address emerging needs in model-based and digital engineering.

Figure 10. INCOSE strategic objective: advance system engineering as the world's trusted authority

PROPOSED INCOSE STANDARD DEVELOPMENT METHODOLOGY WITH SDOS

INCOSE is positioned to play a strategic role in shaping the future of systems engineering standards—an emphasis reinforced in the INCOSE Strategic Plan (Figure 10) (INCOSE 2024). To address interoperability challenges, INCOSE can continue and expand partnerships with SDOs such as ISO, OMG, and IEEE. Acting as an "independent broker," INCOSE can help integrate and govern changes to standards developed by multiple SDOs (see Figure 11).

INCOSE has initiated a MBSE–DE Integration Forum to align and scale the work of individual INCOSE WGs by coordinating users, vendors, researchers, and multiple WGs to identify ISO 15288-linked use cases, apply the interoperability stack, and surface standards gaps. Current examples include the Tool Integration and Model Lifecycle Management (TIMLM) Working Group, which is developing use cases for MBSE architecture co-development between OEMs and suppliers using diverse tools. The MBSE Patterns Working Group is addressing consistency management across lifecycle processes, while the DEIX Working Group is advancing use cases for digital artifact exchange using viewpoint models and ontologies. INCOSE also contributes use cases to OMG's CASCaDE, connecting standards from ISO, OMG, and Modelica. Maintaining a curated

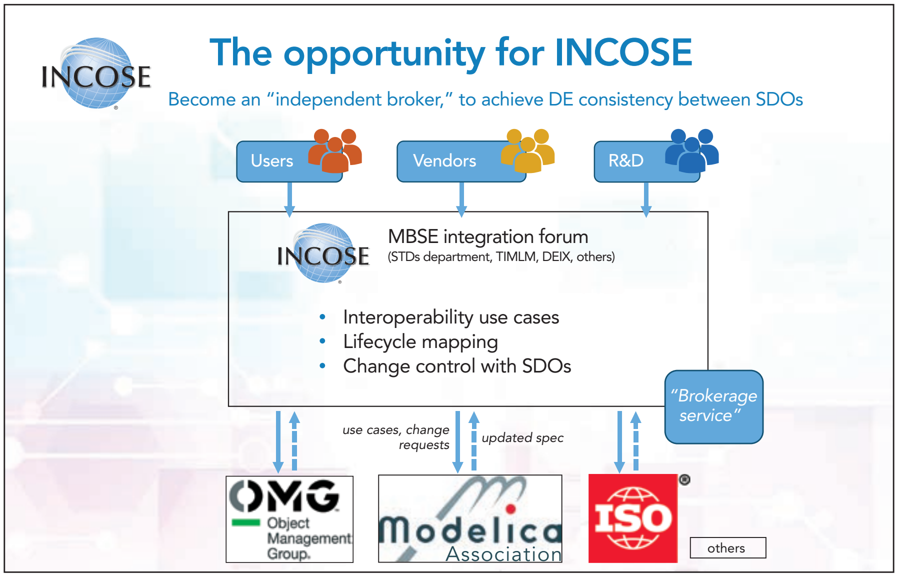


Figure 11. Proposed INCOSE standard development framework

catalog of interoperability use cases and lifecycle mappings would guide SDO collaboration, enabling iterative standards improvement and reinforcing INCOSE's leadership in building agile, standards-based DE ecosystems.

CONCLUSION AND PATH FORWARD

Interoperability is essential to the future of model-based, digitally connected systems engineering. While advances like SysML v2 represent major progress, significant gaps remain—particularly in achieving seamless, semantically aligned integration across tools, domains, and organizations. INCOSE can play a unique leadership role by identifying high-impact interoperability use cases and collaborating with SDOs to close these gaps.

We recommend applying a design science research (DSR) methodology, anchored in the ASELCM Pattern, to ensure emerging standards address real-world lifecycle needs. Consistency management should be the foundation for enabling digital thread continuity and cross-organizational collaboration. The Industrial Ontologies Foundry (IOF) in manufacturing illustrates how logic-based ontologies can enable scalable interoperability and digital twin innovation, but systems engineering still needs a STEM-informed ontology to fill critical foundations.

INCOSE has launched a standards task force to define its long-term standards strategy. Beginning with coordinated working group efforts at INCOSE IW 2026, this initiative will work toward an open, scalable, and semantically robust interoperability framework—advancing the vision set out in the *Systems Engineering Vision 2035*. ■

REFERENCES

- AIAA. 2023. Digital Thread: Definition, Value and Reference Model. American Institute of Aeronautics and Astronautics (AIAA) Digital Engineering Integration Committee (DEIC), June.
- Bittner, T. G., M. Donnelly, and S. Winter. 2006. "Ontology and Semantic Interoperability." DOI:10.1201/9781420036282.pt3.
- Drobnjakovic, M., et al. 2023. NIST TR 8472. "The Industrial Ontologies Foundry (IOF) Core Ontology: Architecture and Design." National Institute of Standards and Technology. https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=935068.
- Fischer, O., M. French, J. Hightower, A. Karl, J. Matlik, L. Pullum, W. Schindel, G. Shao, and N. Taylor. 2025. "Results of the 2024 Rosetta Stone Project—A Cross-Societies Collaboration." ASME VVUQ2025 Symposium, College Station, US-TX. https://www.omgwiki.org/MBSE/lib/exe/fetch.php?media=mbse:patterns:rosetta_stone_project_historical_context_discussion_for_asme_vvuq2025_v1.8.5.pdf.
- Rudnicki, R. 2019. "An Overview of the Common Core Ontologies." CUBRC, Inc, 12 February. https://www.nist.gov/system/files/documents/2021/10/14/nist-ai-rfi-cubrc_inc_004.pdf.
- Gregory, J., and A. Salado. 2024. "Towards a Systems Engineering Ontology Stack." Paper presented at the 34th Annual International Symposium of INCOSE.
- Gregory, J., J. Wheaton, C. Moreland, and C. Tseng. 2025. "Towards a Digital Engineering Ontology to Support Information Exchange." Paper presented at the 35th Annual International Symposium of INCOSE.
- INCOSE (International Council on Systems Engineering). 2021. *Systems Engineering Vision 2035: A World in Motion*. INCOSE. https://www.incose.org/products-and-publications/se-vision-2035.
- ———. 2024. INCOSE Strategic Plan. INCOSE. https://www.incose.org/docs/default-source/strategic-plan/incosestrategicplanpresentation.pdf?sfvrsn=4c4748c7_0.
- ISO (Internal Organization for Standardization). 2023. ISO/IEC/IEEE 15288:2023. International Standard - Systems and software engineering —System life cycle processes. Geneva, CH: ISO.
- ———. 2021. ISO 21838-2:2021 Information technology—Top-level ontologies (TLO) — Part 2: Basic Formal Ontology (BFO). Geneva, CH: ISO.
- Powell, S. 2025. "Digitalization of Standards: The Need to Act and Where to Start." Digital Standards Alliance, A Consortium of SAE and ITC. https://www.sae-itc.com/programs/dsa/whitepaper.
- Schindel, W. 2021. "Consistency Management as an Integrating Paradigm for Digital Life Cycle Management with Learning," INCOSE MBSE Patterns Working Group, download from— https://www.omgwiki.org/MBSE/lib/exe/fetch.php?media=mbse:patterns:aselcm_pattern_-_consistency_management_as_a_digital_life_cycle_management_paradigm_v1.3.1.pdf.
- ———. 2024. "Innovation Ecosystem Dynamics, Value and Learning I: What Can Hamilton Tell Us?" Paper presented at the 34th Annual International Symposium of INCOSE.
- ———. 2016. "Got Phenomena? Science-Based Disciplines for Emerging System Challenges." Paper presented at the 26th Annual International Symposium of INCOSE.
- ———. 2025. "Vector Spaces, Projections, and Patterns: Support for MBSE, Learning, and Trust." INCOSE MBSE Patterns Working Group.https://www.omgwiki.org/MBSE/lib/exe/fetch.php?media=mbse:patterns:expressing_patterns_in_vector_spaces_v1.2.4.pdf.
- ———. 2020. "Discussion Inputs to INCOSE Vision 2035 Theoretical Foundations Section." INCOSE Patterns Working Group. https://www.omgwiki.org/MBSE/lib/exe/fetch.php?media=mbse:patterns:science_math_foundations_for_systems_and_systems_engineering-1_hr_awareness_v2.3.2a.pdf.

- Patterns WG. 2025. "The Systematica Metamodel" Version 8.0. INCOSE Patterns Working Group, INCOSE. www.omgwiki.org/MBSE/lib/exe/fetch.php?media=mbse:patterns:systematica_8_metamodel_v8.0.61.pdf.
- Walden, D., et al, eds. 2023. *INCOSE Systems Engineering Handbook*, Fifth Edition, International Council on Systems Engineering, San Diego, CA.

ABOUT THE AUTHORS

Celia S. Tseng is the director of aerospace and defense consulting at Dassault Systems, with 21 years of systems engineering experience from Raytheon Technologies, Lockheed Martin, and SAIC. Celia is a co-chair of the joint INCOSE/NDIA/ OMG Digital Engineering Information Exchange Working group and INCOSE standard liaison in ISO/IEC JTC 1 SC 7 digital engineering. She has a master's in systems engineering from Cornell University, a certified systems engineering professional (CSEP), a certified system modeling professional (OMG OCSMP), and certified agile scrum master (SAFe).

Joseph W. Marvin, ESEP–Joe has served in various INCOSE roles since 2007 including Central Arizona Chapter president, chair of the Small Business Systems Engineering Working Group (formerly VSEWG), technical operations assistant director for industry, technical operations assistant director for internal operations, and assistant director for outreach. He currently serves as the assistant director of the standards development department and leads INCOSE's delegation in support of ISO/IEC standards development. Joe is the chief value officer at Prime Solutions Group, Inc. (PSG). PSG is an INCOSE CAB member and digital engineering defense industry supply chain contractor.

William D. Schindel, INCOSE fellow and CSEP. Bill chairs the INCOSE MBSE Patterns Working Group. He served as president of the Crossroads of America Chapter, where he is a founding director. He served as a member of the lead team for the INCOSE agile systems engineering life cycle model discovery project. An active member of ASME and AIAA, he served on the authoring team for the AIAA digital thread and digital twin reference models. He has been an active authoring member of the ASME VVUQ50 standards subcommittee on model credibility in advanced manufacturing, and most recently led the joint ASME-AIAA-INCOSE-NAFEMS Rosetta Stone project mapping alignment of the current standards of those SDOs for computational model VVUQ.

Juan Carlos Mendo is a technical lead at Boeing focusing on MBSE interoperability, collaboration, and data standards development. He is the systems engineering team leader for PDES. INC. He leads the LOTAR EN/NAS9300 standard for MBSE. He is the subject matter expert for STEP AP243, "MoSSEC." He chairs the CASCaDE OMG project. He currently supports INCOSE as TIMLM WG co-chair.

Terri W. Chan is an associate technical fellow at the Boeing Company. She is currently working on the 777-9 program electrical subsystems and developing implementation strategy for electrical and connected systems model based engineering across the airplane product lifecycle. She is the chair of the INCOSE/ NDIA/ OMG Digital Engineering Information Exchange Working Group and currently serves on the aerospace and mechanical engineering department advisory board at the University of Southern California.

Digital Engineering: Transforming The Research Into The Business Roadmap

Salvatore R Bruno, Salvatore.R.Bruno@lmco.com; **Dr. Carol Woody,** cwoody@cert.org; **Steve Henry,** tamu1971@msn.com; and **Celia Tseng,** Celia.TSENG@3ds.com

ABSTRACT
Organizations are embedding digital engineering and model-based engineering into their systems engineering lifecycles to achieve cost savings, higher product quality, and earlier delivery. From a business perspective, understanding the success of their digital transformation investment, has been lacking. To address this requested need, government, university, and industry experts have collaborated with the INCOSE Measurement Working Group to enhance the issuance of the initial Digital Engineering Measurement Framework (v1.1) to include measures that track, and report results related to digital engineering business operations and environmental benefits. This article describes the approach used to mature the guidance document, the benefits of the new measures, and recommendations for the sequential release.

INTRODUCTION

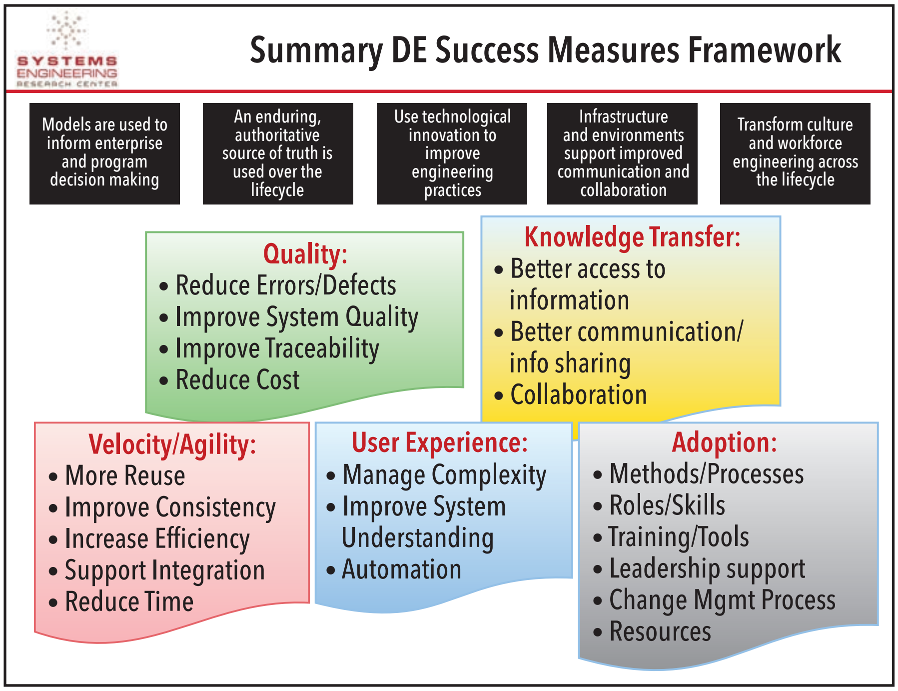
The "Digital Engineering Measures: Research and Guidance" document from the INSIGHT March 2022 issue introduced an initial framework for measuring the value of digital engineering (DE) and model-based systems engineering (MBSE). This framework has since matured to focus on enhancing the precision and application of these measures to facilitate program digital transformations and improve outcomes. Figure 1 on the following page illustrates the summary of the first article "Digital Engineering Measures: Research and Guidance."

The initial research by the Systems Engineering Research Center (SERC) in collaboration with a government/industry Digital Engineering Measures Working Group aimed to fill a value measurement gap for DE and MBSE. This endeavor was motivated by the growing complexity of systems, the increasing use of models in design, and the need for quantifiable evidence of the value of DE and MBSE, which represent a substantial financial investment.

Digital engineering (DE) and model-based systems engineering (MBSE) are approaches to improve the efficiency and productivity of engineering activities, particularly for complex engineered systems, primarily through the integration

Framework 1 identified eight direct benefits of DE/MBSE	
• Higher level support for automation	• Strengthened testing
• Early verification and validation (V&V)	• Better accessibility of information (ASoT)
• Reusability	• Higher level support for integration
• Increased traceability	• Multiple viewpoints of model

Framework 2 shows the benefiting principles used to produce eight digital engineering measures	
• Functional architecture completeness and volatility	• Adaptability and rework
• Model traceability	• Digital engineering automation
• Model product size	• Deployment lead time
• Digital engineering anomalies	• Runtime performance



Primary Benefits	Description	Applicable Measurement Specifications
Higher level support for automation	Use of tools and methods that automate previously manual tasks and decisions	8.6 Product Automation 8.7 Deployment Lead Time
Early Verification and Validation (V&V)	Moving tasks into earlier developmental phases that would have required effort in later phases	8.4 DE Anomalies 8.5 Adaptability and Rework 8.7 Deployment Lead Time
Reusability	Reusing existing data, models, and knowledge in new development	8.4 DE Anomalies 8.5 Adaptability and Rework 8.7 Deployment Lead Time
Increased Traceability	Formally linking requirements, design, test, etc. via models	8.7 Deployment Lead Time 8.8 Runtime Performance
Strengthened Testing	Using data and models to increase test coverage in any phase	8.1 Architecture Completeness and Volatility 8.2 Model Traceability 8.3 Product Size
Better Accessibility of Information (ASoT)	Leveraging an Authoritative Source of Truth (ASoT) to increase access to digital data and models to increase the involvement of stakeholders in program decisions	8.7 Deployment Lead Time 8.8 Runtime Performance
Higher Level of Support for Integration	Using data and models to support integration of information and to support system integration tasks	8.6 Product Automation 8.2 Model Traceability
Multiple Model Viewpoints	Presentation of data and models in the language and context of those that need access	8.1 Architecture Completeness and Volatility 8.7 Deployment Lead Time

Figure 1. DE 1.1 Summary success measures and primary benefits

of data and models, often referred to as the authoritative source of truth (ASOT). MBSE is described as the formalized application of modeling to the systems engineering process. The research indicated that DE/MBSE have measurable benefits, and their measures can be defined and tracked as extensions to well-known software measures, primarily supporting the systems engineering process. A causal model was developed to systematically prioritize metrics, linking direct benefits to secondary benefits (effects/results). The framework identified eight direct benefits of DE/MBSE.

The framework benefiting principles were used to produce eight digital engineering measures, and are utilized multiple times as shown in Figure 1 DE 1.1 Summary success measures and primary benefits.

The main objective of these eight measures were to reduced errors/defects, effort, and time, as well as improved functional completeness and correctness, efficiency in model-based review artifacts, and enhanced collaboration. It adopted the Practical Software and Systems Measurement (PSM) framework as a baseline for specifying measures which comprises operational, system, and discipline-specific models, a data and model ontology, process models, life-cycle models, and digital infrastructure. The initial DE paper supported the DOD Digital Transformation Strategy (Section 231 of Public Law 116-92 SEC. 231. DIGITAL ENGINEERING CAPABILITY TO AUTOMATE TESTING AND EVALUATION) as well as DoDI 5000.97 DIGITAL ENGINEERING.

Building on the original eight measures, the next phase of the digital engineering measurement framework refined precision indicators through operational application and formalized the translation of measurement research into practical business outcomes and expanded support to other elements of the reference documents DoDI 5000.97, "Digital Engineering" (December 21, 2023 10 USC 2223 Section 31). That is, a shift from merely defining what to measure, to refining how to measure effectively and use those measurements for business decision-making. The objective outcome was to empower business leaders to better understand and track the significant contributions and benefits of their digital transformation journey and future tangible expectations. The driving factors in meeting the DoDI 5000.97, Paragraph 3.2, Digital Engineering Capability consisted of:

- **Clarifying Return on Investment (ROI)**: Digital engineering (DE) and model-based engineering (MBE) involve significant investment. By

Business Objective Principals	DE Indicators
Cost	Digital engineering implementation cost
	Near-term ROI
Product quality	Model completeness
	Model requirements coverage
	Engineering product value
Development efficiency	Model progress
	Reused models and elements

Figure 2. Business objective principals to DE indicator mapping

providing more precise indicators, the framework helps companies **track the success of their digital transformation efforts and investments**, making the resulting benefits quantifiable and less uncertain. This moves beyond perceived benefits to demonstrate a clear return on investment (ROI).

- **Informing Decision-Making**: With more precise data, companies can make **more informed decisions** about their DE initiatives, understanding what is working and what needs adjustment. This supports the goal of enabling organizations to develop products sooner, at a lower cost, while maintaining or improving quality.
- **Driving Operational Efficiency**: Measures like **product automation deployment lead time, efficiency, effort, and cost** directly illustrate how digitalization reduces manual steps and improves process workflows. This allows programs to continuously identify bottlenecks and areas for optimization.
- **Enhancing Product Quality**: Indicators such as **model review item discrepancies, defect detection, and defect resolution (by phase)** provide concrete data on how DE/MBSE improves early defect detection, reduces rework, and ultimately leads to higher quality products being deployed.
- **Improving Traceability and Information Accessibility**: Measures like **model traceability** and **ASOT frequency of access** provide insights into how effectively information is linked and shared across disciplines. Increased traceability ensures that system definitions are complete and correct, reducing errors and supporting functional correctness. Greater accessibility via an authoritative source of truth (ASOT) is a leading indicator of collaboration.
- **Fostering Agility and Adaptability**: By tracking metrics related to **deployment lead time and efficiency**, programs can assess their ability to rapidly deliver capabilities and adapt to changes, which are core promises of digital transformation.
- **Supporting Continuous Improvement**: The comprehensive measurement framework enables a feedback loop where organizations can **understand where they have made progress and where they need to still make progress**. This iterative assessment facilitates the ongoing evolution and refinement of digital practices.
- **Bridging the Gap between Research and Practice**: The explicit goal of transforming research into a business roadmap signifies a move towards **practical, actionable guidance** that directly supports companies in leveraging DE for tangible business outcomes, addressing the "missing elements in demand" for benefits tracking.

By focusing on these precision indicators and grounding the framework in real-world experience, the matured guidance aims to provide companies with the concrete tools necessary to not only implement digital engineering but also **quantifiably demonstrate its success and value** across their operations, ultimately leading to improved **product quality, development efficiency**, and **cost savings**.

The team further elaborated, through a double weighted evaluation and assessment approach, to produce seven additional measures supporting the three business objective principals of product quality, development efficiency, and cost savings within the digital engineering transformation development as shown in Figure 2.

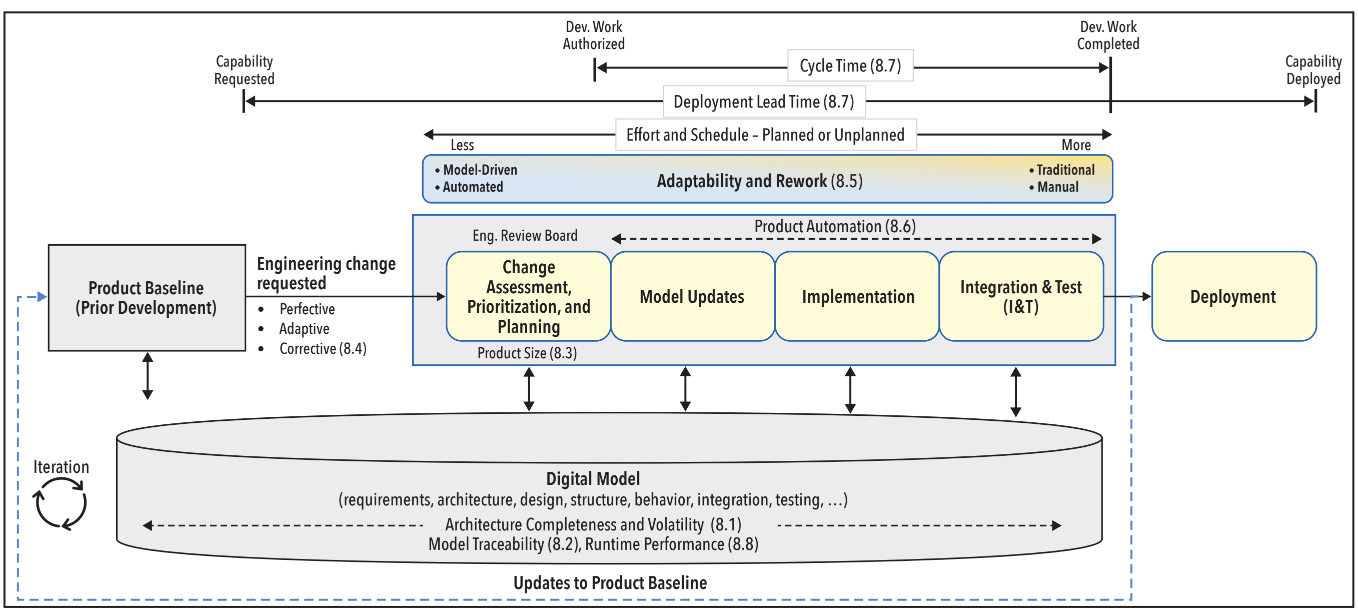


Figure 3. DE 1.1 Measurement framework architecture

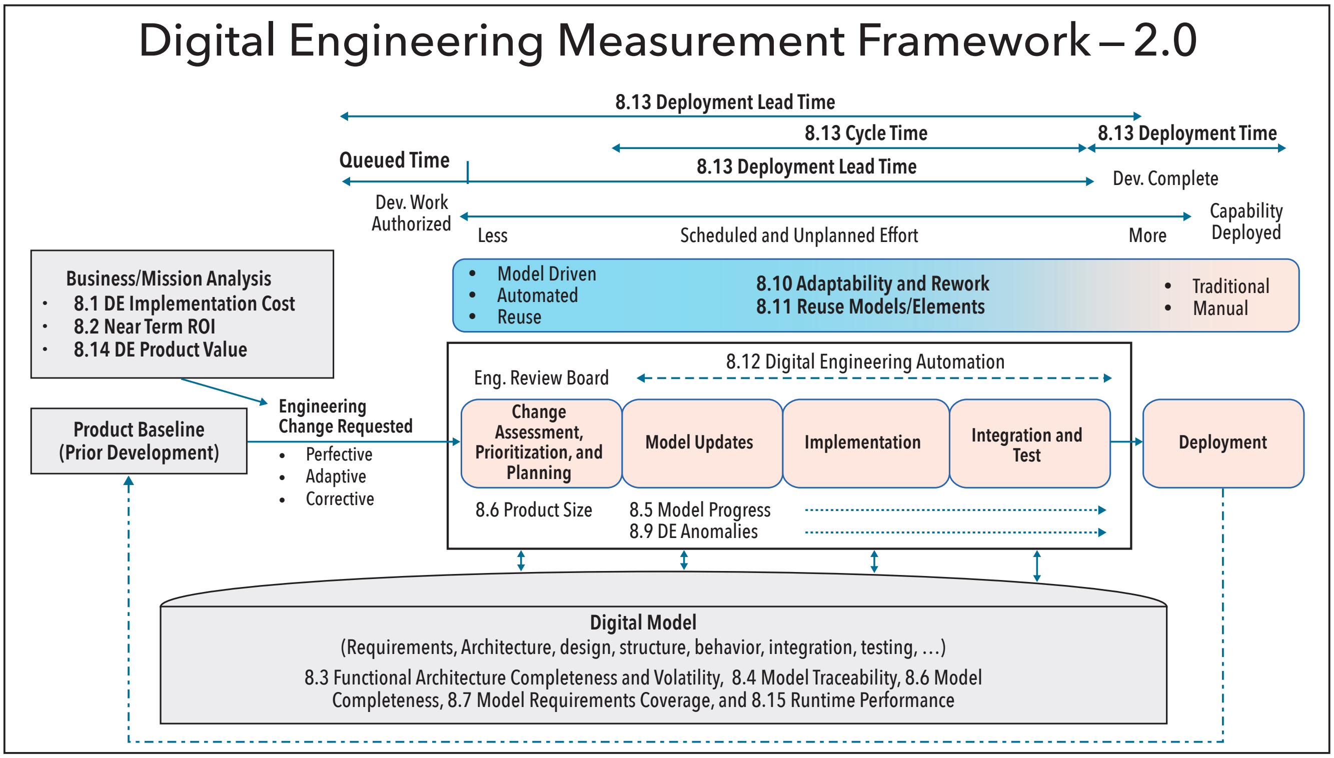


Figure 4. DE 2.0 Measurement framework architecture

The remaining two activities among the team's efforts were to clearly define and describe the new indicators and integrate seamlessly into the current digital engineering framework architecture. This was accomplished by dividing the members into seven parallel sub-teams to define indicators in detail — covering concepts, primitive and derived data elements, calculations, authoritative sources of truth (ASOT), and information needs — and then sharing results across sub-teams to integrate them into the digital engineering measurement framework architecture. A core team of editors performed the final integration, producing Figure 3 and Figure 4 that illustrate the changes before and after implementation respectively.

The following is a description summary of the seven newly added digital engineering indicators.

DIGITAL ENGINEERING IMPLEMENTATION COSTS

The US DoD is directing programs to conduct a comprehensive engineering program for defense systems, pursuant to DoD Instruction (DoDI) 5000.88 and DoDI 5000.97. In support of that effort, the DoD directs use of digital engineering methodologies, technologies, and practices across the life cycle of defense acquisition programs, systems, and systems of systems to support research, engineering, and management activities. DoD requires that digital engineering be addressed in the acquisition strategy, including how and when digital engineering will be used in the system life cycle and expected benefits of its use. Digital engineering (DE) leverages advanced digital tools to model and simulate lifecycle system development and sustainment. DE planning and execution must address the costs for the digital engineering ecosystems, digital models including digital twins, digital threads, and digital artifacts. These DE implementation efforts require lifecycle planning and support.

NEAR-TERM ROI

Digital engineering (DE) utilizes advanced digital tools, models, simulations, and workflow automation to enhance engineering processes enabling the transition from traditional, paper-based methodologies to digital-centric systems. This transformation offers numerous advantages such as accelerated development cycles, reduced costs, enhanced collaboration, and decreased risks. However, this shift also involves significant investments in software, technology, hardware, training, infrastructure, and integration.

The return on investment (ROI) for DE, defined as the ratio of benefits to costs, is crucial for justifying DE implementations within commercial entities, the Department of Defense (DoD), and DoD contractors.

MODEL PROGRESS

The model progress indicator is used to track and measure the development and growth of the one or more digital transformation engineering activities (integrating digital technologies like computer-aided design (CAD), simulation software, and advanced analytics into traditional engineering practices to streamline processes, automate repetitive tasks, improve collaboration, and ultimately enhance product development efficiency and innovation, by leveraging data analysis and predictive maintenance techniques). Each digital transformation engineering activity is tracked and measured against the established project timeline. The results of completed digital transformation engineering activities can be used to plan future digital engineering projects and provide insight into how to make upcoming digital engineering projects more efficient by being more affordable, and more likely completed on time or sooner with fewer delays, rework, and corrections.

MODEL COMPLETENESS

The model completeness indicator measures progress toward completion of models. These models can include descriptive, analytical, design, or other types. These models can include models at any level, including systems, systems of systems, enterprises, solution concepts, etc. Model completeness can be determined in both the problem space and the solution space as well as planned and actual.

MODEL REQUIREMENTS COVERAGE

The model coverage indicator assesses the extent to which the model element requirements of the digital engineering environment have been implemented, reflecting progress toward digital engineering transformation. The indicator provides the progress of the model coverage and provides a performance rating. The model coverage progress is for the current point in time and the performance rating is the expected model coverage index on the final delivery date based upon current model coverage productivity.

REUSED MODELS AND ELEMENTS

The concept of reuse readiness provides an assessment of the current maturity of the models and identifies the effort required to achieve the desired level of maturity. The model reuse readiness levels (RRLs) are focused on pinpointing the ability of model/model components and interfaces to be reused in each context and on assessing the potential reusability of software components, systems, and interfaces downstream. Programs can use the metric to assess the maturity and risk of model reuse by assessing the following elements:

- **Documentation**: Information that describes the software asset and how to use it.
- **Extensibility**: The ability of the model to be grown beyond its current context.
- **Intellectual Property**: The legal rights for obtaining, using, modifying, and distributing models and/or model components.
- **Modularity**: The degree of segregation and containment of a model/model components.
- **Packaging**: The methodology and technology for assembling and encapsulating the components of a software asset.
- **Portability**: The independence of an asset from platform-specific technologies.
- **Standards Conformance**: The adherence of a model/model component to accepted technology definitions.
- **Support**: The amount and type of assistance available to users of the model/model component.
- **Verification and Validation**: The degree to which the functionality and applicability of the model or model component has been demonstrated.

DIGITAL ENGINEERING PRODUCT VALUE

The digital engineering product value (DEPV) is a quantitative metric like a technical performance measure (TPM) used to assess the value, performance, or maturity of a digital engineering product or digital transformation throughout its lifecycle against planned requirements and goals. It helps stakeholders understand how well a digital engineering effort (like a model, simulation, or digital twin) contributes to overall project or organizational goals. It provides insight into the progress of transformation/development, the impact of changes, and the overall health of the transformation. This indicator also works in conjunction with the near-term ROI indicator list above.

In conclusion, the second release of the digital engineering measurement framework guidance document extended what was started to further assist companies and their systems engineers in gaining increased insight to the benefits of their digital enterprise system by tracking, and reporting results of their transformation efforts and resulting business operations. By adopting government studies and policies and levering subject matter experts from universities and corporations, the Digital Engineering Measurement Framework document has matured from eight to fifteen indicators that provide companies the necessary feedback to guide them on their transformation efforts. An important next step for the framework is to incorporate indicators that address cybersecurity considerations. Equally important is gathering user feedback to refine existing indicators and identify additional ones based on emerging needs from the user community. ■

ABOUT THE AUTHORS

Salvatore R Bruno is the engineering enterprise operations measurement and analysis principal engineer at Lockheed Martin Corporation. He oversees the common business, organizational, and program measurements and analysis standards and practices across the corporation's systems engineering operations. This includes development multiple discipline metrics that are highly automated, common framework, templates, dashboards, and likely root cause issues. Mr. Bruno continues to elevate the engineering measurement practices with new and innovative measurement and analysis approaches. He currently serves as the INCOSE Measurement Working Group chair and is an active member of several organization such as Practical Software and Systems Measurement, InterNational Committee for Information Technology Standards, BOEHM – Center for Systems and Software Engineering, and Six Sigma. Mr. Bruno holds a BS in mathematics and BS in computer science from California State University, East Bay and a MS in systems management from the University of Denver.

Dr. Carol Woody is a principal researcher for the CERT division of the Software Engineering Institute at Carnegie Mellon University. She is the technical manager of the CERT cybersecurity engineering team which is building capabilities and competencies for measuring, managing, and sustaining software assurance and cybersecurity for highly complex software intensive systems and supply chains throughout the acquisition lifecycle. Dr. Woody has successfully implemented solutions in many areas, including banking, mining, manufacturing, local and federal government, military, and finance. She coauthored a book *Cyber Security Engineering: A Practical Approach for Systems and Software Assurance* published by Pearson Education as part of the SEI Series in Software Engineering. The CERT cybersecurity engineering and software assurance professional certificate is based on the research she led. Dr. Woody holds a BS in mathematics from William and Mary, an MBA from Wake Forest University with distinction, and a PhD in information systems from NOVA Southeastern University.

Steve Henry is an acquisition consultant providing tailored training to enable programs to effectively plan and start programs right. He has more than 43 years of experience in designing, implementing, testing, and managing large programs. Mr. Henry served 28 years in the US Air Force retiring at the rank of Colonel in 2000. He is a graduate of the USAF Test Pilot School and flew the first flight of the B-1B in 1984. Mr. Henry is the recipient of the Air Force Material Command Chief Engineer of the Year, Air Force Association, Executive Division, Program Manager of the Year, and Armed Force Communications and Electronics Association's International Award for Excellence in Information Technology awards. At Northrop Grumman, he significantly improved the quality of systems engineering as director of engineering for the command-and-control division and sector systems engineering center of excellence lead. He is the past chairman of the NDIA systems engineering division and supports numerous NDIA working groups to support DoD. In 2015 he was named the winner of the National Defense Industrial Association's Lt Gen Thomas R. Ferguson, Jr. Systems Engineering Excellence Award.

Celia Tseng is the director of aerospace and defense consulting at Dassault Systems, with 21 years of systems engineering experience from Raytheon Technologies, Lockheed Martin, and SAIC. She is a co-chair of the joint INCOSE/ NDIA/ OMG Digital Engineering Information Exchange working group and INCOSE standard liaison in ISO/IEC JTC 1 SC 7 digital engineering. Ms. Tseng has a MS in systems engineering from Cornell University, a certified systems engineering professional (CSEP), a certified system modeling professional (OMG OCSMP), and certified agile scrum master (SAFe).

System Thinking in the Design of Enterprises with Support of SEREA (Systems Engineering Reference Enterprise Architecture)

Hugo Ormo, hugo.ormo@nttdata.com

Note to the Readers | Several of the SysML diagrams in the figures are too large to be shown in the "letter" format of *INSIGHT* and were clipped by the author. INCOSE intends to make the full diagrams accessible online.

ABSTRACT
One of the biggest challenges in designing complex systems is to design and specify the components in such a way that they contribute to the overall functionality of the system. Specialization, cost efficiency, or other reasons require that these components be co-developed, verified, and delivered by various suppliers, and then integrated, verified, and validated by the integrator. This makes the approaches of systems engineering crucial for integrators (OEMs). As the complexity of the systems to be managed increases, so does the complexity of designing, maintaining, and further developing these approaches. The methods of model-based systems engineering (MBSE) have already proven their relevance at the system level today. Companies cannot ensure the required quality of their systems without a model-based approach. The transition from document-based to model-based operational architectures presents a comparable challenge. SEREA is listed to be designated as an INCOSE technical product, offering a model-based reference enterprise architecture based on the uified architecture framework (UAF). With SEREA as a basis, a customized enterprise architecture can be created. Since SEREA is based on ISO/IEC/IEEE 15288, it is particularly suitable for companies dealing with the life cycle management of complex systems. This paper describes how SEREA has been developed and how shall be tailored.

1. INTRODUCTION

The specification of the unified architecture framework (UAF) is published by the Object Management Group (OMG) and is currently available in version 1.2 [1]. UAF [1] specifies a modeling language named UAFML, which was created as a profile of SysML 1.7 [2]. Currently, a version 2 of UAF and UAFML is being specified, which will build on SysML v2 [3].

UAF aims to unify existing military frameworks such as DoDAF, MoDAF, and NAF. Although the purpose of UAF is not only to serve as a unified framework for a range of military frameworks but to be suitable for all industries, its introduction faces limitations [4, 57-58].

Nevertheless, the introduction of UAF and UAFML has the potential to benefit any organization, even those not operating in the military context [4, 52-56]. UAF focuses on operational and mission architecture and allows organizations to define stakeholder requirements for their supporting systems precisely and with thorough justification. This is achieved not only initially but also continuously, as the context of the organization changes over time, leading to new drivers that continuously present new challenges and opportunities impacting the capabilities the organization needs to achieve desired effects [5]. These new or altered capabilities are mapped to operational activities executed in a newly defined operational architecture. Ultimately, new personnel, service, and resource architectures will implement the operational agents. These new implementation architectures consist of human resources

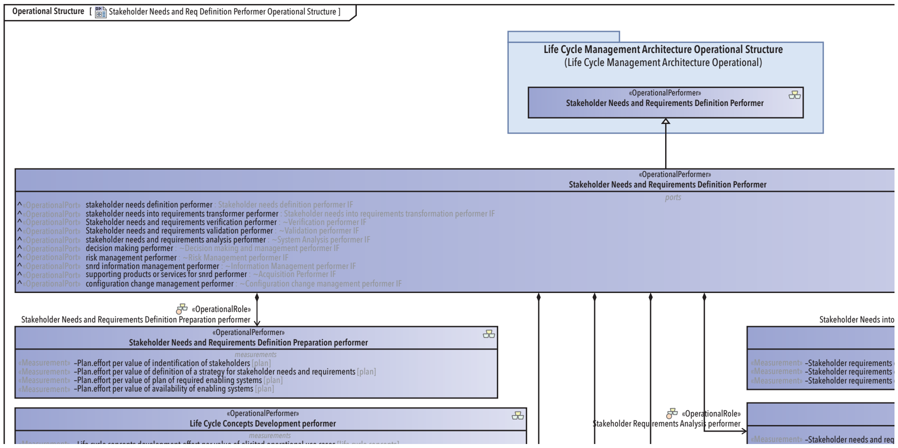


Figure 1. Definition of stakeholder needs and requirements (SNRD) performer and its component performers with defined measures of performance (MoP) for each performer

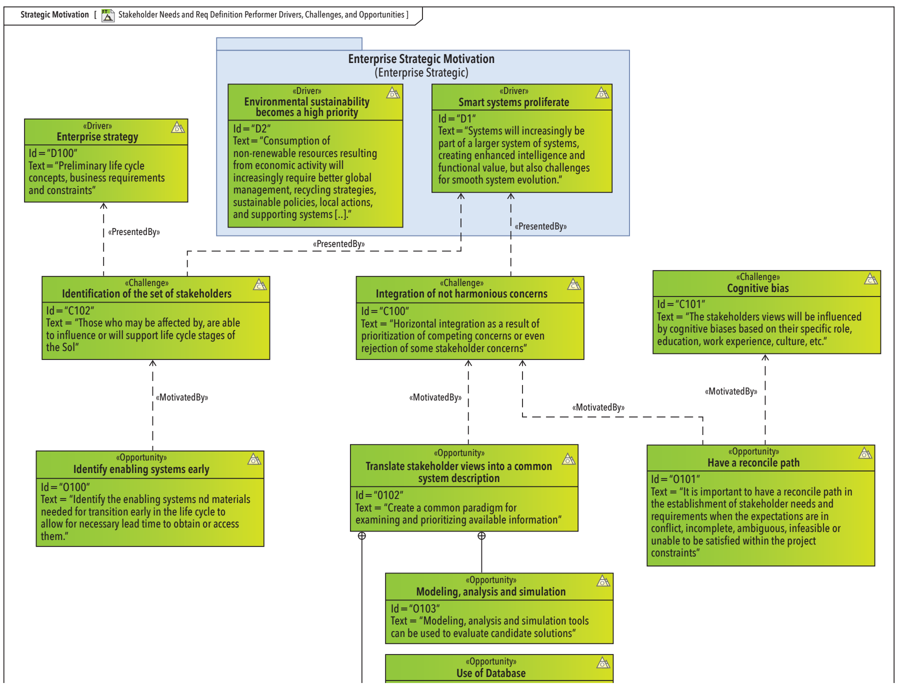


Figure 2. The motivation for the SNRD performer

and their processes, internal or external services, as well as supporting systems that are redefined accordingly. This way, organizations can effectively respond to changes in their environment, adapt their operational architectures, and ultimately train their personnel effectively while also releasing new and appropriate stakeholder requirements for these supporting systems. The ability to respond effectively to a constantly changing environment by redefining the operational and resource architecture of an organization is especially valuable when this organization employs complex supporting systems or develops, uses, maintains, or decommissions complex systems [6]. Therefore, it is significant to empower these organizations explicitly to adopt UAF.

The systems engineering reference enterprise architecture (SEREA) is an architectural model modeled in UAFML and aims to assist organizations in adopting UAF, particularly those dealing with complex systems in an environment characterized by systems of systems. To this end, SEREA is based on ISO/IEC/IEEE 15288 [7] and the INCOSE *Systems Engineering Handbook* [8]. Both documents provide a comprehensive foundation for describing a generic organizational architecture that meets the needs of these organizations. SEREA is an INCOSE Technical Paper available at the INCOSE store.

The UAF defines a series of viewpoints to describe an organization. However, not all viewpoints are utilized in SEREA. As a reference architecture, SEREA aims to provide a logical reference from which organizations can derive their own logical reference architectures and subsequently their resource and/or service architectures.

Currently, there are works supporting this endeavor; however, there is a lack of a reference architecture based on UAF. A reference architecture can serve as a template to enable any organization to implement UAF and UAFML more quickly and efficiently. This paper presents and illustrates how SEREA can be adapted to the needs of the organization. Other works, such as the UAF Guide [1] or the UAF method currently under development, will provide additional normative and methodological guidance.

2. WHAT IS SEREA AND HOW IS IT BUILT

In this paper, the terms logical and physical are used as follows: logical refers to an abstract architecture, similar to how the object oriented systems engineering method (OOSEM) defines a logical architecture versus a physical architecture [9, 465-469]. Physical viewpoints, which are organization-specific, are not included in SEREA.

The strategic viewpoints describe the motivational aspects of the organization, from vision and drivers to capabilities that need to be revealed They describe the problem space. The operational viewpoints focus on describing operational actors, their interfaces, operational activities, and the operational exchange processes, thus describing the solution space. Finally, the standard viewpoints provide a rationale for the reference architecture based on the two source documents [7,8].

ISO/IEC/IEEE 15288 provides descriptions of the processes an organization can apply throughout the system life cycle. However, the standard does not explicitly define an organizational architecture; instead, this architecture is implied through the structure of the standard. Various operational activities are logically grouped and summarized within processes. SEREA respects this logical organization of activities, including defining the desired effects and measures of effectiveness and performance as presented through the operational viewpoints as shown in Figure 1.

The INCOSE *Systems Engineering Handbook* refers to the processes described in ISO/IEC/IEEE 15288, but it provides additional context. SEREA integrates this context into its description of motivation, drivers, challenges, and opportunities, as well as through the strategic viewpoint, as shown in Figure 2.

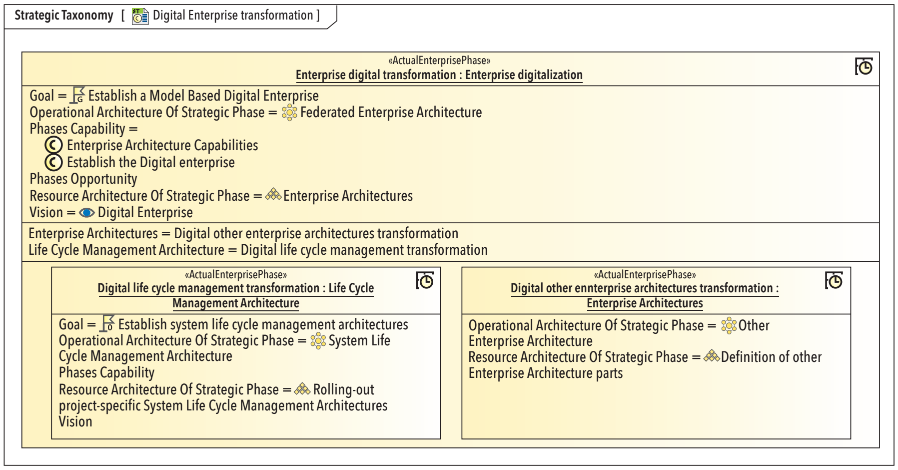


Figure 3. The enterprise considered as enterprise phases

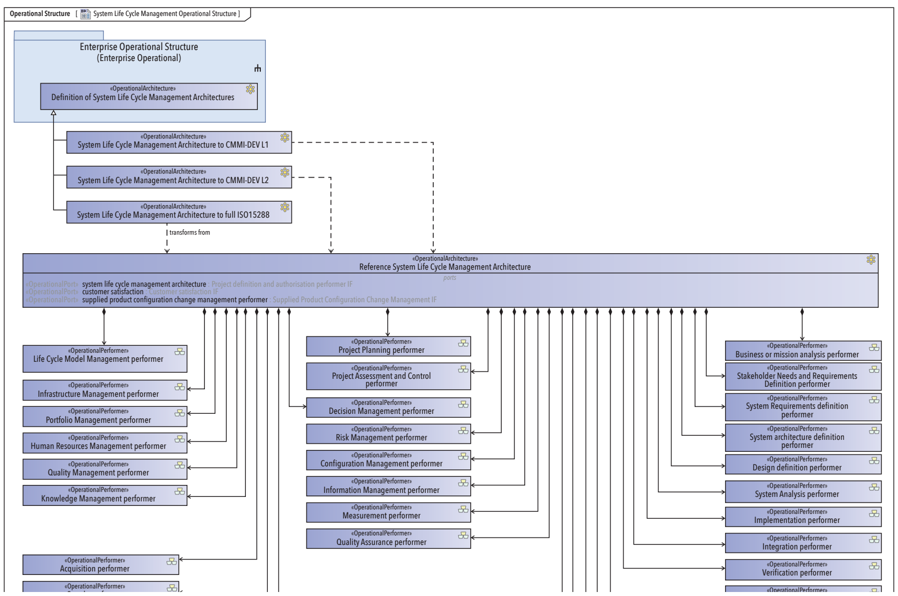


Figure 4. Operational structure of the system life cycle management architecture

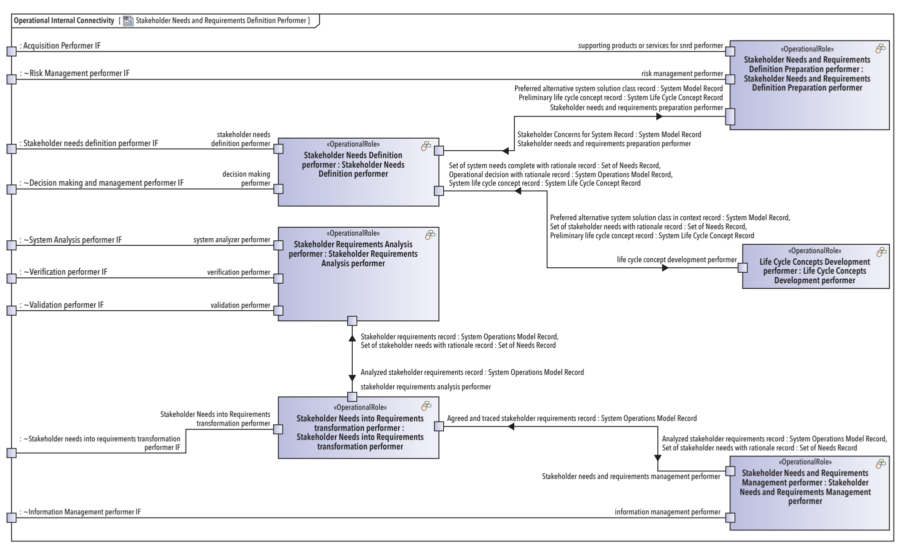


Figure 5. Connectivity view at level 1.2 of the six performers composing the SNRD performer

In defining the motivational aspects of the reference architecture, SEREA remains general enough to be broadly applicable while also being concise enough to serve as an effective reference for every user.

SEREA is structured recursively across three levels of abstraction, hence using the blackbox/whitebox approach of system thinking. The business level—level 0—describes the architecture of the enterprise as a composition of multiple architectures, as shown in Figure 3. One of these architectures, and where the focus of SEREA lies, is the reference architecture for the system life cycle management.

Level 1.1 describes the reference architecture for system life cycle management as a composition of 31 operational performers, each matching the names and scope of the ISO/IEC/IEEE 15288 processes, as shown in Figure 4. The operational level 1.2 provides a further breakdown of each of these 31 performers. The level nomenclature follows the INCOSE Needs and Requirements Manual [10, 33-34].

Level 1.2 provides a white-box view of the 31 performers defined at level 1.1. This is illustrated in Figure 5, which shows the decomposition of the SNRD performer into six additional performers. As a result, the total number of related operational performers at level 1.1 is approximately four times greater than the original 31 performers.

The operational performers at level 1.2 interact both within and outside their overarching performer. These interactions are modeled as activities that include action calls to other performers. As shown in Figure 6, the operational activity initiates calls to operational activities in external performers, which are highlighted in yellow, such as the system analysis performer, the verification performer, and the validation performer.

These called operational activities can, in turn, invoke additional external operational activities. At any given time, multiple performers can be active, either by initiating calls or by responding to calls from other performers, creating a complex network of interactions throughout the architecture. These interactions are captured in sequence diagrams for each performer, as illustrated by the example of the system requirements definition performer in Figure 7.

At level 1.2, SEREA defines the individual motivations of the operational performers in the context of the life cycle management performer at level 1.1. When specifying the drivers for stakeholder needs and requirements, SEREA builds on the drivers of the life cycle management performer and adds additional, performer-specific drivers, as shown in Figure 2. These aggregated drivers lead to specific challenges and opportunities for each performer at level 1.2. In turn, these challenges and opportunities determine the capabilities that each performer at level 1.2 must develop, as illustrated in Figure 8.

Measurements play a crucial role in managing the organization and responding efficiently to incidents and trends. SEREA proposes a set of generic measurements for each performer. Desired effects are captured in the strategic viewpoint and evaluated using effect measurements, as shown in Figure 11, and effectiveness measurements (MoEs), as shown in Figure 8. The former assess the achievement of the desired effects, while the latter evaluate how effectively a capability contributes to these effects. Additionally, measures of performance (MoPs), as referenced in [8, 93-98] and shown in Figure 1, are recorded for each performer in the operational viewpoint. These measurements assess the effort the organization invests in achieving the desired effects.

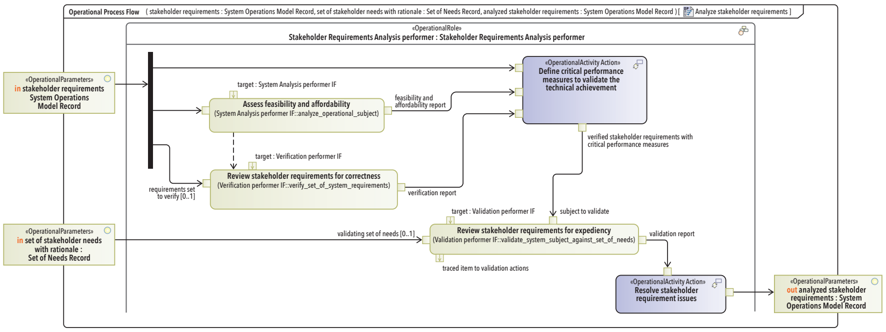


Figure 6. Operational activity with operation calls to other performers

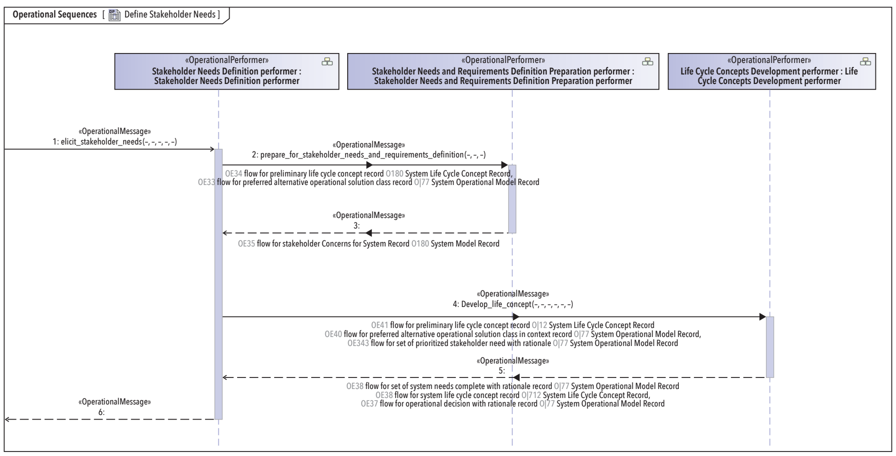


Figure 7. Interaction between performers in the SNRD performer.

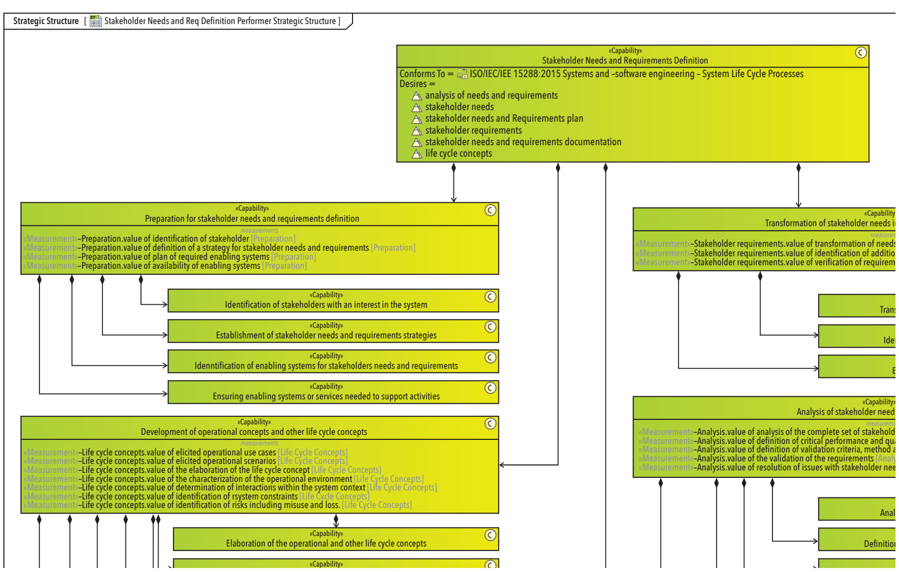


Figure 8. Capability of the SNRD with decomposed capabilities as per ISO 15288

3. THE USE OF SEREA IN THE ENTERPRISE

SEREA is designed for gradual implementation by organizations. It defines the necessary performers for each business phase, aligning with the capability maturity model integrated (CMMI) DEV model [11]. The proposed sequence of business phases is as follows:

1. Adaptation of a company-wide reference architecture for life cycle management to CMMI level 2.
2. Adaptation of a company-wide reference architecture for life cycle management to CMMI level 2.
3. Adaptation of a company-wide reference architecture for life cycle management to CMMI level 2.
4. Implementation of a project-specific resource architecture for life cycle management at CMMI level 2.

The same approach applies to achieving a higher degree of formality at CMMI DEV level 3, as shown in Figure 9.

The strategic viewpoint of SEREA is designed to be adaptable by first defining the company's vision for system life cycle management and identifying the current drivers within the specific context. Depending on the agreed level of formality regarding CMMI DEV, only a portion of the capabilities provided by SEREA needs to be considered, as shown in the capability phase diagram in Figure 10. SEREA offers a reference architecture that covers the full scope of processes defined in ISO/IEC/IEEE 15288, but does not include all of the capabilities proposed by CMMI DEV.

The vision and drivers in SEREA are defined in general terms and represent a universal scenario that serves as an initial orientation and placeholder in the model, as shown in Figure 2. The challenges and opportunities associated with each driver can be defined, modified, or supplemented, as illustrated in Figure 11. These opportunities are then linked to the goals and objectives of the relevant performer. The goals themselves are derived from the company's vision, ensuring that opportunities aligned with the company's vision are prioritized. Ultimately, the capabilities that each performer must develop are defined based on the opportunities examined. The desired effects and their corresponding measurements are aligned with the performer's goals to ensure consistency and

OCTOBER 2025
VOLUME 28/ ISSUE 5

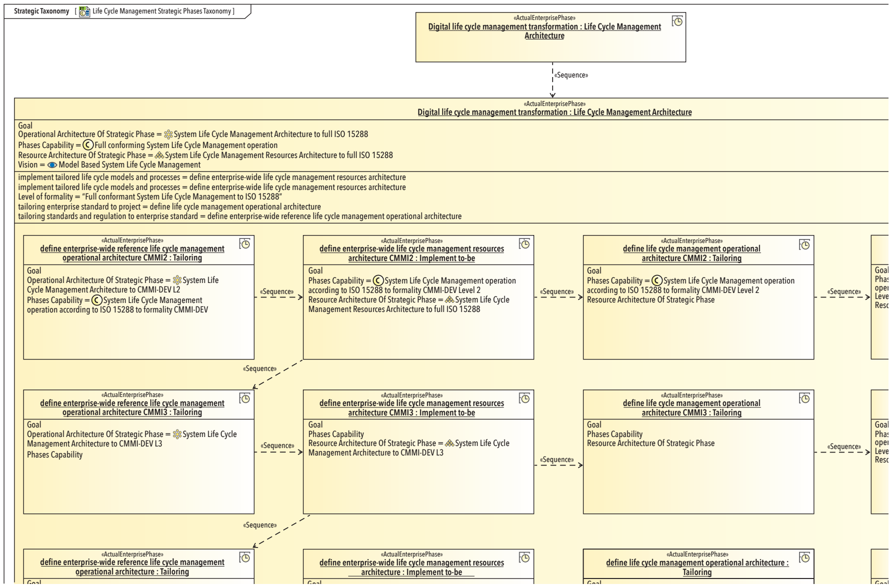


Figure 9. Enterprise phases as recommended by SEREA in accord with the CMMI DEV

effectiveness.

SEREA provides a breakdown of the capabilities and assigns them to the performers, as shown in Figure 12. While SEREA does not define opportunities and goals for these capabilities, they can be developed recursively using the previously described approach. Once new capabilities are defined, they can be related to each other to ensure that dependencies are taken into account when creating a training plan.

The operational viewpoint of SEREA maps the capabilities from the strategic viewpoint to the activities of each performer in the operational viewpoint, as shown in Figure 13. These activities should be adapted to reflect the newly defined capabilities, while maintaining alignment with the requirements of ISO 15288, which underpins SEREA.

Operational activities are assigned to the performers, as shown in Figure 14, and both the performers and the assignments can be adjusted accordingly.

Ultimately, the connectivity between operational performers and operational exchanges can be adjusted accordingly, as shown in Figure 5. The adaptation of SEREA can follow various approaches, with the most comprehensive being a two-step process, as illustrated in Figure 15. First, SEREA is adapted to a company-wide reference architecture. This reference architecture defines the resource and service architecture for a specific business phase. The performers in the operational viewpoint of the reference architecture are implemented through resource configurations. These configurations consist of personnel, supporting systems, and services. If necessary, the company-wide resource architecture can be further adapted for a specific project.

Both the service and resource views in SEREA are considered physical views, based on the reasoning that an organization

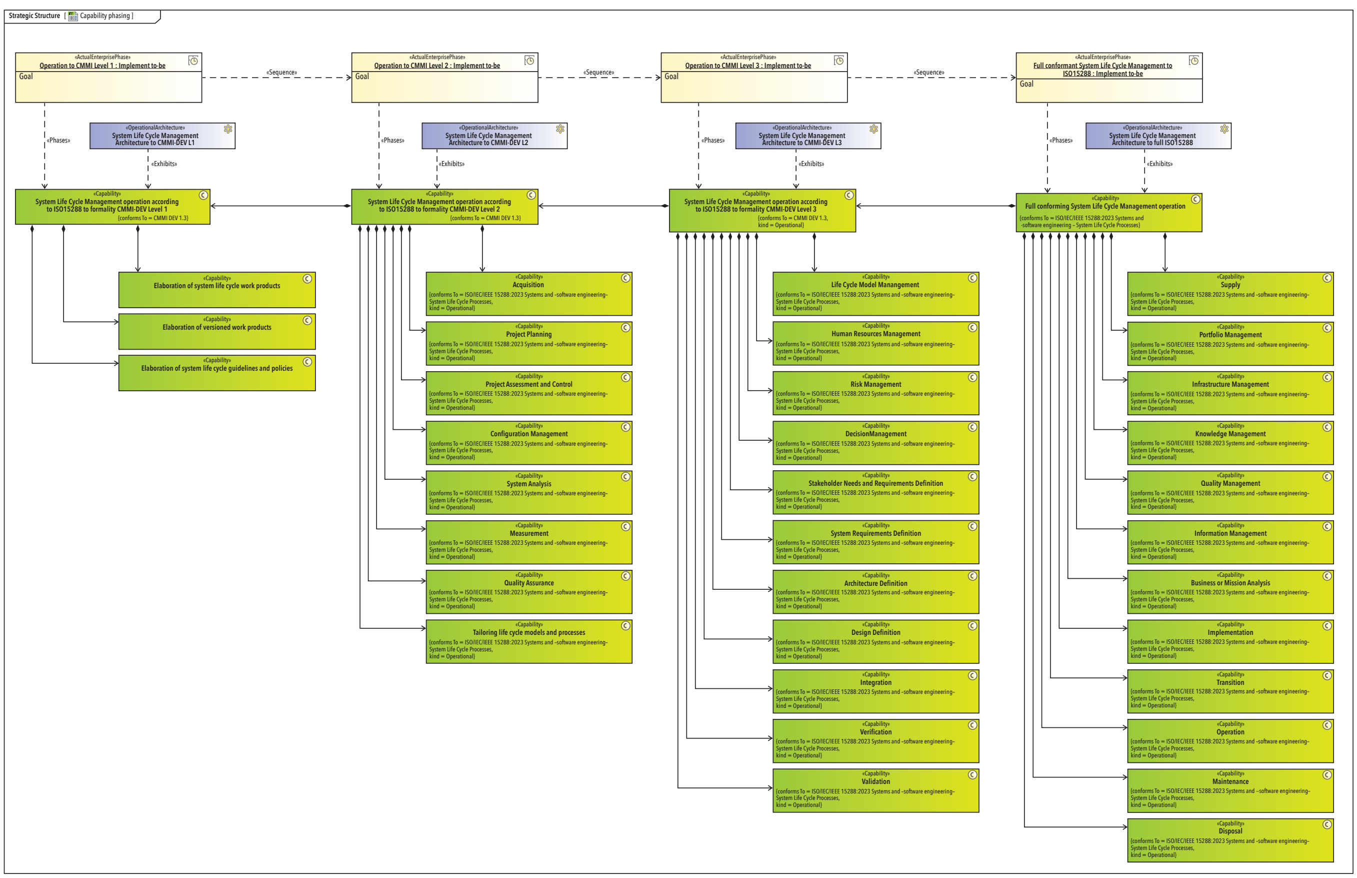


Figure 10. Capabilities to be exposed for each CMMI DEV level

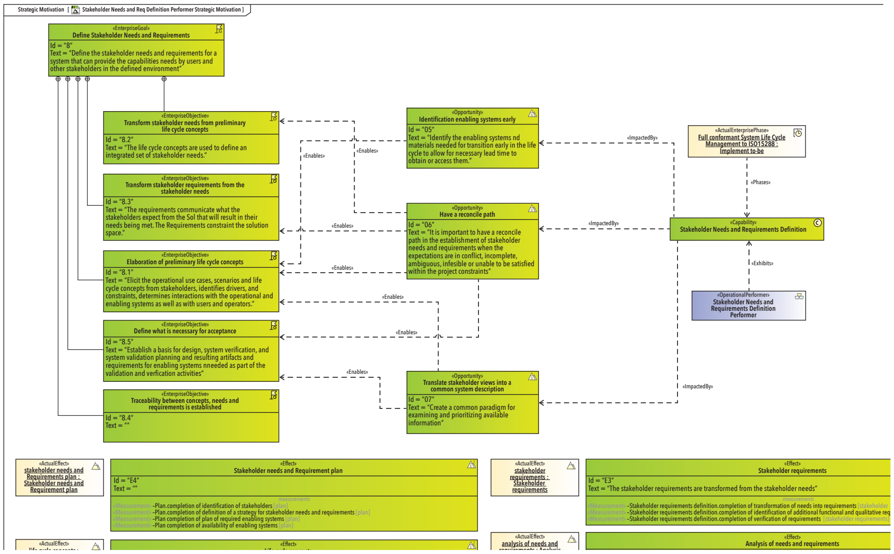


Figure 11. Correspondence between opportunities and objectives of the system life cycle management architecture for the SNRD, derived from the vision of the enterprise

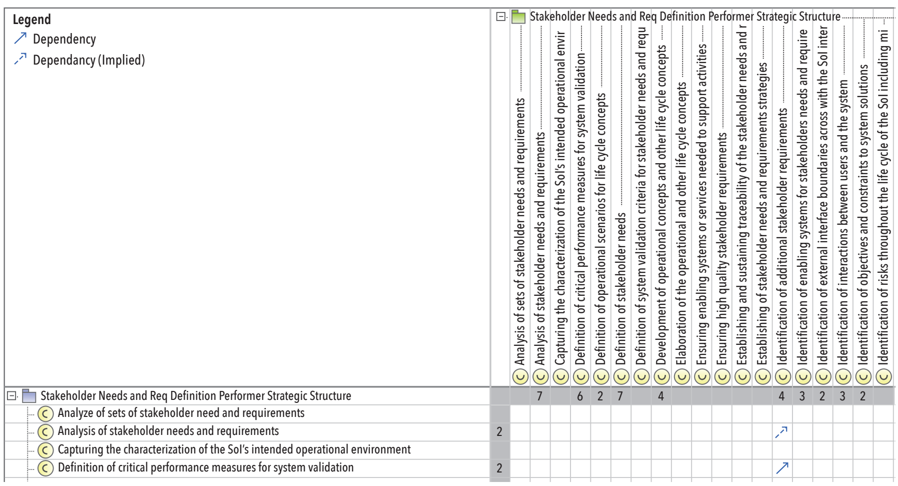


Figure 12. Dependencies between the capabilities of the SNRD in a matrix

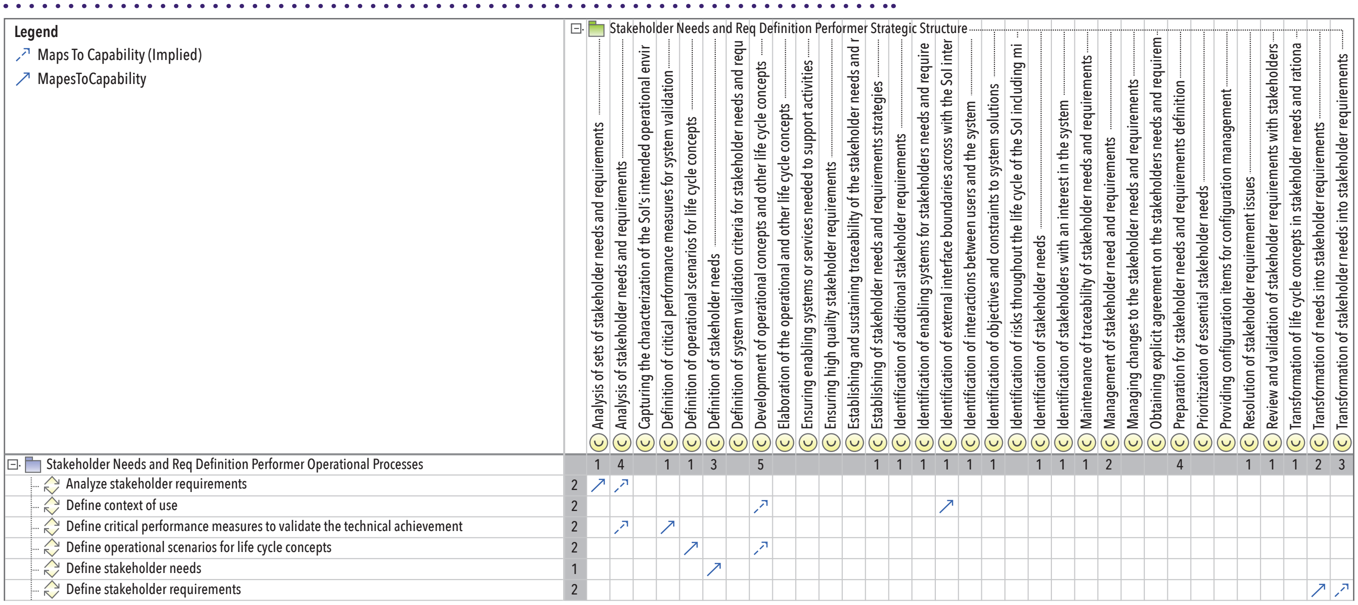


Figure 13. Allocation between capabilities and operational activities of the SNRD performer

must deliver certain capabilities, regardless of whether they are implemented through a resource configuration or a service. The implementation may evolve over time, even if the capabilities themselves remain unchanged.

References to SEREA can be maintained through dependencies to document the rationale for adaptation and to facilitate future updates. Alternatively, adapted architectures can be traced back to the standards if needed. In cases where defining a company-wide reference architecture is not required, adaptation can proceed directly from SEREA to a company-wide resource and/or service architecture.

4. DISCUSSION

Organizations using any modeling tools can benefit from SEREA, even though it is currently only available in CATIA Magic. The model can be created or modified by following the SEREA viewpoints. With the anounced release in 2027 of UAFMLv2, based on SysMLv2, a new version of SEREA will be released, that will improve tool independence.

SEREA is not intended as a methodology but as a template for modeling organizations. The OMG UAF continuously publishes formal methodologies through the UAF Guide and UAF Example to support modelers. Users of SEREA can extend their reference or resource architectures with additional details from standards such as ISO 42020 [12], ISO 29148 [13], or other works like the INCOSE Needs and Requirements Manual [10], the INCOSE Guide to Writing Requirements [14], industry-specific standards, and other relevant sources. In such cases, it is advisable to maintain the specification viewpoint and trace it back to the capabilities as shown in Figure 10.

5. CONCLUSION AND OUTLOOK

The adoption of UAF not only leads to a standardized description of the enterprise architecture, thereby improving internal and external communication, but also enables organizations to manage changes effectively. It allows organizations to trace processes from the organizational vision and environmental drivers all the way to operational and resource architectures, personnel capabilities, and the specifications of supporting systems. Changes in the business environment can initially be modeled, versioned, and managed in the organizational model, leading to the redefinition of new or altered capabilities and requirements for supporting systems. Consequently, training plans that teach the necessary skills can be defined, and necessary suppliers can be procured as needed. Further specification of the supporting systems in a system model in SysML enables ongoing tracking of this chain of effects through the system model itself. The integration of organizational and system models will make it possible to thoroughly analyze potential organizational solutions and plan changes as proposed by [15] and [16]. With the upcoming introduction of SysMLv2 and UAFMLv2, the integration of organizational and system models with analysis models will significantly ease the process, broadening the intended scenario. Although this may not be fully realized in the next two to five years, now is the

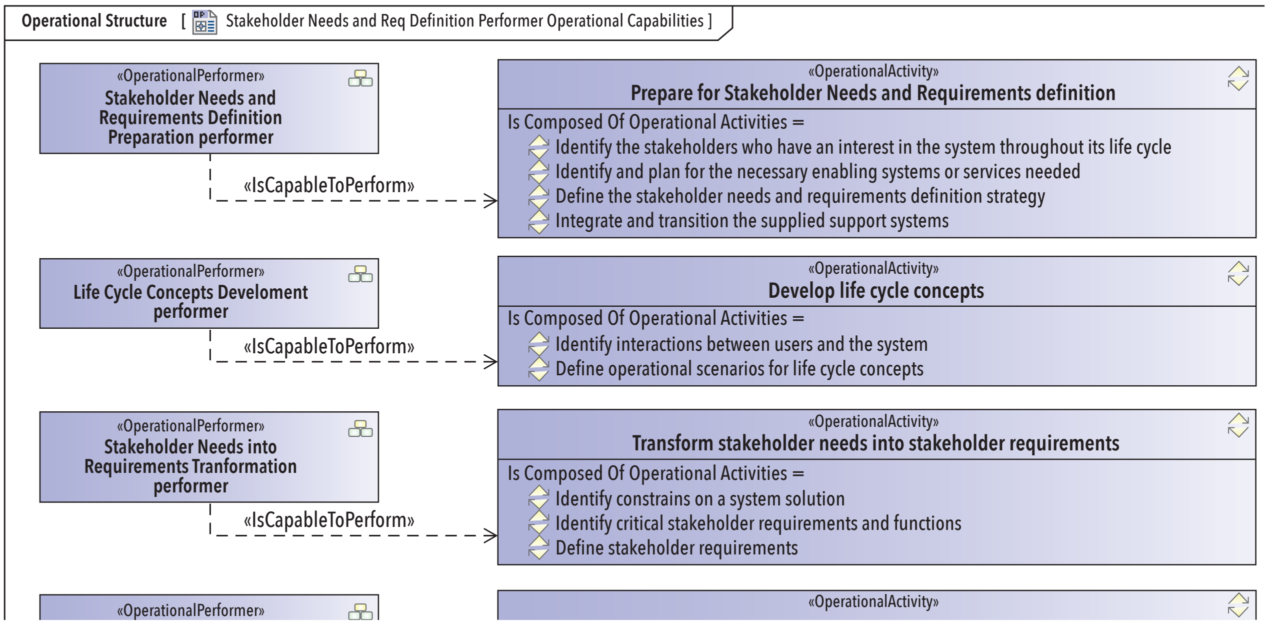


Figure 14. Allocation between operational activities and operational performers

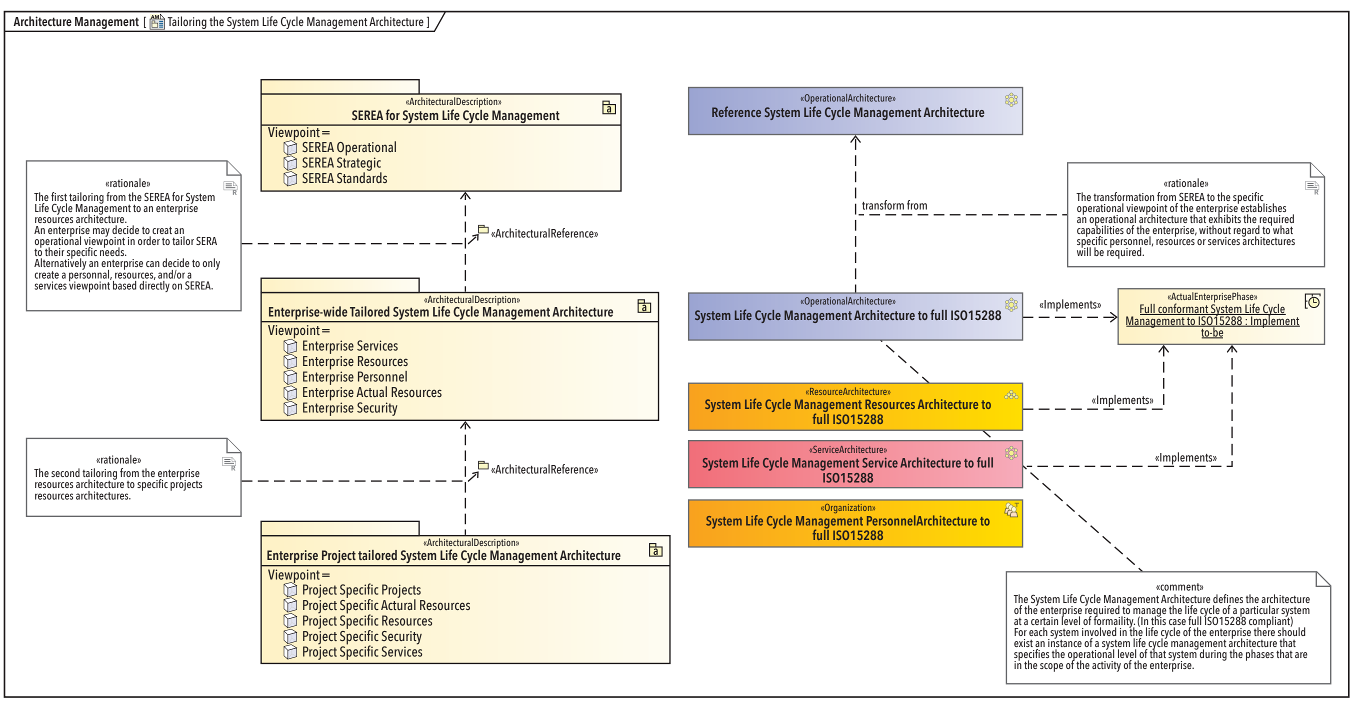


Figure 15. A two-step approach to tailoring SEREA for a specific enterprise and further for a specific project within the enterprise

right time to replan the company's digital engineering strategy.

There is a notable increase in the use of artificial intelligence (AI) agents for tasks related to systems engineering, particularly in requirements management. However, a thorough analysis is necessary to ensure reliable results for each individual AI agent. Even with large language models (LLMs) capable of processing unstructured data, task-specific prompts must be carefully crafted and validated. A LLM equipped with integrated organizational and system models, modeled in UAFMLv2 and SysMLv2, would have a rich, structured data source, enabling the AI agent to understand the organization and associated systems while reducing the effort required to create task-specific prompts. ■

ACKNOWLEDGMENTS

In 2024 the author established the GfSE UAF Working Group [17] to promote knowledge about UAF in German-speaking countries. SEREA is the first project of the group aimed at cultivating a community of UAF practitioners while providing a reference architecture for organizations looking to explore UAF's potential. I am grateful for the discussions and long meetings with group members that have made SEREA what it is today and what it can become.

Special thanks to the INCOSE Systems Engineering Laboratory and Dassault [18] for providing access to CATIA Magic for the group, allowing us to develop SEREA. I am deeply appreciative of INCOSE and Dassault's support in providing the necessary infrastructure for the efforts of the Working Group.

REFERENCES

01. About the Unified Architecture Framework Specification version 1.2. 2022. Retrieved 05 21, 2025, from About the Unified Architecture Framework Specification version 1.2: https://www.omg.org/spec/UAF/1.2/About-UAF/.
02. About the OMG Systems Modeling Language Specification version 1.7. 2024. Retrieved 05 21, 2025, from About the OMG Systems Modeling Language Specification version 1.7: https://www.omg.org/spec/SysML/1.7.
03. About the OMG System Modeling Language Specification Version 2.0 beta 2. 2024. Retrieved 05 21, 2025, from About the OMG System Modeling Language Specification Version 2.0 beta 2: https://www.omg.org/spec/SysML.
04. Siddique, I. 2024. "Transforming Enterprises Through UAF Strategic Planning: Frameworks and best Practices." *International Journal of Progressive Research in Engineering Management and Science*, 4, 50-60.
05. Ormo, H. 2024. UAF in der Produktentwicklung von Systemen. *Tag des Systems Engineering* (pp. 102-109). Leipzig, GfSE: Verlag.
06. *INCOSE Vision35, The Enterprise Environment*. 2022. Retrieved 05 21, 2025, from INCOSE Vision35, The Enterprise Environment: https://violin-strawberry-9kms.squarespace.com/the-enterprise-environment.
07. ISO/IEC/IEEE. 2023. ISO/IEC/IEEE 15288 Systems and software engineering — System life cycle processes. Geneva, CH: ISO.
08. Walden, D. D., T. M. Shotell, G. J. Roedler, B. A. Delicado, O. Mornas, Y. Yew-Seng, and D. Endler. 2023. *Systems Engineering Handbook, a Guide for System Life Cycle Processes and Activities*. San Diego, US-CA: International Council on Systems Engineering.
09. Friedenthal, S., A. Moore, and R. Steiner. 2012. *A Practical Guide to SysML*. Waltham, US-MA: Elsevier.
10. Weatcraft, L., T. Katz, M. Ryan, and R. B. Wolfgang. 2022. Needs and Requiremnts Manual. San Diego, US-CA: International Council on Systems Engineering.
11. ISACA. 2024. *What is CMMI?* Retrieved 05 21, 2025, from What is CMMI?: https://cmmiinstitute.com/cmmi/intro.
12. ISO/IEC/IEEE. 2019. ISO/IEC/IEEE 42010 Systems and Software Engineering – Architectural Description. Geneva, CH: ISO.
13. ISO/IEC/IEEE. 2011. ISO/IEC/IEEE 29148 Systems and Software Engineering – Life Cycle Processes – Requirements Engineering. Geneva, CH: ISO.
14. Ryan, M., L. Wheatcraft, R. Zinni, J. Dick, and K. Baksa. 2022. Guide to Writing Requirements. San Diego, US-CA: INCOSE.
15. Abhaya, L. 2021. UAF (Unified Architecture Framework) based MBSE (UBM) Method to Build a System of Systems Model. *INCOSE International Symposium* (pp. 227-241). International Council on Systems Engineering.
16. Bankauskaite, J., Z. Strolia, and A. Morkevicius. 2023. "Towards an Approach to Co-execute System Models at the Enterprise Level." *INCOSE International Symposium* (pp. 334-348). International Council on Systems Engineering.
17. Gesellschaft für System Engineering e. V. (n.d.). Unified Architecture Framework (UAF). Retrieved 05 21, 2025, from Unified Architecture Framework (UAF): https://gfse.org/de/?option=com_sppagebuilder&view=page&id=428.
18. INCOSE. Systems Engineering Laboratory. (n.d.). Retrieved 05 21, 2025, https://www.incose.org/Learn/se-laboratory.

ABOUT THE AUTHOR

Hugo Ormo is with NTT DATA Deutschland SE, Hans-Döllgast-Str. 26, 80807 München, DE. He established the GfSE UAF Working Group to promote knowledge about UAF in German-speaking countries. SEREA is the first project of the group aimed at cultivating a community of UAF practitioners while providing a reference architecture for organizations looking to explore UAF's potential.

PBSE Data Initialization Framework and Practice by Using LLM

Degang Liang, liangdegang@comac.cc; and **Baoyu Dong,** dongbaoyu@comac.cc

[Editor: This paper was originally accepted for and presented at the 35th Annual INCOSE International Symposium, Ottawa, Canada, 26-31 July 2025.]

ABSTRACT
With the increasing complexity of commercial aircraft and the rapidly changing market demands, the system engineering development pattern extensively adopted by aircraft OEM has evolved from the traditional document-based systems engineering (DBSE) to model-based systems engineering (MBSE) and pattern-based systems engineering (PBSE). MBSE employs models to describe products, while PBSE builds upon MBSE by utilizing engineering patterns, which are validated in advance, to enhance the efficiency and quality of data production in both MBSE and DBSE. However, during PBSE engineering practices, we have observed certain challenges, such as the barriers to initializing product S* models and the low efficiency in generating instances. Artificial intelligence for systems engineering (AI4SE) is an emerging concept aimed at creating a more efficient and user-friendly systems engineering implementation environment through the integration of artificial intelligence (AI), machine learning (ML), and related technologies. This paper explores the application of AI4SE in real-world engineering projects by leveraging large language models (LLMs) to develop a methodology that reduces the deployment threshold of PBSE for enterprises and enhances the efficiency of instance generation.

KEYWORDS: PBSE, AI4SE, artificial intelligence, large language models

INTRODUCTION

Systems engineering, while being one of the proven and effective methodologies for addressing the development of complex products (Lu 2008, He 2023, Galli 2020, Brian 2022, Stockman 2010). However, as Bailey pointed out, many large-scale complex engineering projects that failed over the past 30 years also employed systems engineering (Bailey 2011). The article further explains that a significant portion of these failures can be attributed to engineers' entrenched work habits and the neglect of consistent data management within projects. Nonetheless, it remains essential to address certain issues inherent to the discipline of systems engineering.

Initially, the practice of systems engineering methodology was primarily carried out through documents, which led to two key issues: expressiveness and looseness (Zirnstein 2023). Specifically, these issues manifest in the following ways:

Expressiveness Issue: The expressiveness issue arises because the carrier of product feature descriptions is primarily descriptive technical documentation rather than graphical representations, which may lead to some degree of confusion in understanding. Engineers use written language to describe product without a unified or detailed set of rules. While written language inherently offers flexibility in interpretation, allowing for multiple understandings, the final comprehension depends entirely on the recipient of the information. This implies that during the process of transmitting product information through technical documents, the completeness and accuracy of the data transferring largely depend on individual factors.

Looseness Issue: The issue of looseness arises because the description of elements within product often requires numerous descriptive statements, which are typically scattered across design documentation. This issue becomes particularly prominent in the development of complex engineering products, where engineers are often unable to refer to product in a singular, unified form. In general, engineers describe product elements by pointing to a collection of loosely connected texts, which leads to varying degrees of arbitrariness and subjectivity. Textual representation fails to assign a fixed pattern to the information, resulting in inconsistencies in interpretation and understanding.

Subsequently, relevant scholars proposed the use of models to represent product information (i.e., MBSE) to solve the issues of expressiveness and looseness. While MBSE has accelerated the application of systems engineering, it has also introduced usability issues and model complexity problems, including:

Usability issues: The MBSE methodology requires enterprises to invest additional learning and usage costs. Learning costs refer to the need for acquiring proficiency in modeling languages, while usage costs involve the purchase of specialized modeling software. For enterprises, this means that the implementation of MBSE in several initial projects will lead to increased costs and delays in product delivery to customers, which to some extent hinders the widespread adoption of MBSE.

Model complexity issues: MBSE requires a solid understanding of how the model functions in order to fully leverage its potential. If users do not understand how the model works, they will struggle to trust those who have developed it. Currently, especially in global complex product engineering area, only a small group of experts are capable of effectively utilizing MBSE in conjunction with their own product development expertise. This creates additional communication costs among engineers during the implementation of MBSE within enterprises, while also increasing workload and responsibility.

Therefore, Schindel proposed using patterns to simplify the model generation process, which can effectively reduce the cost investment for enterprises in systems engineering (Schindel 2007, Sherey 2006). This approach has yielded some positive results to a certain extent, but it also introduces three issues: over-complexity (Hohpe 2004), dependency (Buschmann 1996), and limited innovation (Schindel 2015):

Over-complexity issue: When integrating large enterprise systems, excessive use of patterns may lead to an overly complex product system. At times, engineers may attempt to address all potential issues by creating more patterns, but this can result in a system that is excessively complicated, making it difficult to understand and maintain.

Dependency issue: If a product system heavily relies on specific patterns, any issues with these patterns or the need for changes can lead to the failure of the entire system or require large-scale modifications. This, in turn, reduces the efficiency of data scale generation.

Innovation limitation issue: The pattern-based systems engineering approach may constrain engineers' creativity, as they may be more inclined to rely on existing patterns rather than attempting to develop more creative and effective solutions.

Compared to the previous DBSE and MBSE approaches, the threshold for enterprises to implement PBSE has been raised once again. Of course, the cost of product development for each project can indeed be significantly reduced in the later stages. In addition to possessing traditional systems engineering concepts, specific product development knowledge, and modeling languages, engineers now also need to initialize the product's S* patterns. This requires engineers, at the outset of practicing PBSE, to extract key data from the large volumes of technical documents and specific models for identification, analysis, and refinement. Based on these key data, they must establish interrelationships between them, ultimately forming specific S* patterns. From a practical operation perspective, the configuration of S* models and patterns is typically carried out from scratch using tools like EXCEL spreadsheets or MBSE software plugins. Similar to the earlier transition from DBSE to MBSE, there is a lack of transitional tools from MBSE to PBSE, which has actually hindered the development of systems engineering. As a result, many engineers, projects, and companies end up at beginning. Therefore, although PBSE, in the long term, can significantly reduce the cost of model construction, the entire process of pattern initialization and subsequent updates will inevitably increase the enterprise's resource investment in this area (INCOSE 2018).

Following PBSE, concepts such as intelligent-based systems engineering (IBSE), Tolk (2011), and AI4SE, which leverage artificial intelligence technologies to empower systems engineering, have been proposed in succession, aiming to address the growing cost overflow faced by enterprises during S* pattern management (Stockman 2010, McDermott 2020, McDermott 2021). Among these, Abiodun has attempted to use a nonlinear regression model with artificial neural networks (ANN) to successfully identify "patterns" in the manufacturing engineering domain (Abiodun 2019), providing an innovative approach for integrating AI technologies into pattern recognition within PBSE.

Over the past few decades, computer technology has undergone several significant innovations, from machine learning to deep learning, eventually leading to the advent of the AI era. With the release of ChatGPT at the end of 2022, AI entered the era of large language models (LLMs). LLMs are large-scale neural networks built upon the transformer architecture, which was proposed by Google in 2017 and initially applied to machine translation (Vaswani 2017). Today, nearly all large language models are based on the transformer architecture. The transformer-based encoder-decoder structure is shown in Figure 1, with the left side representing the encoder and the right side representing the decoder. Both components are made up of several basic transformer blocks (depicted by the gray boxes).

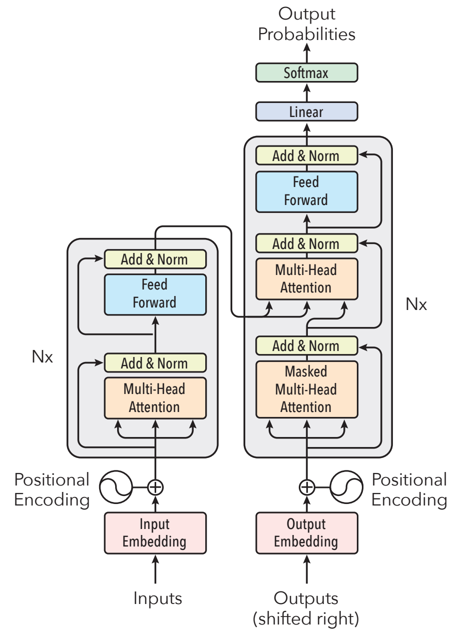


Figure 1. The transformer-based encoder-decoder structure

THEORY, FRAMEWORK, AND PLATFORM DEVELOPMENT

The construction of the S* pattern is one of the critical elements for implementing PBSE. This paper adopts a "participatory" collaboration model (Cummings 2021), leveraging transformer-based LLMs to replace engineers in the initialization phase of the S* model within PBSE. This approach enables the rapid establishment of a design space adhering to the S* pattern, thereby reducing the upfront costs for enterprises in applying PBSE and mitigating its impact on project progress to some extent.

S Meta-Model, S* Model, S* Pattern*

The S* meta-model is a science- and mathematics-based framework designed to describe the intrinsic structure of systems of interest (SoI). Analogous to how the "standard model" in physics is used to describe or explain observed phenomena, the S* meta-model serves as a "standard model" refined and distilled through the methodologies of model-based systems engineering (MBSE) and the system representation standard (ISO 10303-233:2012). It is used to describe system phenomena and enables a more scientific understanding of systems by clarifying the relationships between model data. This meta-model represents a high-level abstraction of system phenomena and encompasses 13 classes and 4 types of coupling relationships. The classes include

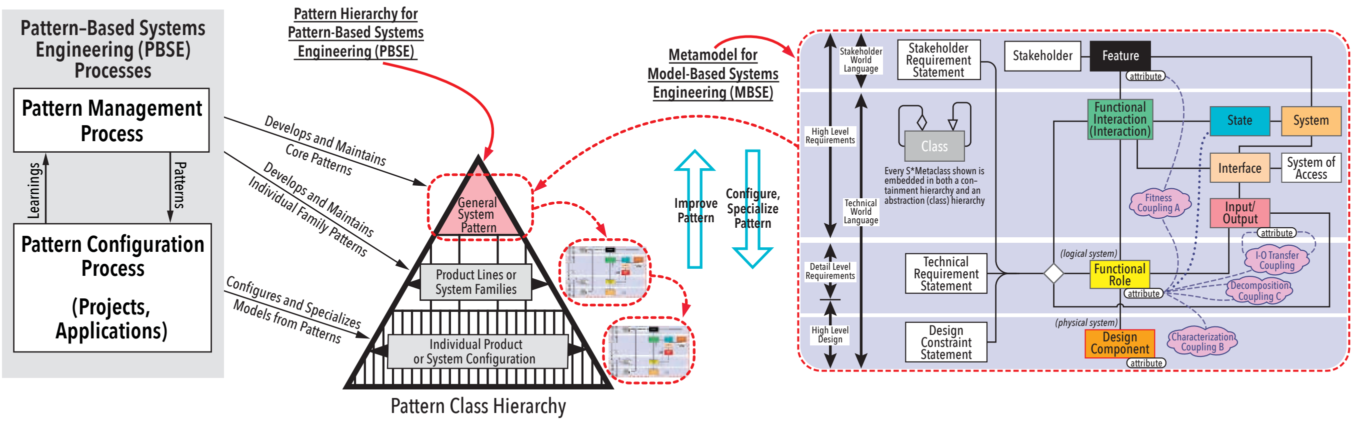


Figure 2. S meta-model, S* model, S* pattern*

feature, functional interaction, functional role, state, design component, interface, requirement, attribute, and attribute coupling, among others. These elements collectively provide a comprehensive framework for modeling system behavior, structure, and interactions, enabling precise and systematic analysis of complex systems.

The S* model refers to an MBSE model that adheres to the S* meta-model. An S* pattern, on the other hand, represents a broader system configuration space constructed using S* models, extending beyond specific system instances. S* patterns are fully parameterized, allowing for the rapid configuration of additional specific S* models, thereby enabling efficient reuse. Once enterprises establish S* patterns across their business domains, product families, and production lines, these patterns can be digitally mapped and replicated to preserve the configuration relationships among S* models within the S* pattern. This ensures that when disruptions occur—whether internal or external, at various levels across different business domains—organizations can leverage "pattern management" to analyze, isolate, and address disturbances swiftly. This approach confines the impact within controlled areas and hierarchy levels, achieving agile responses through effective "configuration" management (Schindel 2015).

The relationships among the S* meta-model, S* models, and S* patterns are illustrated in Figure 2. The architecture, application methods, and specific case studies of S* models and patterns can be found in the works published by Schindel and colleagues. These resources provide detailed insights into the practical implementation and benefits of the S* framework within various system engineering contexts (Schindel 2015, Schindel 2011, Sherey 2006, Schindel 2002, Dove 2017a, Dove 2017b, Dove 2018, Cook 2015, Zielske 2022).

Technical Framework

Integration frameworks for LLMs, such as FastGPT, LangChain (Topsakal 2023), and LlamaIndex (Zirnstein 2023), have significantly facilitated the practical application of these models. Secondary development on this basis—such as integrating external local knowledge bases—can lower hardware and data volume requirements while keeping costs under control. This paper adopts an approach that incorporates a local private deployment of open-source LLMs to construct an S* model initialization platform. To generate more complete and accurate S* models, a "prompt template" method is employed to help users delineate the boundaries of queries posed to the language model. This approach ensures that the generated source data possesses direct engineering value, addressing criticisms of "hallucinations" often associated with LLMs. The source data used is mapped into S* meta-model, which is ultimately transformed to S* model, and delivered to PBSE process. The framework is shown in Figure 3.

Platform Development

Our team has conducted experiments on the "S* data initialization framework based on LLMs" specifically in the field of civil aircraft product development. Because most of engineering sources were produced in Chinese, the core LLMs used in this framework are the open-source Deep-seekR1-70B (Daya 2025) and QwQ-32B (Bai 2023) with the FastGPT integrated to support the entire work-flow. The engineering database is primarily sourced from organizational assets of a civil aircraft manufacturing company, while the prompt template library is constructed by selecting typical systems engineering scenarios to correspond with the design requirements. Specifically, the framework:

Interface: We use the pre-configured UI from FastGPT to develop the user interface in order to specify the product system and S* meta-model. Once these two parameters are selected, the specific template is called, which connects to the prompt template library module, and the configuration result is combined with the template to form the prompt, which is then pushed to the LLM. This ensures that the prompt data template remains highly consistent for each interaction, leading to uniformly consistent feedback results.

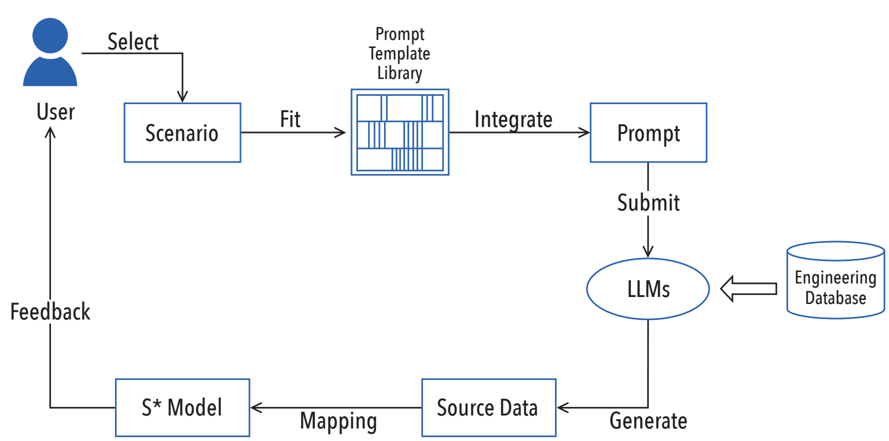


Figure 3. Framework of S data initialization based on LLMs*

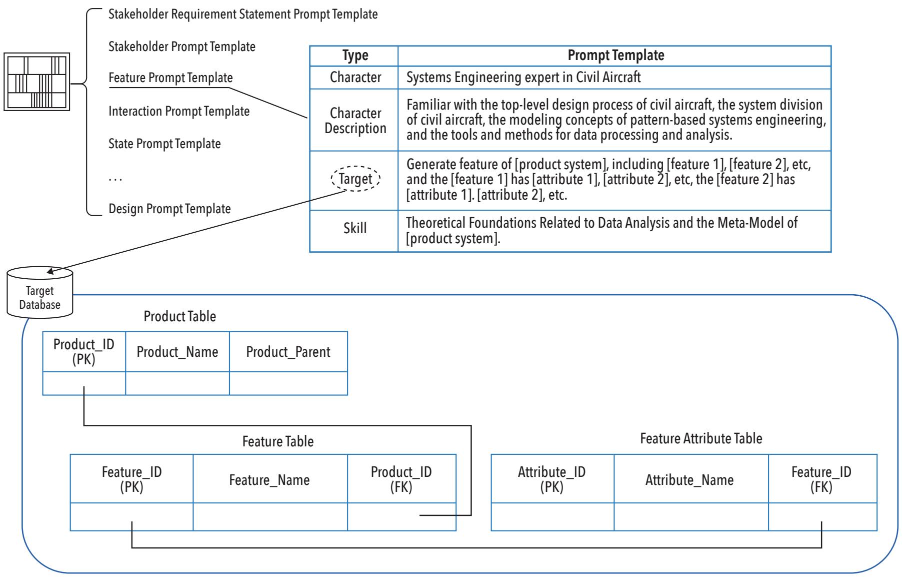


Figure 4. Example of simplified framework of prompt template library

Database: Our team directly utilizes libraries such as PyTorch and NumPy to cleanse and process the organizational assets within the enterprise. These assets include product development processes, design specifications, data standards, material specifications, process standards, as well as instantiated product requirements, functions, interface reports (entries), technical manuals, and more. This data cleansing ensures that the trained LLM can adapt to the current research and development environment.

Library: According to theory of prompt engineering (Jules 2023), we developed a prompt template (pattern) library, which contains a series of S* meta-model element prompt templates, for typical scenarios in systems engineering within the context of civil aircraft product development, as shown in Figure 4.

	A	B	C
1	**Feature**	**Feature Attribute**	**Feature Attribute Value**
2	Geometry	Layout	
3		Distance between Nose LG and Main LG	
4		Main landing gear wheel track	
5		...	
6	Function	Nose landing gear wheel retraction anti-rotation mechanism	
7		Provide landing gear door open and closed status information.	
8		Maximum time from cockpit operation to cockpit indication of completion for the landing gear system under single-engine failure conditions.	
9		...	
10	Non-function	Maximum wet weight	
11		Maximum weight of the control system	
12		Minimum clearance between electronic and electrical equipment and flammable liquid pipelines	
13		...	

Figure 5. Feature table for the S model of landing gear system with uninitialized data*

CASE STUDY

Our team selected the initialization scenario of the "feature" data for the S* model of several products, which including aircraft itself, landing gear system, integrated modular avionics (IMA) system, and power plant system from an aircraft breakdown structure, to conduct engineering experiments. We utilized only the "features" as examples for engineering validation. The initialized data includes the "features," "feature attribute," and "feature attribute value." The following takes the landing gear system as an example only.

Firstly, according to the definition of "features" in the S* meta-model, both the "attributes" and "attribute value" of a feature are derived from "requirements." Therefore, in this experiment, the landing gear system's requirement report is designated as the "local knowledge base," and the complete prompts are synthesized based on the "S* data initialization framework supported by LLM." The S* model features of the landing gear system are categorized into three types: "geometric feature," "functional feature," and "non-functional feature." Among them, "geometric feature" include attributes such as layout, maximum tail-down landing angle, and main landing gear wheel track; "functional feature" include attributes such as the nose landing gear wheel retraction anti-rotation mechanism, the maximum time from cockpit operation to cockpit indication of completion for the landing gear system under single-engine failure conditions, etc; "non-functional feature" include attributes such as maximum wet weight, maximum weight of the control system, the minimum clearance between electronic and electrical equipment and flammable liquid pipelines, etc. All of four systems' feature tables as shown in Figure 5.

After invoking the specified prompt template, we input the prompt "Please list the specific data for the geometric feature, functional feature, and non-functional feature of the landing gear system's S* model." in the dialogue box and received a response from the LLM, as shown in Figure 6.

As can be seen, the response from this attempt did not fully cover all the feature attribute values. After multiple adjustments to the prompt template content and several iterations, the results from the LLM were processed using a pre-compiled program

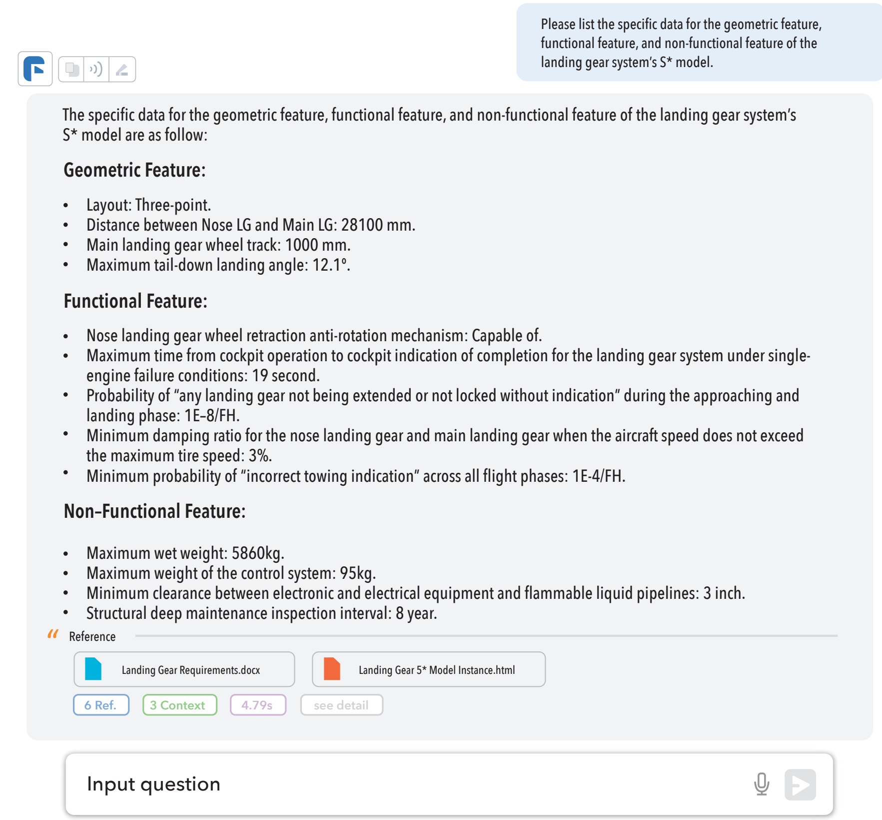


Figure 6. Screenshot of answer from LLM

	A	B	C
1	**Feature**	**Feature Attribute**	**Feature Attribute Value**
2	Geometry	Layout	Three-point
3		Distance between Nose LG and Main LG	28100 mm
4		Main landing gear wheel track	1000 mm
5		…	…
6	Function	Nose landing gear wheel retraction anti-rotation mechanism	Capable of
7		Provide landing gear door open and closed status information.	Capable of
8		Maximum time from cockpit operation to cockpit indication of completion for the landing gear system under single-engine failure conditions.	19 sec.
9		…	…
10	Non-function	Maximum wet weight	5860kg
11		Maximum weight of the control system	95kg
12		Minimum clearance between electronic and electrical equipment and flammable liquid pipelines	3 in
13		…	…

Figure 7. Feature table for the S model of landing gear system with initialized data*

for removing duplication. These results were then written back into the feature table of the S* model for a certain landing gear system, as shown in Figure 7.

EXPERIMENT ANALYSIS

Furthermore, our team used the manually constructed landing gear system "features" as a reference baseline (see Figure 8). We conducted comparative experiments to evaluate the effectiveness of the framework of S* data initialization by employing different levels of hybrid retrieval and re-ranker models in two approaches. The hybrid retrieval is a method that blends different retrieval techniques to enhance the relevance and accuracy of information provided to the LLM (Chandana 2025), and the re-ranker models is a tool that helps make search results more relevant (Nelson 2023).

The first approach involved using the user's raw prompt as input without any prompt engineering techniques (see Figure 6), directly generating S* model data with a LLM. The second approach incorporated prompt template based on the user's selection, followed by generating S* model data using the LLM.

In this experiment, we employed three of the most typical metrics in the LLMs domain — *Recall*, *Precision*, and *F1* (Goutte 2005) — to measure the accuracy level of the "framework of S* data initialization."

$$\mathrm{Recall} = \frac{TP}{TP + FN} \tag{1}$$

The Eq. (1) provides an estimate of *Recall* based on the *TP* and *FN*. *TP* is the number of positive samples which are correctly predicted, and *FN* is number of negative samples which are wrongly predicted.

$$\mathrm{Precision} = \frac{TP}{TP + FP} \tag{2}$$

The Eq. (2) provides an estimate of *Precision* based on the *TP* and *FP*. *FP* is the number of negative samples which are wrongly predicted.

$$F1 = \frac{2 \times \mathrm{Precision} \times \mathrm{Re}call}{\mathrm{Pr}ecision \times \mathrm{Re}call} \tag{3}$$

The Eq. (3) provides an estimate of *F1* based on the result of *Recall* and *Precision*. It is the harmonic mean of accuracy and recall rate. It comprehensively considers both accuracy and recall rate and can more comprehensively evaluate the performance of the model.

The following three tables (Figures 9, 10, 11) present the performance metrics under the combined use of hybrid retrieval and re-ranker models.

Feature	Feature Attribute	Feature Attribute Value	Amount	Chapter	Document ID.
Geometry	Layout	Three-point	8	3 ...	XXXXX
Function	Nose landing gear wheel retraction anti-rotation mechanism	Capable of	12	4 ...	
Non-function	Maximum wet weight	5860kg	22	5	

Figure 8. Evaluation of the completeness of the landing gear system S model*

Methods	LLMs	Amount of Feature Attribute Value	Recall	Precision	F1
Hybrid Retrieval = True Re-ranker = True					
LLM	Deepseek-R1-70b Qwq-32b	9	0.328 0.319	0.355 0.337	0.341 0.328
LLM (with Prompt Template)	Deepseek-R1-70b Qwq-32b	21	0.624 0.611	0.691 0.683	0.655 0.645

Figure 9. No hybrid retrieval and re-ranker model used

Methods	LLMs	Amount of Feature Attribute Value	Recall	Precision	F1
Hybrid Retrieval = True Re-ranker = True					
LLM	Deepseek-R1-70b Qwq-32b	12	0.409 0.398	0.433 0.402	0.421 0.400
LLM (with Prompt Template)	Deepseek-R1-70b Qwq-32b	27	0.702 0.693	0.793 0.764	0.745 0.727

Figure 10. Hybrid retrieval used without re-ranker model

Methods	LLMs	Amount of Feature Attribute Value	Recall	Precision	F1
Hybrid Retrieval = True Re-ranker = True					
LLM	Deepseek-R1-70b Qwq-32b	17	0.535 0.533	0.590 0.581	0.561 0.550
LLM (with Prompt Template)	Deepseek-R1-70b Qwq-32b	31	0.741 0.739	0.802 0.799	0.770 0.768

Figure 11. Hybrid retrieval used with re-ranker model

Based on the above experimental results, we can conclude that the “framework of S* data initialization” achieves the best data generation performance when using hybrid retrieval combined with the re-ranker model by using prompt engineering. Additionally, we used this framework to generate S* data for systems such as aircraft itself, IMA system, and power plant system, achieving the same effective results.

Furthermore, our team also attempted to use the “LLM-based S* data initialization framework” to directly locate and trace product data for a specific system within the product. This approach revealed a significant amount of data redundancy and inconsistency, addressing the issue pointed out by Bailey regarding the lack of emphasis on data consistency management. This improvement further enhances the likelihood of successful implementation of systems engineering within the company.

DISCUSSION

Although the “LLM-based S* data initialization framework” demonstrated a high level of performance during the experimental process, some issues remain, such as:

- The integrity of data initialization is still lacking. After replacing it with other

LLMs in the later stages, the completeness of the initialized data showed varying performance. So far, an open-source LLM capable of fully initializing all data has not been found. Additionally, a standardized prompt template for this field has yet to be established, which has also impacted the results to some extent.

- There is a lack of understanding regarding the same content expressed in different forms. The LLMs used in this study have not undergone any "fine-tuning," and their ability to comprehend domain-specific terminology in the civil aviation manufacturing industry remains at a relatively low level.
- The deployment of such systems in enterprises has certain thresholds. Currently, there are no off-the-shelf market products, and the entire deployment process requires high levels of customization. This imposes specific hardware requirements on the deploying enterprise. Additionally, the deployment team must possess a deep understanding of the enterprise's product development system (such as operational processes, regulatory requirements, etc.), while also being familiar with LLM platform development, natural language processing (NLP) technology, and the PBSE methodology.

CONCLUSION

Overall, the complex product development model constructed by our team under the system engineering framework through the "LLM-based S* data initialization framework" has, to some extent, addressed fundamental issues such as the complexity of constructing S* models. It has reduced the barriers to entry and improved efficiency by leveraging PBSE data reuse. This framework actively contributes to increasing the success rate of enterprises in solving problems under different system engineering approaches. Additionally, it has partially solved the problem raised by scholars such as Pfrommer, regarding the difficulty of capturing system environment requirements or assumptions during the development phase (Pfrommer 2022). The "LLM-based S* data initialization framework" adapts to its environment autonomously.

In the future, it may be necessary to fine-tune different LLMs, use various knowledge retrieval methods, and even build LLMs specifically tailored for the PBSE different domains, such as automobiles, large ships, nuclear power plants, aerospace, etc. Once successfully trained, there will be an opportunity to integrate knowledge and experience from different fields, further enriching and expanding the "coupling" relationships behind the data based on the existing S* meta-model. This would enable the generation of higher-quality, more stable outputs that meet specific requirements for documents, checklists, models, and patterns. Furthermore, as the multimodal capabilities of LLMs are significantly improved, it will be possible to explore more scenarios. Similarly, manufacturing equipment control will be a major challenge for AI4SE as it further integrates the entire product life-cycle development process. We believe that AI4SE is a transitional phase from PBSE to IBSE, and our team's next step will be to focus on the research and application of AI agents for the full life cycle of commercial aircraft systems engineering. We are convinced that in the future, by continuously integrating AI technology and developing systems engineering AI agents that cover the entire life cycle in various domains (society, urban development, education, food, healthcare, finance, industry, etc.), a significant advancement in human civilization as a whole will be achieved. ■

REFERENCES

- Abiodun, O. I., A. B. Jantan, A. E. Omolara, K. V. Dada, A. Umar, O. U. Linus, H. Arshad, A A. Kazaure, U. M. Gana, and M. U. Kiru. 2019. "Comprehensive Review of Artificial Neural Network Applications to Pattern Recognition." IEEE Access 7:158820-158846.
- Bai, J., S. Bai, Y. Chu, Z. Cui, K. Dang, X. Deng, Y. Fan, W. Ge, Y. Han, F. Huang, B. Hui, L. Ji, M. Li, J. Lin, R. Lin, D. Liu, G. Liu, C. Lu, K. Lu, J. Ma, R. Men, X. Ren, C. Tan, S. Tan, J. Tu, P. Wang, S. Wang, W. Wang, S. Wu, B. Xu, J. Xu, A. Yang, H. Yang, J. Yang, S. Yang, Y. Yao, B. Yu, Y. Bowen, H. Yuan, Z. Yuan, J. Zhang, X. Zhang, Y. Zhang, Z. Zhang, C. Zhou, J. Zhou, X. Zhou, and T. Zhu. 2023. Qwen Technical Report. ArXiv, abs/2309.16609.
- Bailey, I. D. 2011. Enterprise Ontologies – Better Models of Business.
- Brian, W. 2022. Federal Aviation Administration (FAA) Next Generation Air Transportation System, SEBoK.
- Buschmann, F., R. Meunier, H. Rohnert, P. Sommerlad, and M. Stal. 1996. Pattern-Oriented Software Architecture Volume 1: A System of Patterns.
- Chandana S. M, G. Gizem, and G. Fosca. 2025. "Hybrid Retrieval for Hallucination Mitigation in Large Language Models: A Comparative Analysis." https://arxiv.org/pdf/2504.05324.
- Cook, D., and W. D. Schindel. 2015. "Utilizing MBSE Patterns to Accelerate System Verification." INCOSE International Symposium, 25.
- Cummings, P., N. Schurr, A. Naber, Charlie, and D. Serfaty. 2021. "Recognizing Artificial Intelligence: The Key to Unlocking Human AI Teams." In *Systems Engineering and Artificial Intelligence*, edited by W. F. Lawless, R. Mittu, D. A. Sofge, T. Shortell, and T. A. McDermott, 23-45: Springer.
- Daya G., Dejian Y., Haowei Z., Junxiao S., Ruoyu Z., Runxin X., Qihao Z., Shirong M., Peiyi W., Xiao B., Xiaokang Z., Xingkai Y., Yu W., Z.F. W., Zhibin G., Zhihong S., Zhuoshu L., and Ziyi G. 2025. "DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning." https://arxiv.org/pdf/2501.12948.
- Dove, R., and B. Schindel. 2017. "Case Study: Agile SE Process for Centralized SoS Sustainment at Northrop Grumman." INCOSE International Symposium, 27.
- Dove, R., W. Schindel, and K. Garlington. 2018. "Case Study: Agile Systems Engineering at Lockheed Martin Aeronautics Integrated Fighter Group." INCOSE International Symposium, 28.
- Dove, R., W. Schindel, and R. W. Hartney. 2017. "Case study: Agile hardware/firmware/software product line engineering at Rockwell Collins." Annual IEEE International Systems Conference (SysCon), 1-8.
- Galli, B. J. 2020. "The Application of Systems Engineering to Project Management: A Review of Their Relationship." Int. J. Syst. Dyn. Appl. 9: 81-106.
- Goutte, C., and E. Gaussier. 2005. "A Probabilistic Interpretation of Precision, Recall and F-Score, with Implication for Evaluation." Lecture Notes in Computer Science. 3408: 345-359.

- He, D., Shen, B., Qi, X., and Qian, Z. 2023. COMAC Systems Engineering Manual (Seventh Edition). Shanghai, CN: Shanghai Jiao Tong University Press.
- Hohpe, G., and B. Woolf. 2004. *Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions*. Addison-Wesley Professional.
- INCOSE. 2018. Model-Based SE Transformation, Aerospace Corporation System Engineering Forum.
- Jules, W., Quchen F., Sam H., Michael S., Carlos O., Henry G., Ashraf E., Jesse S., and Douglas C. 2023. "A Prompt Pattern Catalog to Enhance Prompt Engineering with ChatGPT." In Proceedings of the 30th Conference on Pattern Languages of Programs (PLoP '23). The Hillside Group, US, Article 5, 1-31.
- Lu, Y. 2008. "Engineering Practice of engineering construction and management—Taking the Three Gorges Project as an Example." *Strategic Study of CAE*, 10 (12): 7.
- McDermott, T., D. DeLaurentis, P. A. Beling, M. R. Blackburn, and M. A. Bone. 2020. "AI4SE and SE4AI: A Research Roadmap." *INSIGHT* 23 (1): 8-14.
- McDermott, T. A., M. R. Blackburn, and P. A. Beling. 2021. "Artificial Intelligence and Future of Systems Engineering." In *Systems Engineering and Artificial Intelligence*, edited by W. F. Lawless, R. Mittu, D. A. Sofge, T. Shortell, and T. A. McDermott, 47-59: Springer.
- Nelson F. L., L. Kevin, H. John, P. Ashwin, Michele B., Fabio P., and Percy L. (2023). "Lost in the Middle:
- How Language Models Use Long Contexts." https://arxiv.org/abs/2307.03172.
- Pfrommer, J., T. Usländer, and J. Beyerer. 2022. KI-Engineering – AI Systems Engineering. at Automatisierungstechnik, 70, 756–766.
- Schindel, B. 2007. "What is the Smallest Model of a System?"
- Schindel, W. D., S. A. Lewis, and J. J. Sherey. 2015. "Accelerating MBSE Impacts Across the Enterprise:
- Model-Based S* Patterns." INCOSE International Symposium 25 (1): 1159-1176.
- Schindel, W. D. 2011. "1.4.2 What Is the Smallest Model of a System?" INCOSE International Symposium, 21.
- Schindel, W. D. 2015. "System Life Cycle Trajectories: Tracking Innovation Paths Using System DNA." INCOSE International Symposium, 25.
- Schindel, W. D., and V. R. Smith. 2002. Results of Applying a Families-of-Systems Approach to Systems Engineering of Product Line Families.
- Schindel1, W., S. Lewis1, J. Sherey1, and S. Sanyal2. 2015. "Accelerating MBSE Impacts Across the Enterprise: Model-Based S*Patterns." INCOSE International Symposium, 25.
- Sherey, J. J. 2006. "Capitalizing on Systems Engineering." INCOSE International Symposium 16 (1): 762-774.
- Stockman, B. P., J. Boyle, and J. B. Bacon, J.B. 2010. International Space Station Systems Engineering. Case Study.
- Tolk, A., and C. Lakhmi. 2011. Intelligence-Based Systems Engineering.
- Topsakal, O., and T. C. Akinci. 2023. "Creating Large Language Model Applications Utilizing LangChain: A Primer on Developing LLM Apps Fast." International Conference on Applied Engineering and Natural Sciences.
- Vaswani, A., N. M. Shazeer, N. Parmar, N., J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, L., and I. Polosukhin. 2017. "Attention is All you Need." Neural Information Processing Systems.
- Zielske, M., and T. Held. 2022. "Agile Methods Used by Traditional Logistics Companies and Logistics Start-Ups: A Systematic Literature Review." J. Syst. Softw., 190: 111328.
- Zirnstein, B. 2023. "Extended Context for InstructGPT with LlamaIndex." ResearchGate. https://doi.org/10.13140/RG.2.2.17701.31204.

ABOUT THE AUTHORS

Degang Liang. Graduated from the University of Southern California, US, serves as the head of systems engineering research and project application at COMAC, senior engineer, responsible for the *COMAC Systems Engineering Manual* and the training of systems engineers within the company. Long-term engagement in commercial aircraft products design, technical management, and project management, participating in the development of multiple COMAC commercial aircraft series. Specializes in systems engineering management, agile systems engineering, MBSE, and AI4SE, etc.

Baoyu Dong. Graduated from Nanjing University of Aeronautics and Astronautics, China, engaged in research and project application at COMAC, long-term commitment to improving the company's PLM platform architecture, particularly in data cleaning, analysis, and governance in the field of large language models. Proficient in deployment of AI platforms and serves as the expert in the company's AI4SE domain.

Solving the Octopus Problem in Digital Engineering – Towards Reusable Asset Specification 3.0

Matthew Hause, mhause@systemxi.com; **Sriram Krishnan,** skrishnan@istaridigital.com; **Mark Petrotta,** mpetrotta@systemxi.com; **Michael Shearin,** Michael.Shearin@gtri.gatech.edu; and **Tomas Vileiniskis,** tomas.vileiniskis@3ds.com

ABSTRACT
In the animal kingdom, survival depends on knowledge transfer. But the solitary octopus is an exception. Brilliant and adaptable, each octopus dies with its hard-won knowledge. Engineering projects often face the same plight: insights remain locked in individual models and documents, never reaching others who could benefit subsequently. As a result, teams repeatedly reinvent the wheel–recreating models, wasting time, and compounding technical debt. To break this cycle, digital engineering needs curated, discoverable assets. This article outlines recent efforts to update the OMG Reusable Asset Specification (RAS 3.0) to enable better discovery through structured metadata, searchable asset catalogs, and curation services, and accelerating reuse, collaboration, and scalability.

KEYWORDS: MBSE, assets, reuse, curation, libraries, knowledge

1. OVERVIEW

Model-based systems engineering (MBSE) projects regularly generate reusable knowledge—such as algorithms, patterns, components, and interfaces—and new standards like SysML v2 (OMG 2025a) are shifting toward library-centric reuse to support that. Yet much of this knowledge remains siloed, like the octopus in the animal kingdom—intelligent and adaptive, but unable to pass along what it learns. In digital engineering, repositories often lack metadata, curation, and search, preventing reuse and forcing engineers to start from scratch.

While there are valuable examples, such as the NIST 1,200-control (NIST 2020) security library created for the Unified Architecture Framework (UAF), (OMG 2025b) by MITRE, most organizations still lack the infrastructure or awareness needed to share, evolve, and safely reference reusable assets. Prior studies (Reymondet et al. 2016, Rhodes 2019, Wu 2021) highlight common reuse barriers – limited access, trust gaps, and low reuse maturity – that fragment knowledge and lead to repeated effort across programs.

To address this, the OMG is updating the Reusable Asset Specification (RAS 3.0) (OMG 2020, Hause 2014), to define asset cards, catalogs, and supporting APIs that enable discoverability, curation, and reuse across organizational and technical boundaries. Asset metadata catalogs, libraries, and associated APIs will also support related initiatives such as model-based acquisition (MBAcq), product line engineering, and other emerging digital engineering needs. This article explores the evolving RAS 3.0 specification.

The rest of the paper is organized as follows: Section 2 presents use cases related to discoverability that help inform reusable asset metadata. Section 3 describes the core data model and building blocks of discoverability. Section 4 introduces relevant specifications and format options informing RAS 3.0. Section 5 outlines the proposed RAS 3.0 API services. Section 6 concludes with next steps and a call for community feedback.

2. MOTIVATION: USE CASES DRIVING REUSE AND DISCOVERABILITY

While digital engineering teams are producing more reusable assets, they often lack a shared way to describe, discover, or exchange them across tools or organizations. As adoption grows, so do the challenges – fragmented metadata, inconsistent governance, and siloed repositories all make reuse difficult.

Table 1. Core and specialized extensible meta-data options for asset cards in RAS 3.0

Core Metadata (Required for all assets)	Extension Metadata (Optional: domain, tool, or enterprise-specific)
Identity & Intent • Artifact ID • Name/Title • Description/Purpose	Classification & Navigation • Topics/Keywords • Categories • Tags/Labels
Stewardship & Access • Maintainer/Contact/ Organization • License type • Repository/Access URI	Digital Engineering Environment context: • Tool-type standard Taxonomy alignment • Inter-tool data exchange schema
Version & Lifecycle • Version • Lifecycle status (e.g., draft, released, deprecated)	Organizational & Semantic Context • Author(s) • Organization/Source • Expression format (e.g., SysML, JSON) • Asset type (e.g., model, code) • Modeling approach • Ontologies/Information models
Governance & Provenance • Parent asset ID • Lineage relationship type • Version history • Related assets/Lineage links	Security & Legal Extensions • Certifying authority • Certifications (e.g., VV&A) • NDA or legal constraints • White box/Black box designation • Security classification

To address this, the RAS 3.0 team gathered a diverse set of real-world use cases that grounded the specification in practical reuse and discovery challenges. These helped clarify which metadata fields are essential and where extension mechanisms are needed.

1. **Asset reuse across organizations**
   A product owner must package a model or component for use by a team in another organization. Reuse depends on shared metadata for structure, license, and provenance.
2. **Curation of an internal asset library**
   An internal asset librarian needs to track and maintain digital assets across repositories. They rely on metadata to support search, lifecycle management, and relevance curation.
3. **Building a reference architecture**
   Before designing a new technology stack, a team searches for existing models or reference architectures. Discoverability depends on standardized topics, tags, and usage annotations.
4. **Reusing models for a new simulation study**
   A simulation team requires agent-based models for a novel study involving human–autonomous vehicle interactions. Reuse depends on metadata describing modeling approach, validation, and context.
5. **Managing technical debt while modernizing legacy systems**
   A team updating legacy mechanical systems needs access to both old and new design artifacts. Effective comparison depends on consistent metadata for identity, lineage, and format.

These use cases revealed a common set of core metadata needs — such as identity, description, ownership, repository, and license — alongside domain-specific needs like certifications, ontologies, and access restrictions. Rather than over-standardizing, RAS 3.0 will adopt a core asset card with structured extensibility, supporting consistency where needed while enabling domain variation. This approach promotes reuse at scale without forcing uniformity across tools and workflows. See Table 1 for example metadata options under review, which will continue to be refined with feedback.

3. BUILDING BLOCKS OF DISCOVERABILITY: MODELS, CATALOGS, AND REPOSITORIES

Effective asset discoverability requires a two-level data model: a conceptual layer that defines what an asset is and how it relates to catalogs, repositories, curators, taxonomies, and lifecycle phases; and a physical layer that governs how this information is stored, accessed, and exchanged. In RAS 3.0, taxonomy and lifecycle states are treated as first-class concepts – essential for enabling faceted search, maturity tracking, and provenance queries.

Many organizations today blur the line between asset and catalog metadata, apply inconsistent semantics, and rely on non-aligned APIs. These issues make search unreliable, cross-repository discovery difficult, and governance unclear. To address this, RAS 3.0 introduces a clear separation of components, as illustrated in Figure 1:

- Digital asset: A reusable model, pattern, or component.
- Asset card: The asset's self-describing metadata, including identity, purpose, status, owner, relationships, license/access, tags, and versioning, with optional domain-specific extensions.

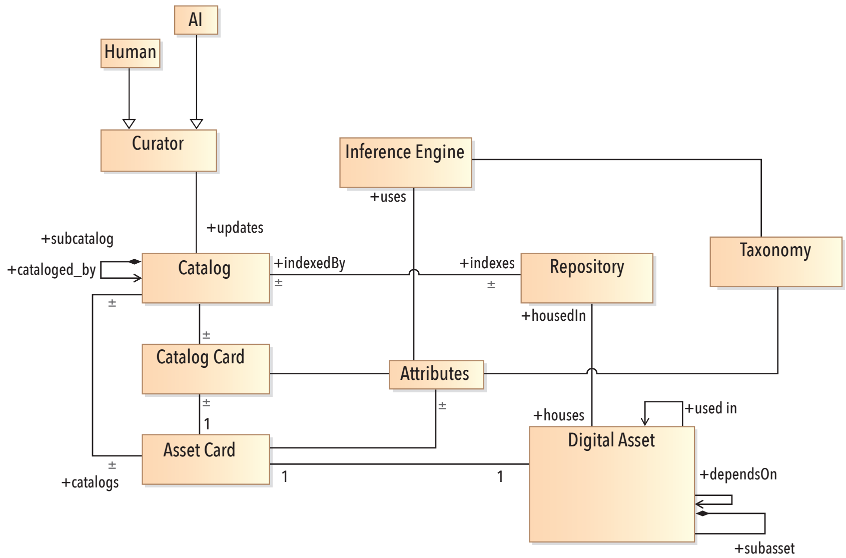


Figure 1. RAS conceptual data model linking assets, asset cards, catalogs, and repositories

- Catalog: A curated index, public or private, that enables faceted search. Assets can appear in multiple catalogs.
- Catalog card: Supplemental metadata specific to a catalog (e.g., usage notes, domain tags), layered on top of the asset card without altering it.
- Repository: The version-controlled storage system where assets reside (e.g., Teamwork Cloud, Git, or PLM platforms).

This separation of concerns is central to RAS 3.0: the asset card defines authoritative metadata about the asset itself, while the catalog card allows additional, context-specific metadata without altering the original asset. This enables catalogs to reflect different viewpoints or organizational needs while maintaining a consistent core definition. To support discoverability, governance, and API consistency, RAS 3.0 aligns its service interfaces 1:1 with the conceptual model.

For physical realization, RAS 3.0 seeks to implement the conceptual model via standard APIs for discovery and access, and (where needed, e.g., git-based stores) standard file formats primarily JSON with JSON Schema to support validation, extensibility, and machine-processability, and XML where required by legacy. This ensures that physical asset metadata remains portable, predictable, and interoperable across tools, repositories, and domains.

4. DISCUSSION ON REUSABLE ASSET SPECIFICATION 3.0

The Reusable Asset Specification (RAS) was first published in 2005 (v2.2) to standardize software reuse through consistent packaging—focusing on granularity, visibility, and completeness. Today, digital engineering calls for a more modern approach: one that supports asset discoverability, curation, and reuse across teams, tools, and organizations.

A key goal for the RAS 3.0 team is to simplify the specification and avoid creating new metadata where proven standards already exist. We're reviewing vocabularies like DPROD and MSC-DMS to identify which elements can be reused or adapted. This includes comparing their purpose, design principles and metadata structures to those needed for RAS. We're also evaluating related OMG initiatives that use metadata cards to find further alignment.

To balance structure with flexibility, RAS 3.0 defines an asset card with a small, required core–fields like identity, purpose, owner or maintainer, version history, license, repository location, and status (similar to the structure and options outlined in Table 1, balancing consistency across reuse contexts with flexibility for domain-specific needs). Additional metadata can be added through an extensible layer, tailored to specific domains or processes (e.g., certifications, ontologies, or security requirements).

Serialization format discussions are ongoing. The leading option is JSON with JSON Schema, offering validation, extensibility, and compatibility with modern APIs. XML remains in use where required, while RDF/OWL may support richer semantic links. Markdown templates are also being explored for more readable authoring, with automated conversion to structured formats.

We're also considering how to balance tool-specific and tool-independent formats. JSON supports lightweight, validated data exchange in API-driven environments, while formats like XMI may still be needed in tool-native workflows. Any format must support hierarchy, linking, and extensions. Readability matters too—especially when humans curate or author metadata—so more accessible formats like Markdown may play a role, as long as they remain machine-convertible.

5. API SERVICES – ENABLING ACCESS AND SEARCH INTEGRATION IN RAS 3.0

RAS 3.0 requires a standard, repository-agnostic services layer to support search, discovery, navigation, and curation. To inform its design, we follow a similar data model + services layer paradigm introduced in the recent OMG SysML v2 standard, and its Systems Modeling API and Services specification.

We propose a RAS 3.0 API specification that defines a platform-independent (PIM) service set mapped 1:1 to the RAS data model, with REST/HTTP as the exemplar platform-specific model (PSM). Payloads are kept consistent and can be validated through an auto-generated, versioned JSON Schema (See Figure 2).

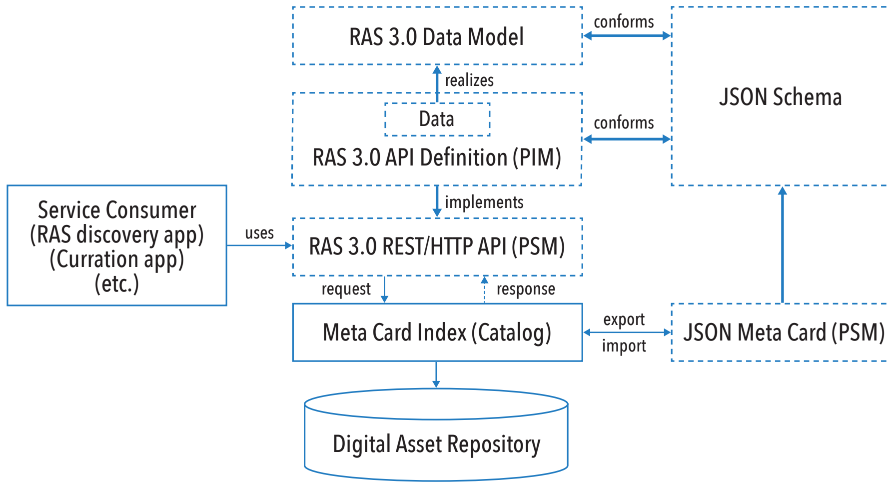


Figure 2. RAS 3.0 API and services architecture

Core capabilities include faceted search and query, retrieval of asset cards and catalog metadata, submission and update of assets, version and lineage tracking, bulk import/export, and support for paging and sorting—all while preserving custom or domain-specific fields end-to-end.

Governance, security, and portability are treated as first-class concerns: APIs and schemas are versioned, a conformance test suite is defined, and the architecture supports authentication/authorization, policy enforcement, audit logging, and air-gapped JSON exchanges validated against schema.

6. FUTURE WORK AND NEXT STEPS

This article outlines the direction for RAS 3.0: a simplified, extensible specification for reusable digital assets that supports discovery, curation, and cross-organizational reuse. The approach builds on real-world use cases, simplified formats, established metadata vocabularies, and lessons from SysML v2 to define asset cards, catalogs, APIs, and supporting services.

As the specification progresses, we actively welcome input from tool vendors, standards contributors, and digital engineering users beginning in Q3 2025. We are particularly interested in feedback on metadata structure, extensibility, schema alignment, and implementation priorities. Organizations interested in piloting or adopting RAS 3.0 are encouraged to submit letters of intent and engage with the OMG RAS working group. ■

REFERENCES

- Hause, M. 2014. “SOS for SoS: A New Paradigm for System of Systems Modeling.” Paper presented at the IEEE, AIAA Aerospace Conference, Big Sky, US-MT, 1-8 March.
- NIST. 2020, NIST Publishes SP 800-53, Revision 5 September 23, 2020, Security and Privacy Controls for Information Systems and Organizations.
- OMG. 2005, Reusable Asset Specification (RAS), Version 2.2, online, available from: http://www.omg.org/spec/RAS/2.2/PDF, formal/05-11-02.
- Object Management Group (OMG). 2025a. OMG2023-03-01. OMG Systems Modeling Language (OMG SysML™), V2.0. http://www.omg.org/spec/SysML/2.0/PDF/. (Link will be updated as SysML v2 progresses through finalization.)
- Object Management Group (OMG). 2025b, The Unified Architecture Framework (UAF) version 1.5, expected date of publication, March 2025.
- Reymondet, L., D. H. Rhodes, and A. M. Ross. 2016 (April). “Considerations for Model Curation in Model-Centric Systems Engineering.” 10th Annual IEEE Systems Conference, Orlando, US-FL.
- Rhodes, D. 2019. “Model Curation: Requisite Leadership and Practice in Digital Engineering Enterprises.” 17th Annual Conference on Systems Engineering Research (CSER).
- Rhodes, D. H., and A. M. Ross. 2015. Interactive Model-Centric Systems Engineering Pathfinder Workshop Report. MIT. Cambridge, US-MA.
- Wu Q., D. Gouyon, and E. Levrat. 2021. “Maturity Assessment of Systems Engineering Reusable Assets to Facilitate MBSE Adoption.” 17th IFAC Symposium on Information Control Problems in Manufacturing, INCOM 2021, Budapest (virtual), HU: June, ffhal-03253865f.

ABOUT THE AUTHORS

Matthew Hause is a distinguished engineer at SSI, chair of the OMG UAF and Reusable Asset Specification (RAS) groups, and a member of the OMG SysML team. With over 45 years of experience in complex systems development, he is an active contributor to standards, training, and mentoring in model-based systems engineering.

Dr. Sriram Krishnan is vice president of strategy at Istari Digital and co-chair of the OMG RAS working group. He leads intellectual property strategy, helps shape digital engineering standards, and has published research on digital ecosystems and deeptech transformation.

Mark Petrotta is a principal engineer at SSI and teaches systems engineering courses at the University of Michigan, Ann Arbor, and Georgia Tech Professional Education. He has 30+ years’ experience with software engineering, systems engineering, and digital transformation.

Mike Shearin is a senior research engineer at Georgia Tech Research Institute, leading the DE Tools and Environments Branch. He has 13 years of MBSE experience and 8 years in military aircraft software. He co-chairs the OMG Model-Based Acquisition Group and DE Environment Working Group.

Tomas Vileiniškis is senior manager at Dassault Systèmes and product manager for Teamwork Cloud. He brings 15 years of experience in software engineering and Semantic Web technologies, and plays a key role in shaping OMG’s Systems Modeling API and RAS 3.0 standards.

The authors are participants of the OMG’s Reasonable Asset Specification working group within the model based acquisition managed community (https://communities.omg.org/wg/mbacq-ras/workgroup).

The Decision Analysis Data Model

Gregory S. Parnell, gparnell@uark.edu; **C. Robert Kenley,** kenley@purdue.edu; **Devon Clark,** devclark@deloitte.com; **Jared Smith,** jarsmith@deloitte.com; **Frank Salvatore,** frank.salvatore@saic.com; **Chiemeke Nwobodo,** cnwobodo@deloitte.com; and **Sheena Davis,** sheedavis@deloitte.com

Note to the Readers | This article is a summary of the following paper: Parnell, G. S., C. R. Kenley, D. Clark, J. Smith, F. Salvatore, N. Chiemeke, S. Davis. 2025. "Decision Analysis Data Model for Digital Engineering Decision Management." Artificial Intelligence and Digital Systems Engineering Special Issue, *Systems* 13 (7): 596. https://doi.org/10.3390/systems13070596 (DADM).

ABSTRACT
Decision management is the systems engineering life cycle process for making program/system decisions. The purpose of the decision management process is: "… to provide a structured, analytical framework for objectively identifying, characterizing, and evaluating a set of alternatives for a decision at any point in the life cycle and select the most beneficial course of action. Systems engineers and systems analysts need to inform decisions in a digital engineering environment. This paper describes a decision analysis data model (DADM) developed in model-based systems engineering software to provide the process, methods, models, and data to support decision management. DADM can support digital engineering for waterfall, spiral, and agile development processes. This paper describes the decision management processes and provides the definition of the data elements. DADM is based on ISO/IEC/IEEE 15288, the INCOSE *Systems Engineering Handbook*, the SE Body of Knowledge, the Data Management Body of Knowledge, systems engineering textbooks, and journal articles. The DADM was developed to establish a decision management process and data definitions that organizations and programs can tailor for their system life cycles and processes. The DADM can also be used to assess organizational processes and decision quality.

KEYWORDS: digital engineering; decision management; data modeling; logical data modeling; model-based systems engineering; patterns; decision analysis

INTRODUCTION

To address the problem of the lack of a widely available and reusable model-based decision support system for systems engineers, the INCOSE Decision Analysis Working Group developed a decision analysis data model (DADM) in model-based systems engineering software that provides decision management guidance to support multi-factored decisions, such as design comparisons or trade studies, while leveraging a model-based environment to improve how those decisions are analyzed and communicated. This data model was developed using the decision management methodology defined in the INCOSE *Systems Engineering Handbook* (Walden et al. 2023), and defines the steps in the decision management process and identifies the data exchanged between those steps. The model can accelerate trade-off analyses, increase consistency, and support the documentation of decision outcomes in a digital model, enabling collaboration in a digital ecosystem in all life cycle stages.

The importance of this work is amplified by the INCOSE *Systems Engineering Vision 2035* (2021), which outlines several challenges that must be realized to achieve the vision for the future state of systems engineering, including:
(1) Enable trusted collaboration and interactions through a digital ecosystem
(2) Provide analytical frameworks for managing the lifecycle of complex systems
(3) Widely adopt reuse practices.

Review of Research on Decision Analysis Data in Systems Engineering

The need for systems engineers to obtain a comprehensive set of data to support system engineering decision making has been documented and reinforced across many decades.

- MIL-STD 499B (Department of Defense 1994, 31), published in 1994, describes a requirement for a decision data base, which is "a repository of information used and generated by the systems engineering process, at the appropriate level of detail …. The intent of the decision data base is that, when properly structured, it provides access to the technical information, decisions, and rationale that describe the current state of system development and its evolution."

- The NASA Systems Engineering Handbook was initially published as NASA/SP-6105 in 1995. The 2007 revision describes a need to obtain a comprehensive set of data to support decision making while the initial 1995 publication does not. The most recent version of the NASA handbook ("6.8 Decision Analysis" 2019), published in 2019, states, "Once the technical team recommends an alternative to a NASA decision-maker (e.g., a NASA board, forum, or panel), all decision analysis information should be documented. The team should produce a report to document all major recommendations to serve as a backup to any presentation materials used…. The important characteristic of the report is the content, which fully documents the decision needed, assessments done, recommendations, and decision finally made." In addition to prescribing the need to document decision analysis information, the NASA Systems Engineering Handbook prescribes that the process must be risk-informed, which may include both qualitative and quantitative techniques.
- The INCOSE *Systems Engineering Handbook* was initially published in 1997. Version 4 of the Handbook, published in 2015, was the first version to describe the need to obtain a comprehensive set of data to support decision making. The INCOSE *Systems Engineering Handbook* (Walden et al. 2023, 81), published in 2023, states "Decisions should be documented using digital engineering artifacts. Reports that include the analysis, decisions, and rationale are important for historical traceability and future decisions. The INCOSE *Systems Engineering Handbook* prescribes that the process must identify uncertainties and conduct probabilistic analysis.

Review of Research on Architectural Patterns, Reference Models, and Reference Architectures for Engineering Decisions

Bass, Clements, and Kazman (Bass et al. 2003, sec. 2.3) provide a framework for developing software that can be applied to develop a decision analysis data model to capture digital engineering artifacts that document decisions. They define three architectural structures:

(1) An architectural pattern is a description of element and relation types together with a set of constraints on how they may be used. A pattern can be thought of as a set of constraints on an architecture—on the element types and their patterns of interaction—and these constraints define a set or family of architectures that satisfy them. One of the most useful aspects of patterns is that they exhibit known quality attributes.
(2) A reference model is a division of functionality together with data flow between the pieces. A reference model is a standard decomposition of a known problem into parts that cooperatively solve the problem…. reference models are a characteristic of mature domains.
(3) A reference architecture is a reference model mapped into software elements (that cooperatively implement the functionality defined in the reference model) and the data flows between them. Whereas a reference model divides the functionality, a reference architecture is the mapping of that functionality onto a system decomposition.

There are many examples of architectural patterns, reference models, and reference architectures that have contributed to the advancement of digital engineering. See the DADM paper for a detailed review of this literature.

Systems Modeling Language (SysML) is a general-purpose modeling language for model-based systems engineering (MBSE) ("OMG Systems Modeling Language (OMG SysML) Version 1.7" 2024). It provides the capability to represent a system using the three UML diagram types and adding a fourth diagram type (Buede and Miller 2016):

(1) Structure, which includes Class – renamed to be Block, Package, and Parametric (new) diagrams and eliminates Component, Composite Structure, Deployment, and Object diagrams;
(2) Behavior, which includes Activity (modified), State Machine, and Use Case diagrams;
(3) Interaction, which includes Sequence diagrams and eliminate Collaboration – Communication, Interaction Overview, and Timing diagrams; and
(4) Requirements (new), which includes the Requirement (new) diagram.

Main Aim of DADM and Principal Conclusions

The main aim of DADM was to develop a model that:

(1) enables trusted collaboration and interactions through a digital ecosystem,
(2) be deployable and configurable for multiple decision domains
(3) provides an analytical framework for decision making across the lifecycle stages of complex systems,
(4) will be widely adopted,
(5) enables both traceability and reuse of analysis, decisions, and rationale for decisions,
(6) incorporates guidelines for identifying uncertainties and conducting probabilistic analysis and for documenting the rationale and results,
(7) provides information models built using composable knowledge and process models that emphasize learning in the presence of uncertainty, and
(8) can be tailored to agile development for DEVOPS.

To meet these aims, the DADM was developed using composable SysML activity diagrams for process modeling and block definition diagrams for information modeling. The Magic System of Systems Architect was selected to implement the model, as it is widely designed for trusted digital collaboration, allows for traceability and reuse, allows capturing of guidelines and documentation, and can be tailored.

METHODS

The methods used in developing DADM follow data modeling by the International Data Management Association (DAMA) and were informed by the INCOSE agile systems engineering life cycle model (ASELCM). Both DAMA and ASELCM shaped the approach described below.

Data Modeling

According to DAMA, data management is a wide-ranging set or activities, which includes the abilities to make consistent decisions about how to get strategic value using data (Data Management Association 2017). These activities are organized into the data nanagement framework, which identifies ten (10) categories of data management activities, which interact with an organization's data governance to inform the data, information, and content lifecycle. These ten (10) data management activities begin with the definition of data architecture and design using data models. "Data modeling is the process of discovering, analyzing, and scoping data requirements, and then representing and communicating these data requirements in a precise form called *the data model*. This process is iterative and involves conceptual, logical, and physical models" (Data Management Association 2017).

The DADM mapped the inputs and outputs of the decision management processes defined in ISO/IEC/IEEE 15288 (ISO–International Organization for Standardization 2023), the INCOSE *Systems Engineering Handbook* (Walden et al. 2023), and the SEBoK (SEBoK Editorial Board 2024) to identify the high-level 'decision concepts'

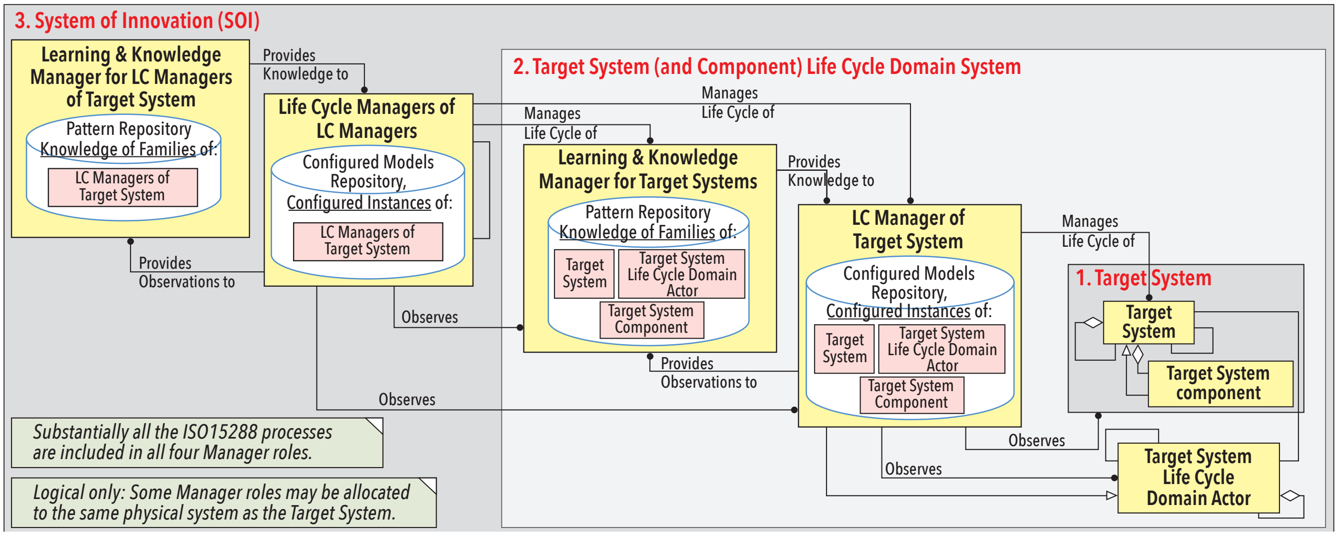


Figure 1. Top-level agile system engineering life cycle model (Schindel and Dove 2016)

needed to execute the high-level decision management process. From there, the conceptual data model was decomposed and traced to an implementation-agnostic logical model, which defines data-driven processes and the data needs connecting them. Finally, implementation-specific physical data models would be tailored from the logical data model on a project-by-project basis.

Conceptual Data Model (CDM)

"A conceptual data model captures the high-level data requirements as a collection of related concepts. It contains only the basic and critical business entities within a given realm and functions." (Data Management Association 2017) In this regard, the conceptual DADM provides an executive-friendly definition of the key concepts that apply to the business needs for their decision making. For example, the concept of a decision should include definitions of the decision itself, including the decision-maker(s), the alternatives being considered, and the values against which those alternatives will be evaluated. A summary of the DADM's conceptual model is provided in Section 3.1.

Logical Data Model (LDM)

A CDM is too high level to be implemented. The CDM defines needs without mapping those needs to data solutions. "A logical data model (LDM) captures the detailed data requirements within the scope of the CDM." (Data Management Association 2017) For example, whereas our CDM defines the decision concept as including a decision-maker, a set of alternatives, and a set of values, our LDM maps those to specific data points, to include a decision authority, courses of action, and objectives. As the LDM was further refined, those data needs also included properties such as data attributes and domains, and the structure of data began to take shape without discussing additional properties of the data structure, such as format and validation rules, which are reserved for the physical data model. A summary of the DADM's logical data model is captured in Section 3.2.

Physical Data Model

"Logical data models require modifications and adaptations in order to have the resulting design perform well within storage applications." (Data Management Association 2017) Therefore, the implementation specifics for any given solutions are defined in the physical data model for that individual system. Note, the purpose of the decision analysis data model is to provide a reference for implementation for systems engineers and developers. Therefore, the DADM does not include a physical data model.

Agile Systems Engineering Life Cycle Model

The INCOSE agile systems engineering life cycle model (ASELCM) has three major systems (Figure 1). The INCOSE ASELCM is a model of learning patterns that describes how systems improve through three interconnected levels: 1) the target system; 2) the lifecycle domain system; and 3) The system of innovation. System 1 of the ASELCM represents an engineered system of intertest (and its components) about which development and operations decisions are being made. Note that many different instances of Systems 1 may be present over time. Examples include aircraft, automobiles, telephones, satellites, software systems, data centers, and health care delivery systems. System 2 is the life cycle domain system, which is the system within which different instantiations of System 1 will exist during their life cycle. This includes any system that directly manages the life cycle of an instance of a target system during its development, production, integration, maintenance, and operations. If System 1 is managed as a program over its life cycle, then Systems 2 can be thought of as a program office responsible for the life cycle stages of System 1 (and the program's associated processes). Different instances of System 2 can occur, e.g., one instance could be a program office responsible for development, production, and integration; and another instance could be a program office responsible for maintenance and operations. System 3 is the system of innovation that includes System 1 and System 2, and that is additionally responsible for managing the life cycles of instances of any System 2. System 3 develops, deploys, and manages System 2 work processes and evaluates them for improvements. System 3 can be thought of as the enterprise innovation system, e.g., an organization that develops, produces, and integrates many diverse kinds of systems, or a user that operates, integrates, and maintains many different systems. The system of innovation produces better lifecycle processes…and better lifecycle processes get implemented to produce better target systems.

Figure 2 shows how DADM can be embedded in the ASELCM to support digital engineering. The System 3 enterprise-level system of innovation can deploy an instance of the DADM logical model to a System 2 program management entity within the enterprise. The System 2 life cycle manager is responsible for using the logical process model to develop a physical model of the DADM artifacts.

Data Model Validation

From a design perspective, data mod-

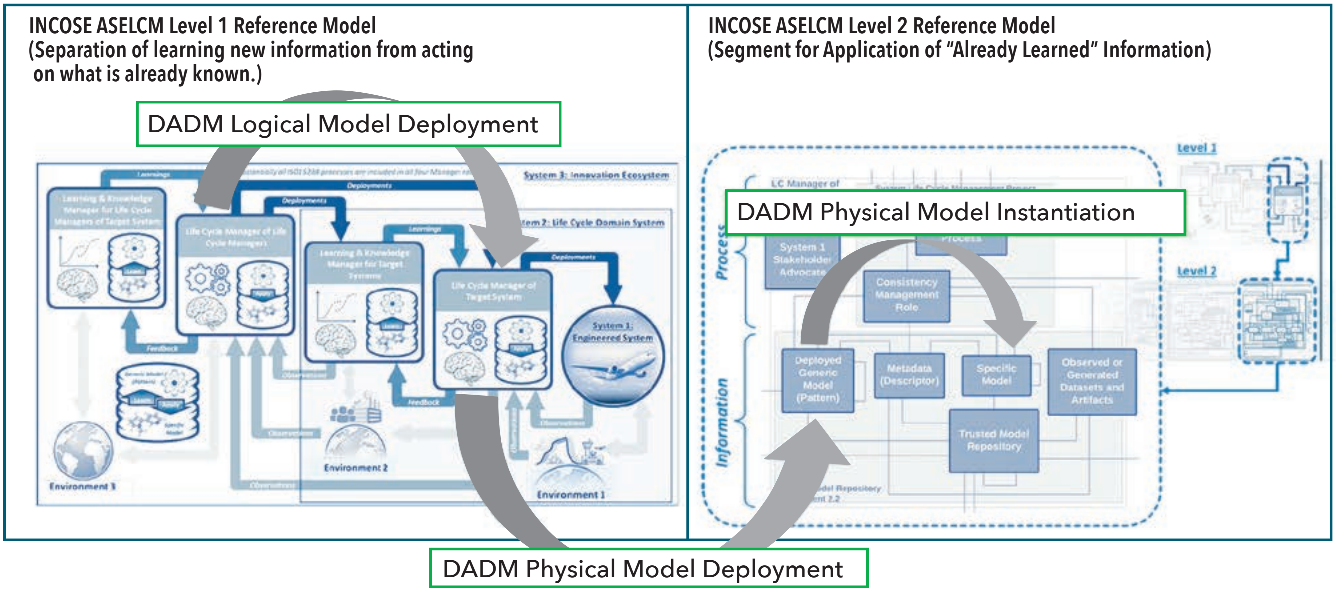
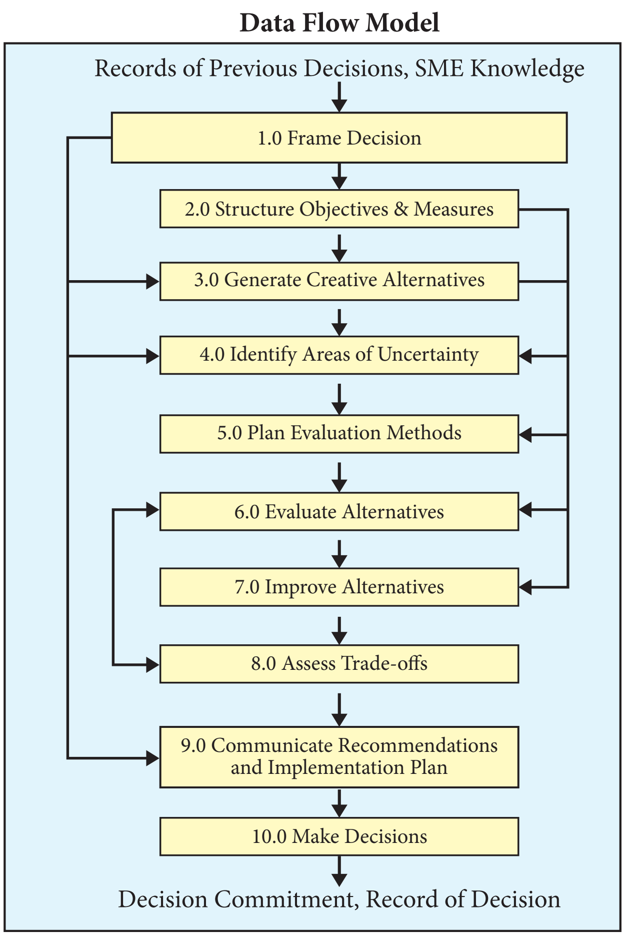


Figure 2. DADM is a deployed generic model (pattern) to be applied for making decisions about an engineered system (Pinon Fischer et al. 2022)

els are validated by mapping inputs and outputs through an organization's business processes, by assuring traceability across the conceptual, logical, and physical models, by justifying the models' activities and data conform to best practices that are documented in the literature, and by independent review of the model by subject matter experts. The business process for the DADM are the decision management processes defined in ISO/IEC/IEEE 15288 (ISO–International Organization for Standardization 2023), the INCOSE *Systems Engineering Handbook* (Walden et al. 2023), and the SEBoK (SEBoK Editorial



Illustrative Data Artifacts

Stakeholders, Stakeholder Need, Decision Context, Scenarios, Use Cases, Vision, Issues, Decision Hierarchy, Influence Diagram, Uncertainty, Decision Frame

Values, Decision Objectives, Value Measures (Performance, Cost, and Schedule), Value Hierarchy, Requirements

Context, Value Hierarchy, Qualitative Value Space, Options, Potential Alternatives

Decision Frame, Value Hierarchy, Scenarios, Use cases, Uncertainties (Stakeholder, Performance, Cost, Schedule, Other, Courses of Action, Previous)

Previous Systems Analysis Plan, Value Hierarchy, Courses of Action, Data Models, Simulations Assessment Flow Diagram

Assessment Flow Diagram, Quantitative Value Model, Alternative Values, Deterministic Analysis, and Probability Analysis

Risk Treatments, Opportunity Treatments, Revised Courses of Action, Reevaluate Alternatives

Value Hierarchy, Courses of Action, Tradespace Analysis (Deterministic, Probabilistic), Trade-offs

Decision Frame, value Hierarchy, Courses of Action, Requirements, Decision Story, Recommendation, Risks, Implementation Plan

Decision Hierarchy, Value Hierarchy, Tradespace, Course of Action, Rationale, Implementation Plan, Decision Record

Figure 3. DADM data flow model and illustrative data artifacts. Some artifacts are updated in subsequent steps.

Table 2. DADM purpose and alternative methods for each DADM conceptual model process step

Step	Purpose	Common Methods
1.0 Frame Decision	Identify the purpose and scope of the decision in the context of the system life cycle.	Problem Definition, Opportunity Definition, Influence Diagram, Stakeholder Analysis, Use Cases, Scenarios, Systems Thinking
2.0 Structure Objectives and Measures	Identify the decision objectives and the measures that will be used to assess achievement of the objectives before we develop the alternatives.	Value Hierarchy, Objectives Hierarchy, Value Tree, Requirements Analysis
3.0 Generate Creative Alternatives	Develop creative alternatives that span the decision space of feasible alternatives including innovative technologies and new production, service, and delivery processes.	Zwicky's Morphological Box, Structured Creativity Techniques, Optimization, Set-Based Design, Pugh Method, TRIZ
4.0 Identify Areas of Uncertainty	Identify uncertainties to understand potential risks and opportunities and allow decision-makers to evaluate choices more accurately.	Risk and Opportunity Analysis, Influence Diagram, Scenario Analysis
5.0 Plan Evaluation Methods	Plan evaluation methods to identify the data, the models, simulations, process flow, resources and time required to perform an alternative assessments that biases and enhances the reliability of the analysis.	Assessment Flow Diagram, Systems Engineering Management Plan
6.0 Evaluate Alternatives	Implement the alternative evaluation plan to assess the performance, value, cost, and schedule of the alternatives.	Deterministic Analysis, Probabilistic Analysis, Portfolio Analysis, Benefit Cost Analysis, Modeling, Simulation, and Analysis, Optimization, Systems Analysis, Risk Analysis, Mission Analysis, Life Cycle Cost Analysis, Scheduling
7.0 Improve Alternatives	Use the evaluation information to improve the alternatives by increasing value and reducing risk.	Value-Focused Thinking, Risk Analysis, Risk Management, Systems Thinking, Systems Analysis, Optimization
8.0 Assess Trade-offs	Assess the value trade-offs including performance, value, cost, and schedule trade-offs.	Tradespace, Trade-off Analysis, Systems Analysis, Pareto Analysis, Cost as an Independent Variable
9.0 Communicate Recommendations and Implementation Plan	Communicate the trade-offs, recommendation(s), and implementation plan to the system decision maker (s) for their decision.	System Decision, Solution Implementation, Implementation Schedule
10. Make the Decision	Make the decision and document the rationale.	Decision Record

Board 2024). Traceability for the DADM is achieved by decomposing the conceptual data model activities and data in a traceable manner to the logical model. Validating that the models' activities and data conform to best practices is done by following relevant standards, bodies of knowledge, systems engineering textbooks, and journal articles in developing the models. Conformance to best practices is demonstrated in this article by the extensive citations in the Section 3 Results and Appendix A. DADM has been reviewed by the INCOSE Impactful Products Committee and approved for release in the INCOSE Systems Engineering Laboratory ("SYSTEM ENGINEERING LABORATORY" 2025).

Validation of the feasibility and effectiveness of the DADM model will be achieved by seeking government and industry users who will deploy the DADM MBSE implementation and provide feedback on its use in their organizations.

RESULTS – THE DECISION ANALYSIS DATA MODEL

This section describes the DADM conceptual data model for each of the ten steps in the decision management process and provides illustrative data artifacts for each step. For each step, we describe the purpose and some of the system engineering methods used to create decision management data. We also provide tailoring and reuse guidance for each step. The logical process model is described by Parnell et al. (2025) and illustrates key concepts for each of the ten steps. For each of the ten steps, they have activity diagrams that provide the activity and information flow to develop the decision data elements and a block definition diagram that shows the relationships between the data elements. In addition, they illustrate the major data artifacts, provide definitions of all the decision data elements, and provide references to the systems engineering and decision analysis literature.

3.1 DADM Conceptual Data Modeling

The foundation of DADM can be summarized via a data flow model, Figure 3, that identifies the sequence conceptual management steps that are executed sequentially in the numerical order shown without skipping any of the steps. The data flow model shows the inputs to the first step, the outputs from the final step, has arrows to indicate the general flow of data between the steps. The figure lists the illustrative data artifacts generated for each step in the process. Each step describes an important activity and includes data artifacts used to inform the decision management team. More detailed discussion of the data and conceptual management steps will be provided in the subsequent sections. The data flow and the data artifacts can be reused with appropriate modifications in subsequent system life cycle stages.

Table 2 provides the name of each of the ten steps, the purpose of each step, and common methods used in other systems engineering decision models and the system engineering literature.

3.2. Logical Data Modeling

For each of the 10 process steps the DADM paper provides a logical process model and a logical data model: block definition diagram. See the paper for the other 9 steps. The DADM paper also provides a list of the definitions of all the terms used in DADM in Appendix A.

Step 1: Frame Decision

Spetzler, Winter, and Meyer define the decision frame as a collection of artifacts that answer the question: "What problem (or opportunity) are we addressing?" That is comprised of three components: (1) our purpose in making the decision; (2) the scope, what will be included and excluded; and (3) our perspective including, our point of view, how we want to approach the decision, what conversations will be needed, and with whom. (Spetzler et al. 2016, 13). Figure 4 uses an activity diagram to depict the logical process for defining a decision frame.

The frame decision process is described

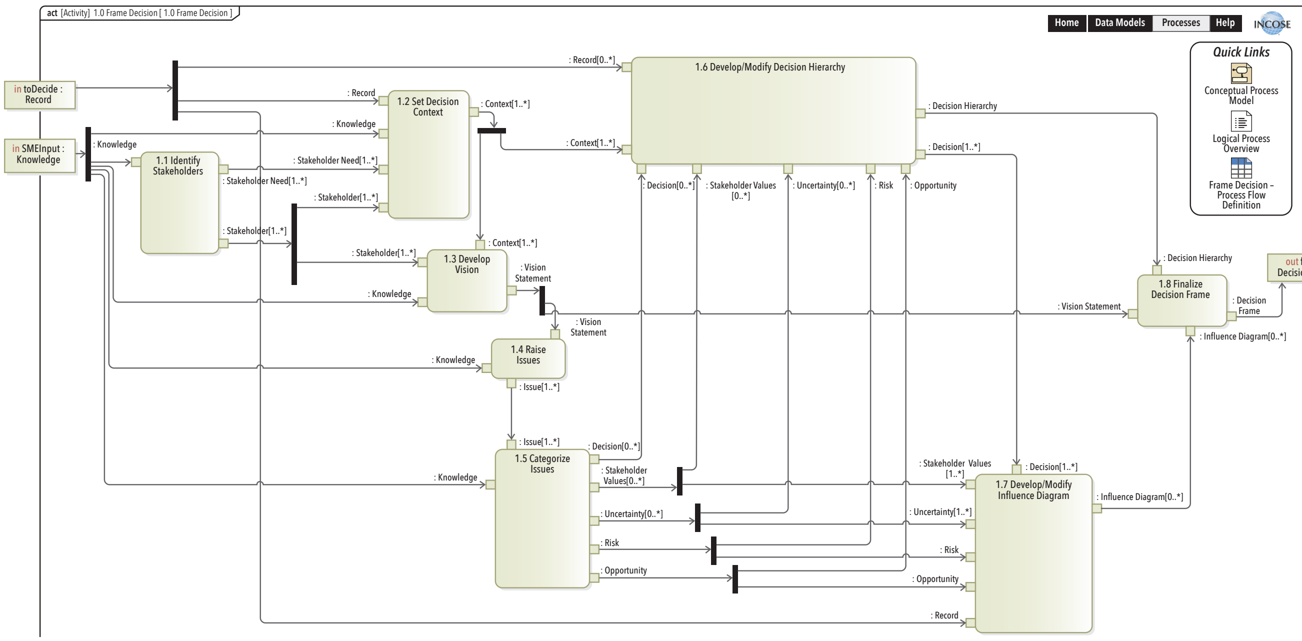


Figure 4. *Logical process model: activity diagram for step 1, frame decision*

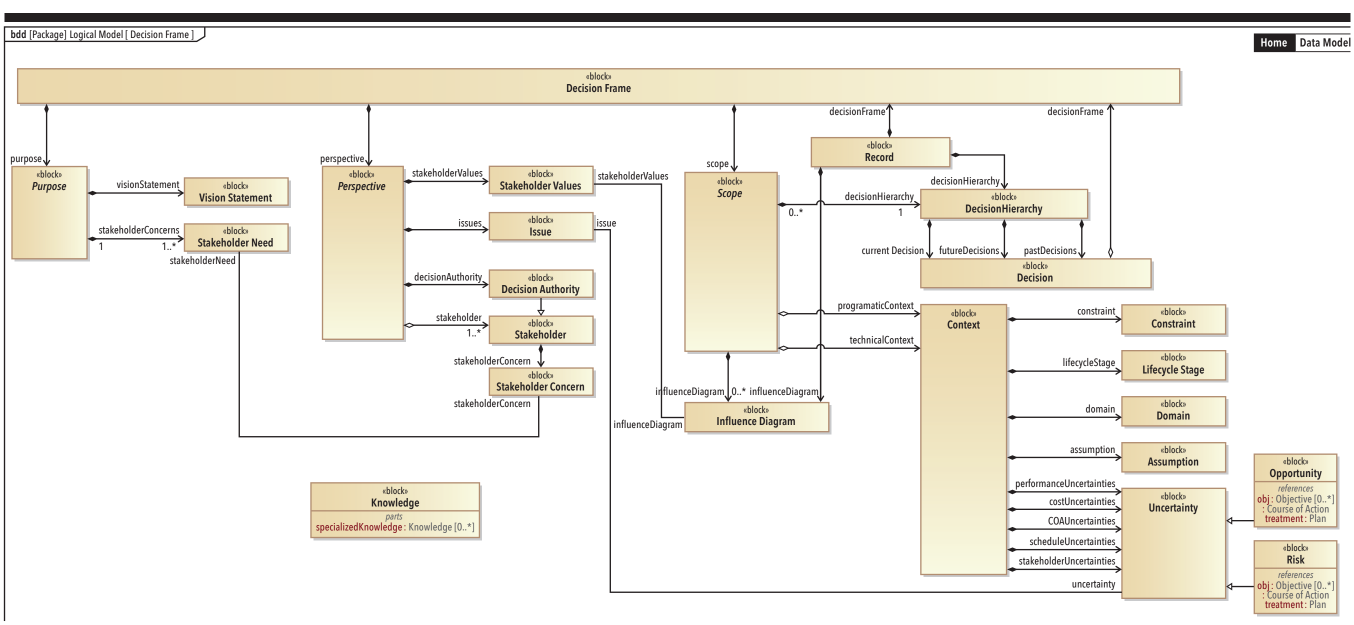


Figure 5. *Logical data model: block definition diagram for step 1, frame decision*

Table 3. DADM tailoring and reuse guidance

Step	Tailoring and Reuse Guidance	Impact of Not Doing
1.0 Frame Decision	The decision(s) should be clearly identified in each life cycle stage or decision point. It is usually helpful to begin with the decisions made in the previous life cycle stage and identify revised and additional decisions.	The system may be developed for the wrong problem, not take advantage important opportunities, or not address stakeholder concerns.
2.0 Structure Objectives and Measures	The value hierarchy should be updated based on the current information in each stage. New objectives and measures may be needed based on latest information about the problem and the system.	The alternative design and evaluation will not focus on the decisionmaker and stakeholder objectives. This may lead to rework and schedule delays.
3.0 Generate Creative Alternatives	It is important to generate creative alternatives that span the decision space. The alternatives may include previous alternatives and new alternatives to increase value and/or reduce risk.	The decision can only be as good as the best alternative. Poor alternatives lead to poor decisions.
4.0 Identify Areas of Uncertainty	The sources of uncertainty will vary in each life cycle stage. Some uncertainties will be resolved, and new risks and opportunities will be identified.	Systems are designed for years and sometimes decades. There are many uncertainties. If they are not considered, rework will be required.
5.0 Plan Evaluation Methods	As better information becomes available on the system alternatives in the life cycle, higher fidelity evaluation methods can be used.	The lack of an alternative evaluation plan to obtain appropriate fidelity evaluation methods for the life cycle stage can lead to rework and schedule delays.
6.0 Evaluate Alternatives	The alternative evaluations should be updated based on new alternatives, new uncertainties, improved evaluation methods, and updated data.	Poor evaluation can lead to rework, schedule delays, or system cancelation.
7.0 Improve Alternatives	It is important to improve the alternatives using the information from the alternative evaluations. Value gaps can be identified and filled. Risk can be mitigated, and new opportunities can be addressed.	Not improving the alternatives before the decision can result in less value, more risk, and rework.
8.0 Assess Trade-offs	Once the alternatives have been improved, the next step is to update the value, performance, cost, and schedule trade-offs. The impacts on other systems and stakeholders should also be considered.	Unless there is one alternative that dominates all the others, there will be trade-offs between the alternatives that decision makers may not consider.
9.0 Communicate Recommendations and Implementation	The systems engineers (system analysts) should communicate the trade-offs, recommendation(s), and implementation plan to the system decision maker (s) for their decision using the organization's recommended decision process.	Without clear and concise communication of the recommendations, the best decision may not be made or be actionable.
10.0 Make the Decision	Record the decision and rationale using organizational procedures.	Without records, we lose traceability of past decisions and rationale.

by the 8 steps (1.1 to 1.8) shown in Figure 4. The process uses inputs from past decisions and from subject matter experts (SMEs). The process ends with a well-formed decision frame. The first step is to *1.1 identify the stakeholders* that need to be included in the decision at hand. Every decision is potentially unique and may require different stakeholders. By using interviews, surveys, and facilitated brainstorming sessions, the stakeholders' needs will be captured to help *1.2 set the decision context*, and *1.3 develop the vision* that will define the decision purpose and help keep the decision analysis efforts focused. In addition to the context and vision it is also important to *1.4 raise issues* that may have been identified by the stakeholders, SME knowledge or past decisions, and *1.5 categorize issues*. The issues, stakeholder needs, context, vision, are used to help identify uncertainty, risk, and or opportunity. This information is used to *1.6 develop and modify a decision hierarchy*. It is also used to *1.7 develop/ modify influence diagrams*. The context and vision for the decision are combined with the decision hierarchy and influence diagram resulting in the *1.8 decision frame*. (See Figure 5)

Table 3 provides important guidance for reuse and tailoring for each step to the decision management process for your organization (System 2) and your system of innovation (System 3). The quality of each step should be validated with the decision-makers(s) and key stakeholders. We do not recommend skipping any step, but your organization may choose to combine steps. For example, the systems decision process (Driscoll et al. 2022) uses problem definition (steps 1 and 2), solution design (steps 3 and 7), decision making (steps 5 and 8) and solution implementation (steps 9 and 10).

Table 4. Illustrative decisions and data availability throughout the system life cycle (adapted from Walden et al. 2023)

Life Cycle Stage	Illustrative Decisions	Data Availability
Concept Stage	Assess Technology Opportunity / Initial Business Case • Of all the potential system concepts or capabilities that could incorporate the emerging technology of interest, do any offer a potentially compelling and achievable market opportunity? • Which should be pursued, when, and in what order? Inform, Generate, and Refine a Concept • What requirements should be included? • What needs to be accomplished and what can be traded away to achieve it within anticipated cost and schedule constraints? • How should requirements be expressed such that they are focused, yet flexible? • How can the set of requirements be demonstrated to be sufficiently compelling while at the same time achievable within anticipated cost & schedule constraints?	Descriptive data on existing systems. Predictive data using low fidelity models for new concepts Predictive data on new technologies and concepts using low fidelity models
	Create Solution Class Alternatives and Select Preferred COA • After considering the system level consequences of the sum of solution class alternatives across the full set of stakeholder values (to include cost and schedule), which solution class alternative should be pursued?	
Development Stage	Select/Define System Elements • After considering the system-level consequences of the system element design choices across the full set of stakeholder values (to include cost and schedule), which system-element alternatives should be pursued? • Make or buy decisions for system, subsystems, and elements	Descriptive data on existing system elements Predictive data on new system elements and the system using high fidelity models
	Select/Design Verification and Validation Methods • Is prototyping warranted? • What verification and validation methods should be performed (test, demonstration, analysis/simulation, inspection)? • What are the verification and validation plans?	Development test data in later parts of the stage
Production Stage	Develop Production Plans • What is the target production rate? • To what extent will low-rate initial production be used? • What is the ramp-up plan? • What production processes will be used? • Is the system still affordable?	Initial operational test and evaluation data Descriptive production quality data
Utilization and Support Stages	Utilization decisions • What are the best operations concepts for the resources available? • Support decisions • What is the maintenance strategy? • What is the logistics concept? • What is the preventive-maintenance plan? • What is the corrective-maintenance plan? • What is the spare-parts plan? • Is the system still affordable?	Operational and logistics descriptive data Predictive data using high fidelity operational and logistics models
Retirement Stage	Retirement Plan • When is it time to retire the system? • How will disposal of the system materials be accomplished?	Operational and logistics descriptive data Predictive data using high fidelity operational and logistics models

DISCUSSION

The DADM can be used in the concept stage and reused to inform life cycle decisions throughout the system life cycle. In Table 4, we use the generic life cycle (*15288:2023–ISO/IEC/IEEE International Standard–Systems and Software Engineering—System Life Cycle Processes* 2023, Walden et al. 2023) to provide decisions opportunities to improve the system value that are commonly encountered throughout a system's life cycle. Many of these decisions would benefit from a DADM that integrates the data produced by performance, value, cost, and schedule models that are meaningful to the decision makers and stakeholders. The table also lists the types of data that would be available in DADM for each life cycle stages.

Looking ahead, the DADM has a clear path for incremental development through a series of targeted future activities. The next step is the development of example implementations and case studies. By piloting DADM in real-world contexts—such as system acquisitions, technology selection, or lifecycle planning—we intend to demonstrate its practical value, collect lessons learned, and gather specific feedback to further refine the model. These case studies will help build a stronger evidence base for DADM's utility and support broader adoption by offering concrete examples and guidance for potential users.

We will continue iterating on the model alongside this effort, aiming for the release of DADM v2.0 in late 2025. This version will include a physical data model, shaped by insights from pilot use cases, and will serve as a detailed reference for organizations evaluating DADM. Maintaining this feedback-driven refinement process will help ensure the model continues to meet the needs of the systems engineering community.

Technical integration is another near-term opportunity. Aligning DADM with SysML v2 and BPMN will promote interoperability with contemporary MBSE tools and strengthen traceability from requirements through decision analytics. Another active area of exploration is the incorporation of artificial intelligence (AI)—especially large language models (LLMs)—to assist with data ingestion, recommendation generation, and automation of alternative evaluation or risk assessment. A longer-term goal is to enable auditable, AI-enabled decision support by capturing rationale and potential bias, thereby supporting regulatory and ethical compliance.

In parallel, pursuing formal standardization remains important. As we validate and improve DADM through pilot projects, workshops, and collaboration with industry, steps toward a standards designation will encourage wider, more consistent adoption across organizations. Through these targeted activities, DADM will continue to advance toward enabling data-driven, transparent, and auditable decision-making in MBSE and digital engineering environments. ■

REFERENCES

- "6.8 Decision Analysis." 2019. In *Systems Engineering Handbook*. NASA. https://www.nasa.gov/reference/6-8-decision-analysis/.
- 15288:2023 - *ISO/IEC/IEEE International Standard – Systems and Software Engineering—System Life Cycle Processes*. 2023. IEEE.
- Bass, Len, Paul Clements, and Rick. Kazman. 2003. *Software Architecture in Practice*. 2nd ed. SEI Series in Software Engineering. Addison-Wesley.
- Buede, Dennis M, and William D. Miller. 2016. *The Engineering Design of Systems: Models and Methods*. 3rd ed. Wiley. http://ebookcentral.proquest.com/lib/purdue/detail.action?docID=4391537.
- Data Management Association. 2017. *Data Management Body of Knowledge (DMBOK)*. Second edition. 1 online resource (626 pages). Technics Publications. WorldCat. https://proquest.safaribooksonline.com/978163462247.
- Department of Defense. 1994. *Systems Engineering*. MIL-STD-499B. Version Draft. Washington, DC, May 6.
- Driscoll, Patrick J., Gregory S. Parnell, and Dale L. Henderson. 2022. *Decision Making in Systems Engineering and Management (3rd Edition)*. John Wiley & Sons. https://app.knovel.com/hotlink/pdf/id:kt013XX851/decision-making-in-systems/stakeholders-vested-interests.
- ISO - International Organization for Standardization. 2023. *Systems and Software Engineering — System Life Cycle Processes*. ISO/IEC/IEEE 15288:2023. International Organization for Standardization.
- "OMG Systems Modeling Language (OMG SysML) Version 1.7." 2024. June. https://www.omg.org/spec/SysML/1.7/.
- Parnell, Gregory S., C. Robert Kenley, Devon Clark, et al. 2025. "Decision Analysis Data Model for Digital Engineering Decision Management." *Systems* 13 (7). https://doi.org/10.3390/systems13070596.
- Pinon Fischer, Olivia J., John F. Matlik, William D. Schindel, et al. 2022. "Digital Twin: Reference Model, Realizations, and Recommendations." *INSIGHT* 25 (1): 50–55. https://doi.org/10.1002/inst.12373.
- Schindel, Bill, and Rick Dove. 2016. "Introduction to the Agile Systems Engineering Life Cycle MBSE Pattern." *INCOSE International Symposium* 26 (1): 725–42. https://doi.org/10.1002/j.2334-5837.2016.00189.x.
- SEBoK Editorial Board. 2024. *The Guide to the Systems Engineering Body of Knowledge (SEBoK)*. V 2.11. The Trustees of the Stevens Institute of Technology. www.sebokwiki.org.
- Spetzler, Carl S., Hannah Winter, and Jennifer Meyer. 2016. *Decision Quality : Value Creation from Better Business Decisions*. Wiley.
- "SYSTEM ENGINEERING LABORATORY." 2025. June. https://www.incose.org/learn/se-laboratory.
- *Systems Engineering Vision 2035*. 2021. International Council on Systems Engineering. https://www.incose.org/publications/se-vision-2035.
- Walden, David D., Thomas M. Shortell, Garry J. Roedler, et al., eds. 2023. *Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities*. 5th ed. INCOSE Systems Engineering Handbook. Wiley.

ABOUT THE AUTHORS

Gregory S. Parnell is a professor of practice in the department of industrial engineering at the University of Arkansas and director, system design and analytics laboratory (SyDL), and director of the MS in operations management and MS in engineering management programs. He was lead editor of *Decision Making for Systems Engineering and Management*, (2nd Ed, 2011), lead author of the *Handbook of Decision Analysis, Wiley Operations Research/Management Science Series* (2013), and editor of *Trade-off Analytics: Creating and Exploring the System Tradespace* (2017). He previously taught at the US Military Academy at West Point, the US Air Force Academy, the Virginia Commonwealth University, and the Air Force Institute of Technology. He has a PhD from Stanford University. He is a retired Air Force Colonel.

C. Robert Kenley is a professor of engineering practice at Purdue's Edwardson School of Industrial Engineering, where has been developing courses and curricula to support the educational objectives of the Purdue Systems Collaboratory. He has over 30 years' experience in industry, academia, and government as a practitioner, consultant, and researcher in systems engineering. He has published papers on systems requirements, technology readiness assessment and forecasting, Bayes nets, applied meteorology, the impacts of nuclear power plants on employment, model-based systems engineering, and agent-based modeling for systems of systems. He is an expert system engineering professional (ESEP), and a fellow of INCOSE. He holds an SB in management from the Massachusetts Institute of Technology, an MS in statistics from Purdue, and an MS and a PhD in engineering-economic systems from Stanford.

Devon Clark is a leader in systems engineering, integration, and test (SEI&T), digital engineering, and decision analysis with over 25 years combined experience as a contractor and civilian working in (or for) US defense, security, and justice government agencies. Devon is a leader in Deloitte's mission solutions community of practice (CoP), a member of the INCOSE Corporate Advisory Board, and co-chair of the Decision Analysis Working Group. Devon specializes in digital transformation, model-based systems engineering (MBSE), and decision analysis. He holds a bachelors degree in physics from Georgia Tech and a masters degree in systems engineering from the Naval Postgraduate School.

Jared Smith is a manager at Deloitte Consulting, specializing in digital engineering for defense programs. He serves as co-chair of INCOSE's Decision Analysis Working. Prior to joining Deloitte, he supported multiple defense manufacturers as a model-based systems engineer and trainer. His expertise includes digital development environments, enterprise architectures, data-driven decision analysis, and generative AI applications. He holds a BS in industrial and systems engineering from the University of Alabama in Huntsville and is a certified six sigma black belt.

Frank Salvatore received his bachelor of science degree (BS) in electrical engineering (EE) from the New Jersey Institute of Technology (NJIT) and his Master of Science (MS) in computer science (CS) from American University (AU.) He is a senior systems engineer and technical fellow in the area of model-based systems engineering (MBSE) for SAIC. He has been a consultant for SAIC since 1999 providing various degrees of systems engineering support to the US Army, the Office of the Under Secretary of Defense, and the Navy. He also worked for the US Army and ITT industries providing a variety of systems engineering support for the development of army smart munitions, GPS navigation payload, and a software programmable radio. He is an expert systems engineering professional (ESEP) an Object Management Group (OMG) certified systems modeling professional (OCSMP), and six sigma green belt. He is also a member of the International Council on Systems Engineering (INCOSE), is the past president of the Liberty Chapter, the chair of the Decision Analysis Working Group, and the co-chair of Digital Engineering Information Exchange Working Group (DEIX WG.)

Chiemeke Nwobodo is a senior consultant at Deloitte with experience in the IT, cybersecurity, and systems engineering industry serving several clients. He works to solves complex problems within organizations using his knowledge of IT asset management, enterprise architecture, and model-based systems engineering. He is constantly expanding his expertise by obtaining certifications such as AWS, ServiceNow, TOGAF, Sec+ which assist him in critically considers different approaches when tackling new challenges. With the passion he has for MBSE, he has become deeply involved with the model development of the decision analysis data model in pursuit of achieving the INCOSE *Systems Engineering Vision 2035*. He holds a BSc in information sciences and technology from the Pennsylvania State University.

Sheena Davis is a consultant at Deloitte. She has utilized her healthcare and technological knowledge to providing case management, quality assurance, and IT support for the Department of Veteran Affairs. Additionally, she is an AWS certified cloud practitioner and is working to obtain to more certifications to expand her knowledge and skills. She holds a BS in computer science from Old Dominion University, a BA in psychology from The University of North Carolina at Chapel Hill, and an occupational therapy doctorate from Virginia Commonwealth University.

An Implementation of DADM Using Semantic Interoperability and Visualization

Thomas Hagedorn, thagedor@stevens.edu; **Daniel Dunbar,** ddunbar1@stevens.edu; **Joshua Bernstein,** jbernst1@stevens.edu; and **Mark Blackburn,** mblackbu@stevens.edu

ABSTRACT

The decision analysis data model (DADM) formalizes an architecture for decision making in engineering. This article describes a DADM implementation called the armaments interoperability and integration framework (IoIF). IoIF is a configurable software framework that supports engineering analysis and decision-making. IoIF uses linked data to facilitate data interoperability across mission, system, and discipline-specific models. At its core is a semantic representation of a system and a formalized model of the system analysis process. This can be applied to decision analysis as described by DADM. Using IoIF and linked front ends, a user can incorporate engineering analyses into analytic workflows to aid in decision making.

KEYWORDS: decision-making, decision analysis, data visualization, digital-thread system analysis, linked data, semantic web, ontology

INTRODUCTION

High quality decision making is of the utmost importance across many fields and needs to factor. Decision making methods in engineering must associate stakeholder preferences with computable metrics and thus must consider both mission and systems engineering processes and discipline-specific engineering analyses. Thus, research into decision making has yielded a wealth of preference elicitation methods, analytic techniques, decision methods or aids. A comprehensive survey can be found in the "Decision Analysis Data Model for Digital Engineering Decision Management" (Parnel et al.), which describes a conceptual and logical formalization that aims to describe the process and associated artifacts underpinning decision analysis. It defines ten broad steps and associated artifacts to ascertain the decision space and characterize uncertainty. The ten steps aim to comprise all components of a rigorous decision-making process. This article summarizes an implementation of DADM supporting a tradespace analysis methodology using IoIF The IoIF linked data platform for decision making is described in the Handbook on Digital Engineering using Ontologies (Blackburn et al. 2025).

BACKGROUND AND UNDERLYING TECHNOLOGIES

The authors developed IoIF leveraging a linked data platform to aid the integrated systems engineering decision method (ISEDM) for a US Army sponsor (Cilli 2015). ISEDM describes a process for decision making that hews closely to the one formalized in DADM. ISEDM's value model uses a weighted sum of single attribute, 5-point piecewise functions. These map raw metrics to stakeholder value on a normalized scale from 0 to 1. Objectives are organized into a hierarchy, with issues of co-variance and relative importance handled based on subject matter expert (SME) and stakeholder rankings of importance and differentiation. These are used to compute weights for the single-attribute value.

Semantic web technology (SWT) is a term characterizing standards, tools, and methods for describing the connections between data. This overlaps with the modern notion of linked data. SWT envisions various data items or repositories linked by directed, named connections with pre-agreed meanings defined in an ontology. Data are described as a directed, labelled

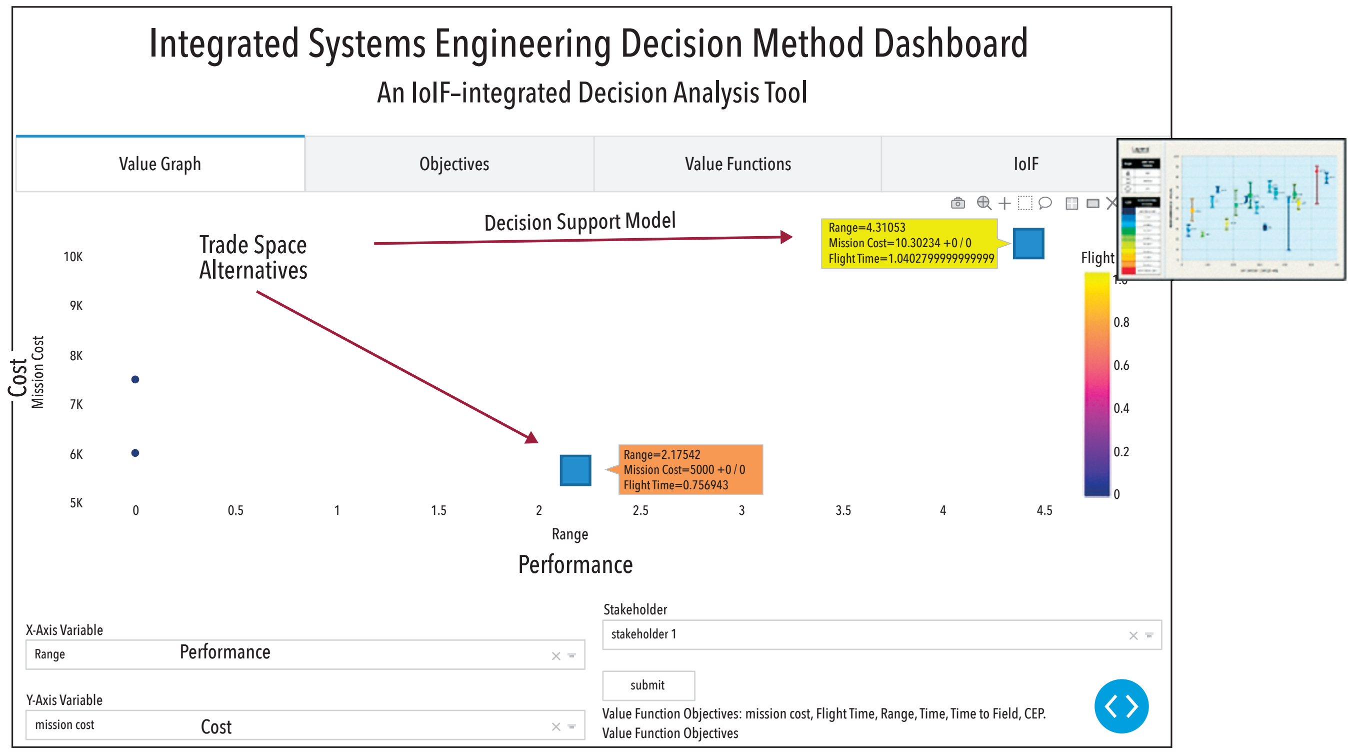


Figure 1. Decision visualization page of the DAD

graph and accessed by traversing connections or describing connection patterns.

VISUALIZATION OF DECISION TRADE SPACE OF ALTERNATIVES

The IoIF implementation of DADM relies upon several pieces: a mission and system model (step 2: structure objectives and measures), an analysis model describing as an assessment flow diagram (AFD) (step 5: plan evaluation methods; discussed below), a structural model of the decision data required by a specific method (step 1: frame decision), a semantic tool to parse this information into a tool-agnostic system representation, and tools capable of interacting with the semantic layer. The IoIF decision analysis dashboard (DAD) (Figure 1) provides tradespace visualization, supporting DADM step 7: improve alternatives, step 8: assess trade-offs, step 9: communicate recommendations and implementation plan and step 10: make decision. The dashboard uses an application agnostic query to retrieve information related to a decision analysis from IoIF.

DAD is split into a visualization tab and tabs to edit the preference model and synchronize to IoIF. The backend implements the weight calculation and computation of stakeholder value via the piecewise value functions, with user modifications triggering re-calculation. Of particular importance to the DADM process is the visualization tab, which takes data from IoIF defined in the system models and calculated according to AFD (step 6: evaluate alternatives). The AFD provides a representation of the links between system attributes, objectives, alternatives, and uncertainty represented as error bars. The DAD implements ISEDM's value functions in the context of a provided decision space, yielding an interactive scatterplot defaulting to a cost metric vs. overall value. Users can assign values to either axis, as well as to color and marker size scales. Across the DADM process, a user can load the dashboard and inspect performance, uncertainty, etc.

IMPACT VISUALIZATION

Like the DAD, the digital thread impact analysis dashboard (IAD) queries IoIF to deliver mission and system agnostic visualization and functions. The IAD is organized into a base impact visualization used for what-if analysis, a linked instantiation tab that facilitates subsequent analysis by permitting a user to create and define new alternatives, a data inspection tab to view parameter values for a given alternative, a requirement definition tab, and a tab that displays performance values against requirements.

The IAD uses the information encoded within the AFD and translated to graph patterns in the semantic layer to track relations between parameters, analysis steps, requirements, and system components rendered as color-coded network graph. When a user selects a parameter, the dashboard traces these

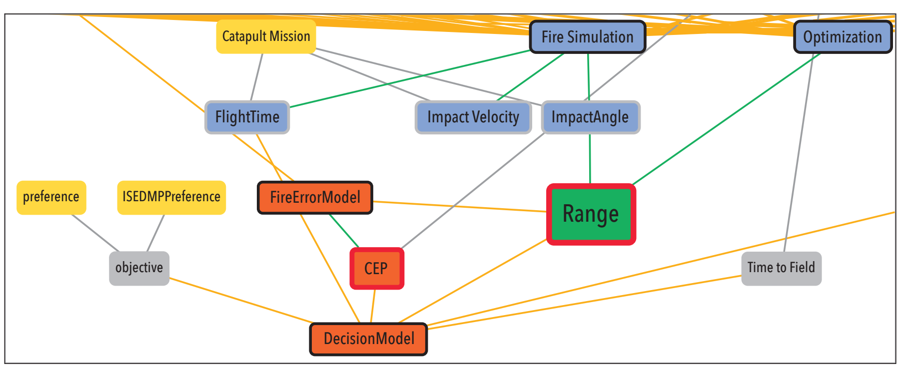


Figure 2. IAD visualization of the AFD and its connections used to display part of the digital thread defined in the AFD using black, gray, and yellow to represent model interfaces parameters and system components respectively

connections to determine impacts and dependencies in the mission and system. Impacts are traced to affected system parts, requirements, analysis sub-steps, and parameters. Dependencies are traced back to design inputs. A user can either view the impact of a change in terms of affected parameters or re-analyze or track a performance parameter they wish to improve back to a design parameter. These views provide insights related to step 7: improve alternatives, step 9: communicate recommendations and implementation plan, and step 10: make decision. (See Figure 2)

Irrespective of use case, the dashboard provides tools for setting up subsequent analyses to try to improve or verify the system in the case of some unexpected change. The instances tab lets a user implement a contextual copy from some reference state – a run of the analysis with specific parameters – to a second, subject to the impact/dependencies calculated for the selected parameters. Upon a user prompt, a new analysis instance is added to the graph and initialized with unaffected parameters copied from the referent or provided by the user and affected parameters left blank for subsequent analysis. This individual can then be used to conduct a subsequent run of the analysis described in the AFD.

MODELS UNDERLYING THE DECISION METHODOLOGY

A descriptive model implemented in the System Modeling Language (SysML) forms the backbone of the implementation. This serves as the implementations basis for understanding the specific context of a decision and is used to instantiate the semantic layer's backend graph database (aka triplestore). Modeling choices are left to the system modeler.

Irrespective of the specific content of a model, a SME can extend the model with metadata. The tags correspond to terms in the ontology ecosystem and help indicate the model's meaning so IoIF can populate a semantic database. This tagging approach means that modeling and SWT modeling SMEs can work independently using imports or branches to avoid conflicts, and existing models can be retrofitted as needed without model rework.

Analytic Model – Assessment Flow Diagrams

The system model is extended with diagrams describing an analysis (or analyses) in the context of a system model. This model of the analysis provides an explicit, model-based representation of the steps, data, and tools used to characterize a system. This describes dataflows through

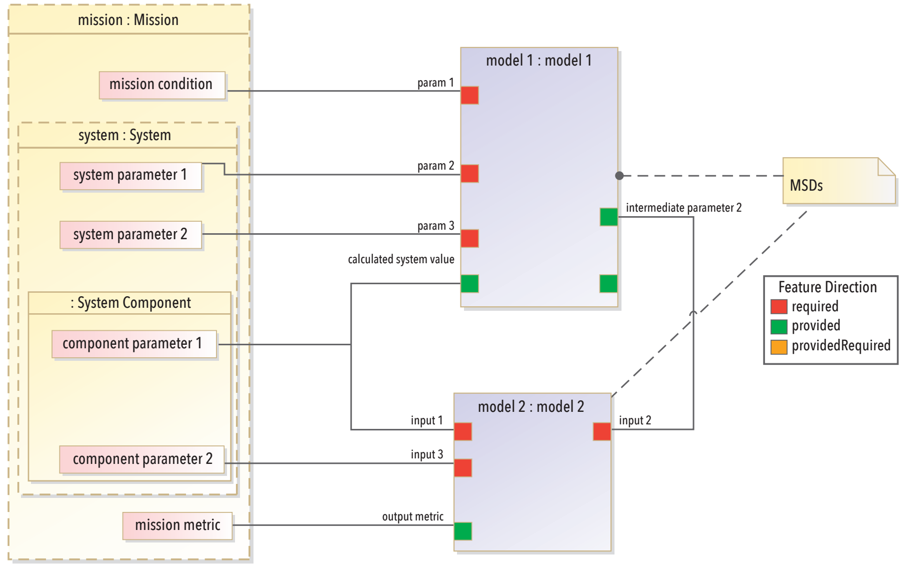


Figure 3. Simple AFD for a generic System

a data ecosystem and their metadata and format requirements while preserving understanding of what every data item means in the context of the mission and system.

The model of the analysis is implemented as a SysML formalization of the ISEDM notion of an AFD (step 5: plan evaluation methods in DADM). The AFD describes a mapping from low-level design parameters through various intermediate steps and their parameters, until linking to high-level metrics used characterize performance against objectives. Each AFD is modeled as a parametric (Figure 3) that represents intermediate steps as a black box with ports indicating input and output data and its labelling. As these typically correspond to a specific model or tool to be invoked in the analysis, the black boxes are called model interface specification diagrams (MISDs) (Dunbar et al. 2023) and represent model or tool's own conventions for data markup. For example, labelled tagged port labels might correspond to variable names in a MATLAB model.

Connectors wired from ports to other ports or system attributes describe the flow of data through the analysis process and the context of the mission and system. These connections define a mapping between the system components, activities, and attributes and the data values exchanged in an analytic process happening outside the model. The AFD support aspects of step 1: frame decision and step 2: structure objectives and measures.

Decision Model

This AFD can be extended with a decision analysis model (Figure 4). While not currently linked to DADM's architecture, it provides a lightweight representation general to any decision, and describes similar entity types (objectives, preferences, stakeholders, etc.) corresponding to similar ones found in DADM.

The decision model extends and encompasses the AFD. A modeler extends

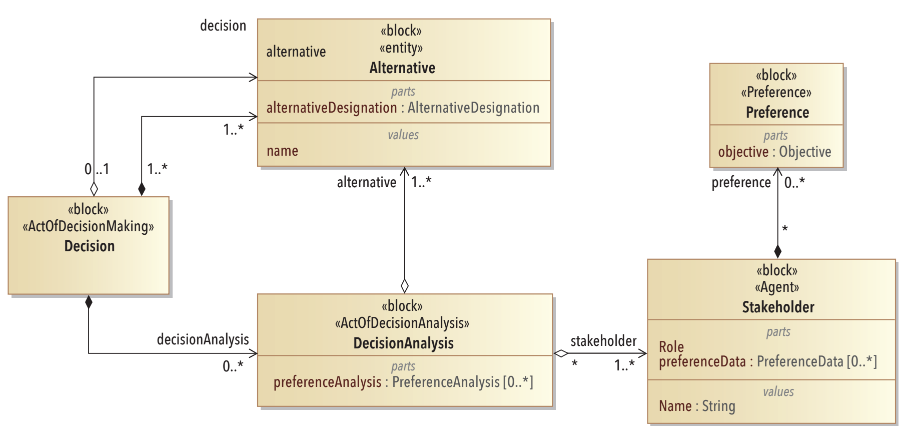


Figure 4. Partial view of the root decision model with metadata tags

the model such that parameters described in the AFD become metrics of objectives and act as inputs to the decision analysis. Collectively, this creates a representation of the model wherein alternatives comprise system configurations to be characterized in the AFD process. A modeler can extend the model to describe the data requirements of specific decision analysis methods. The DAD relies upon an extension modeling ISEDM's handling of weights, SME ranking, and the piecewise value functions. When instantiated as SysML instance specification elements, the resulting model structures have slots for every value that is needed to compute an overall stakeholder value for some alternative, which the DAD can then access via IoIF's database.

Instance specifications link specific objectives. The decision is modeled via an analysis element which generalizes to a ISEDM decision analysis block. An MISD then links objective information to metrics calculated as part of the analysis process. When instantiated in the model, the decision information is ingested by the semantic layer and exposed via interfaces to its database.

DISCUSSION AND CONCLUSIONS

While IoIF and the two dashboards proved useful in DADM's implementation, this proof of concept does not completely cover DADM. Nonetheless, the IoIF framework, methodology and dashboards are largely sufficient to aid in the execution of DADM's model of the decision-making process. The model-centric approach formalizes many of the steps' outcomes in a computational format supporting linked tools. Characterizing the decision yields a model describing links between objectives and the mission or system attributes. The AFD further links analytic parameters to mission and system traits.

The model formalization works as both documentation and refinement of the activities of steps 1-5 of DADM's process. The semantic framework and visualization tools provide graphical front-end and data access to aid in the remaining steps. During step 6, IoIF exposes customized payloads of parameters to individual modeling tools or analyses as described in the AFD, leaving only a question of how to receive data. During step 7 the DAD provides a visual representation of each alternative's performance, like that presented alongside DADM and used in ISEMD. The interactive environment allows multi-dimensional problems analyses visualized for stakeholders. The IAD provides insights and mechanical setup for completing step 8. As the decision maker and SMEs strive to communicate recommendations and make the decisions (steps 8-10). Thus, while not a complete execution, IoIF provides aids to implement much of the DADM. ■

REFERENCES

- Blackburn, Mark, Daniel Dunbar, Tom Hagedorn, Steven Hespelt, Charles Collard, and J. Steven Jenkins. 2025. "Handbook on Digital Engineering with Ontologies version 2.0." Report, Stevens Institute of Technology, Systems Engineering Research Center. https://www.cto.mil/wp-content/uploads/2025/06/SERC_Handbook-on-Digital-Engineering-with-Ontologies_2.0.pdf.
- Cilli, M. 2015. "Seeking Improved Defense Product Development Success Rates Through Innovations to Trade-Off Analysis Methods." PhD dissertation, Stevens Institute of Technology (Hoboken, US-NJ).
- Dunbar, Daniel, Thomas Hagedorn, Mark Blackburn, and Dinesh Verma. 2023. "A Three-Pronged Verification Approach to Higher-Level Verification Using Graph Data Structures." *Systems* 12 (1): 27. https://doi.org/10.3390/systems12010027.
- Hagedorn, T., M. Bone, B. Kruse, I. Grosse, M. Blackburn. 2020. "Knowledge Representation with Ontologies and Semantic Web Technologies to Promote Augmented and Artificial Intelligence in Systems Engineering." *INSIGHT* 23 (1): 15-20.

ABOUT THE AUTHORS

Dr. Tom Hagedorn is a senior research scientist in the Systems Engineering Research Center (SERC) at Stevens Institute of Technology. His research focuses on the use of semantic web technologies and linked data to support systems engineering, engineering design, analysis, and decision-making.

Dr. Daniel Dunbar is a research scientist in the Systems Engineering Research Center (SERC) at Stevens Institute of Technology. His research focuses on the use of semantic web technologies and linked data to support systems engineering, engineering design, analysis, and decision-making.

Joshua Bernstein is a research associate in the Systems Engineering Research Center (SERC) at Stevens Institute of Technology. He leads the development of the interactive service dashboard which include the decision and impact analysis dashboards.

Dr. Mark R. Blackburn is a senior research scientist in the Systems Engineering Research Center (SERC) at Stevens Institute of Technology. He has been the principal investigator on 20 SERC research tasks for the US Navy, Army, Space Force, and OUSD(R&E), broadly spanning digital engineering models, methods, and tools.

The Need for a Shared Vocabulary of Digital Engineering

Joe Gregory, joegregory@arizona.edu; **Clarence (Moe) Moreland,** moe.moreland@mtsi-va.com; **James S. Wheaton,** james.wheaton@terumobct.com; and **Celia Tseng,** celia.tseng@3ds.com

ABSTRACT
In recent years, many organizations have embraced digital engineering (DE) as a strategy for enhancing integration and interoperability across the system lifecycle. However, research into the development of DE practices has shown that a lack of consensus on terminology can hinder progress. Common terms such as 'digital thread', 'digital twin', and 'authoritative source of truth' are defined inconsistently across domains and organizations, creating friction in digital information exchange. Automated, efficient information exchange requires a precise lexicon to facilitate understanding for humans and machines. The INCOSE Digital Engineering Information Exchange (DEIX) Working Group is working to address this challenge by developing a formal ontology of DE concepts. This article addresses some of the key terminology challenges facing DE practitioners and describes how a machine-readable ontology can help to create a shared understanding of DE and enable more effective implementation of DE practices.

KEYWORDS: digital engineering, model-based engineering, vocabulary, ontologies

INTRODUCTION

Digital engineering (DE) is transforming how systems are designed, developed, and managed. By linking models, data, and tools across the lifecycle, DE promises traceability, interoperability, and improved decision-making (US DoD 2023, Voth and Sturtevant 2022). Yet this transformation depends not only on technical integration, but also on a shared understanding of the concepts involved (Gregory et al. 2025). Despite the increasing adoption of DE principles, terminological inconsistency remains a significant barrier to effective implementation.

Terms like 'digital twin,' 'digital thread,' and 'authoritative source of truth' are used frequently – but rarely consistently. Different standards bodies, organizations, and tools define these terms in subtly different ways. This variation creates ambiguity in communication, leads to misalignment across toolchains, and undermines efforts to build a coherent digital thread. Even within a single project, teams may interpret these concepts differently, resulting in mismatched assumptions and duplicated effort.

The International Council on Systems Engineering (INCOSE) Digital Engineering Information Exchange Working Group (DEIX WG) has identified these terminology challenges as a priority (INCOSE DEIX WG 2025). This article explores the implications of inconsistent DE terminology, offers examples of problematic definitions, and describes how the DEIX Ontology Working Group is developing a formal, concept-based ontology to address them.

THE CHALLENGE OF TERMINOLOGY IN DIGITAL ENGINEERING

While digital engineering is widely promoted as a unifying framework for modern system development, practitioners quickly encounter a problem: the same words mean different things to different stakeholders. 'Digital thread' is often assumed to mean a collection of linked authoritative digital information (AIAA 2023), but some define it as an enterprise-level analytical framework (DAU 2017), while others say it refers to the traceability from the digital twin back to the requirements (ISO/IEC/IEEE 2023). Similarly, 'digital twin' may refer to a high-fidelity simulation model (DAU 2017), a virtual representation of a connected physical asset (AIAA 2020), or, more generally, a digital asset on which services can be performed that provide value to an organization (ISO/TS 2019). In fact, through the course of this research, the authors have so far identified eight different definitions of digital twin (ISO 2021b, 2025, 2022; ISO/IEC 2023; Stark and Damerau 2019). While some of these definitions may be similar, it becomes very difficult to understand precisely what is required of a digital artifact for it to be considered a digital twin. The precise definition seems to vary depending on who is using the term and in what context. This is understandable as different domains of discourse may use digital twins in different ways, but it is important to understand where the common ground begins and ends so that practitioners can communicate consistently.

This lack of semantic alignment creates serious challenges for integration – particularly when integrators assume that different usages of a particular term are equivalent. Research into model-centric engineering (MCE) practices has shown that "confusing and overlapping terminology was hindering advancement and understanding of MCE" (Bone et al. 2019). When different groups have different understandings of what a particular term *means*, and the underlying definitions diverge, it becomes difficult to federate data, verify consistency, or ensure interoperability.

One of the most illustrative examples is the term 'authoritative source of truth' (ASOT). In some documentation, it refers to a dataset or model that holds verified information (US Space Force 2021). In others, it refers to the organizational authority responsible for maintaining that dataset (Object Management Group 2024). When engineers label a file as the ASOT, do they mean that it is authoritative because of its content, or because it was produced by a recognized source? Some practitioners also view the ASOT as the repository in which a particular document or model is kept, which may or may not conflict with the concept of 'digital repository.' Without clarification, tools that use this label cannot reliably interoperate.

Another example lies in the distinction between a 'digital engineering environment' and a 'digital engineering ecosystem.' While some sources use these terms to distinguish between an internal tool infrastructure (the environment) and a broader community of competing tools, standards, and stakeholders (the ecosystem), others appear to conflate the two (Gregory et al. 2025). This lack of clarity can become a barrier when coordinating across government-industry collaborations or multi-tier supply chains.

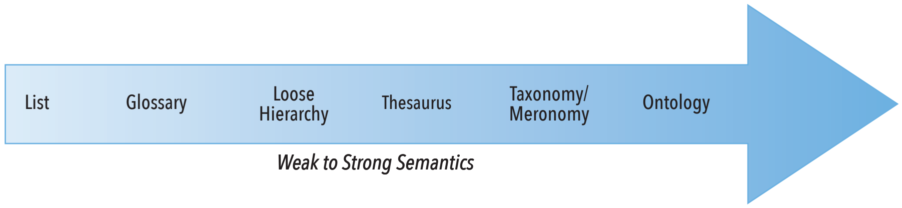


Figure 1. Knowledge information systems: weak to strong semantics, adapted from Lomax and Wolf (2021)

THE NEED FOR A SHARED ONTOLOGY

To resolve these inconsistencies, several DE practitioners and organizations are turning to ontologies (Dunbar et al. 2023; Gregory, Iyer, and Salado 2025). Ontologies are formal representations of domain concepts and their relationships (Gruber 1991). An ontology is more than a glossary: it defines the structure, constraints, and logical connections among concepts. In particular, a concept-based ontology allows practitioners to focus on the real-world phenomena that they are interested in, without getting caught up in terminology. By separating concepts from the terms used to describe them, an ontology enables alignment across tools, teams, and standards by recognizing that different domains may use different *terms* to refer to the same *concept*. In Figure 1, adapted from (Lomax and Wolf 2021), different knowledge organization systems are presented on a scale from weak semantics to strong semantics.

For example, an ontology can distinguish the concept of a 'digital artifact' from the various roles it might play (such as 'ASOT') and then define the conditions under which that role applies. These kinds of distinctions, captured logically in a formal ontology, enable both human understanding and machine reasoning and offer a step towards the formalization of an organization's lexicon. The use of ontology also allows organizations to use different terms (e.g., 'canonical model', 'trusted baseline') to refer to the same underlying concept, so long as the ontology maps them clearly – thus aiding interoperability across domain and organizational boundaries.

DEVELOPMENT OF THE DEIX ONTOLOGY

The INCOSE DEIX WG is addressing this need by developing a formal ontology tailored to digital engineering. Built using the Web Ontology Language (OWL) (W3C 2012) and grounded in the Basic Formal Ontology (BFO) (ISO 2021a), the DEIX ontology organizes key DE concepts into a coherent, machine-readable model.

The development process follows the well-established methodology by Noy and McGuinness (2001), starting with domain scoping and use case definition. This approach is displayed in Figure 2.

The WG began by defining the scope and domain of interest, selecting representative use cases, and identifying key concepts from authoritative sources. This iterative process ensures that the ontology addresses real-world engineering needs while maintaining conceptual rigor. The group identified three core use cases that the ontology must support:

1. Classifying digital artifacts
   i.e., determining the role or type of an artifact based on its characteristics, provenance, and context.
2. Assessing reusability
   i.e., verifying whether an artifact meets structural, contextual, or quality conditions to be reused in other projects or lifecycle stages.

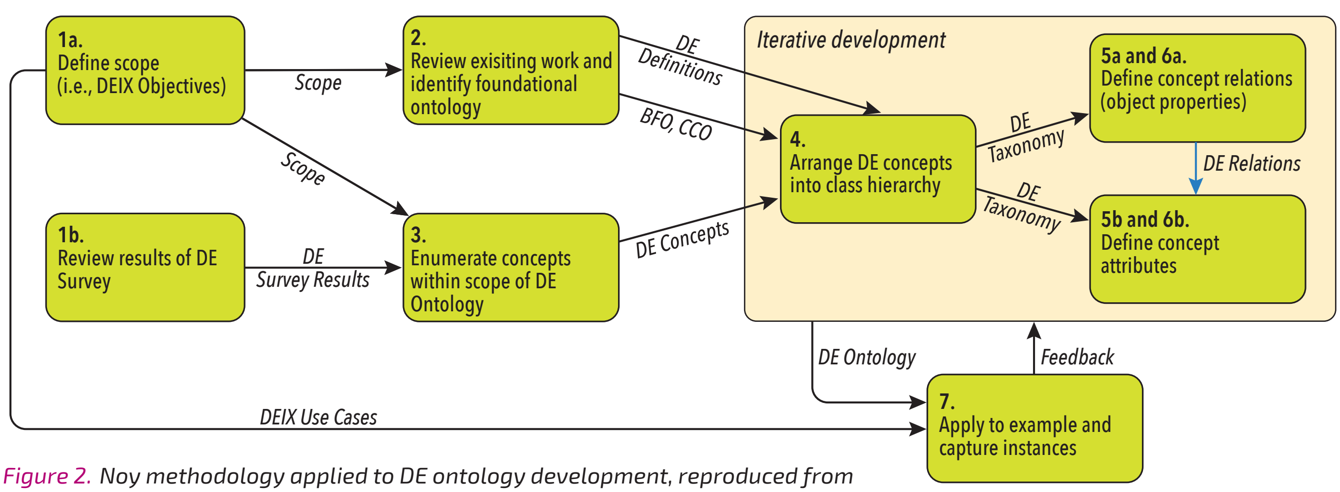


Figure 2. Noy methodology applied to DE ontology development, reproduced from Gregory et al. (2025)

3. Evaluating utility
   i.e., determining whether an artifact supports specific stakeholder goals, such as traceability, decision justification, or compliance.

To support these tasks, the ontology defines each concept not only by a textual description but by its logical relationships to other concepts. This enables automated classification, consistency checking, and inferencing across datasets. Off-the-shelf logical reasoners allow practitioners to automatically check whether any particular dataset adheres to the DEIX ontology and can detect any specific inconsistencies within it.

A key example of this approach is the modeling of the term ASOT. Across DE literature and practice, ASOT is used inconsistently: sometimes referring to a particular file, model, or repository; other times referring to the organizational role responsible for the information. In the DEIX ontology, ASOT is treated as a role that an artifact plays in a specific context. An artifact may bear the ASOT role if it (a) is designated as such by a recognized authority, (b) participates in an act of engineering (e.g., modeling, verification), and (c) conveys system-relevant information, such as stakeholder needs or technical intent. This definition distinguishes between the artifact and the authoritative role it plays, enabling clearer reasoning and reducing ambiguity when integrating across tools and domains. If, for example, we specify that an artifact bears the role of ASOT, but we do not specify the authority that designated that role or the context (i.e., engineering process) in which it applies, then a reasoner would flag this as being inconsistent with the DEIX ontology.

The ontology also supports query generation. By capturing formal relationships between artifacts, roles, and lifecycle stages, users can pose questions like ‘Which artifacts are missing required views?’ or ‘Which models were reused without meeting reusability criteria?’ These capabilities enhance transparency and reduce the manual burden of verification.

FUTURE WORK AND COMMUNITY INVOLVEMENT

The DEIX WG plans to continue refining the ontology through community feedback and real-world application. Key priorities include expanding coverage to new concepts (e.g., governance, trustworthiness) and aligning with other ontologies (e.g., the Industrial Ontology Foundry (IOF) systems engineering ontology). In the future, it may also be possible to develop plug-ins to integrate ontology-based reasoning into digital engineering tools.

The DEIX ontology will be verified using datasets that are representative of real digital artifacts contained within a digital environment. Use cases will be identified and developed to validate the DEIX ontology’s utility in the context of DE activities. One way this may be achieved is to use the DEIX ontology in conjunction with the shapes constraint language (SHACL) to validate datasets.

Ultimately, the DEIX WG is working towards the release of the DEIX ontology as a standard that captures the core concepts associated with DE and the terms that can be used to refer to these concepts across different domains.

The group also encourages participation from across the DE ecosystem. The ontology is hosted on the INCOSE GitHub (DEIX WG 2025), and practitioners are invited to contribute definitions, test use cases, highlight potential issues, and propose extensions. By collaborating around a shared semantic framework, the DE community can move beyond terminological confusion and toward truly interoperable digital engineering.

CONCLUSION

Digital engineering is only as effective as the clarity of its concepts. When key terms are used inconsistently, even the best-integrated toolchains will fail to deliver on the promise of the digital thread. The DEIX ontology provides a practical way to address these challenges by offering a logical foundation for how digital engineering data is structured, interpreted, and exchanged.

As the DE community continues to scale its ambitions, a shared ontology will be critical for building trust across disciplines, tools, and domains. The work is still evolving, and we must proactively align our language if we want to integrate our systems. ■

REFERENCES

- AIAA. 2020. “Digital Twin: Definition & Value.” *American Institute of Aeronautics and Astronautics (AIAA) Digital Engineering Integration Committee (DEIC)*, no. December: 1–16.
- ———. 2023. “Digital Thread: Definition, Value and Reference Model.” *American Institute of Aeronautics and Astronautics (AIAA) Digital Engineering Integration Committee (DEIC)*, no. June.
- Bone, Mary A., Mark R. Blackburn, Donna H. Rhodes, David N. Cohen, and Jaime A. Guerrero. 2019. “Transforming Systems Engineering through Digital Engineering.” *Journal of Defense Modeling and Simulation* 16 (4): 339–55. https://doi.org/10.1177/1548512917751873.
- DAU. 2017. “DAU Glossary.” Defense Acquisition University (DAU). 2017. https://www.dau.edu/glossary.
- DEIX WG. 2025. “INCOSE DE Ontology GitHub.” 2025. https://github.com/INCOSE/DEIX_Ontology.
- Dunbar, Daniel, Thomas Hagedorn, Mark Blackburn, John Dzielski, Steven Hespelt, Benjamin Kruse, Dinesh Verma, and Zhongyuan Yu. 2023. “Driving Digital Engineering Integration and Interoperability through Semantic Integration of Models with Ontologies.” *Systems Engineering*, no. January: 1–14. https://doi.org/10.1002/sys.21662.
- Gregory, Joe, Visalakshi Iyer, and Alejandro Salado. 2025. “Semantically Enabled Dashboards to Support Systems Engineers.” In *INCOSE International Symposium*, 1–17. Ottawa, CA.
- Gregory, Joe, Clarence Moreland, James S Wheaton, and Celia Tseng. 2025. “Towards a Digital Engineering Ontology to Support Information Exchange.” In *INCOSE International Symposium*, 1–16. Ottawa, CA.
- Gruber, Thomas R. 1991. “The Role of Common Ontology in Achieving Sharable, Reusable Knowledge Bases.” *Principles of Knowledge Representation and Reasoning: Proceedings of the Second International Conference*, 601–2. http://www.cin.ufpe.br/$~$mtcfa/files/10.1.1.35.1743.pdf.
- INCOSE DEIX WG. 2025. “INCOSE DEIX WG – Working Group Purpose and Mission.” https://www.incose.org/communities/working-groups-initiatives/digital-engineering-information-exchange.
- ISO/IEC/IEEE. 2023. “ISO/IEC/IEEE 24641:2023 Systems and Software Engineering — Methods and Tools for Model-Based Systems and Software Engineering.”
- ISO/IEC. 2023. “ISO/IEC 30173:2023. Digital Twin — Concepts and Terminology.”

- ISO/TS. 2019. “ISO/TS 1810101:2019 - Automation Systems and Integration – Oil and Gas Interoperability.”
- ISO. 2021a. “ISO 21838-2: Information Technology — Top-Level Ontologies (TLO) — Part 2: Basic Formal Ontology (BFO)” 2010.
- ———. 2021b. “ISO 23247-1:2021. Automation Systems and Integration — Digital Twin Framework for Manufacturing — Part 1: Overview and General Principles.”
- ———. 2022. “ISO 23704-2:2022. General Requirements for Cyber-Physically Controlled Smart Machine Tool Systems (CPSMT). Part 2: Reference Architecture of CPSMT for Subtractive Manufacturing.”
- ———. 2025. “ISO 16484-2:2025. Building Automation and Control Systems (BACS) — Part 2: Hardware.”
- Lomax, Jane, and Elizabeth Wolf. 2021. “The Evolution and Importance of Biomedical Ontologies for Scientific Literature [White Paper].” https://www.copyright.com/wp-content/uploads/2021/02/White-Paper-Evolution-Importance-of-Biomedical-Ontologies.pdf.
- Noy, Natalya F, and Deborah L McGuinness. 2001. “Ontology Development 101: A Guide to Creating Your First Ontology.” Stanford Knowledge Systems Laboratory.
- Object Management Group. 2024. “OMG Wiki - MBSE.” 2024. https://www.omgwiki.org/MBSE.
- Stark, Rainer, and Thomas Damerau. 2019. “Digital Twin.” In *CIRP Encyclopedia of Production Engineering*, edited by Lucienne Laperrière and Gunther Reinhart, 1–8. Berlin, Heidelberg, DE: Springer. https://doi.org/10.1007/978-3-642-35950-7_16870-1.
- US DoD. 2023. “Digital Engineering DoD Instruction 5000.97.”
- US Space Force. 2021. “U.S. Space Force Vision for a Digital Service.”
- Voth, J. M., and G. H. Sturtevant. 2022. “Digital Engineering: Expanding the Advantage.” *Journal of Marine Engineering and Technology* 21 (6): 355–63. https://doi.org/10.1080/20464177.2021.2024382.
- W3C. 2012. “OWL 2 Web Ontology Language. Structural Specification and Functional-Style Syntax (Second Edition).” https://www.w3.org/TR/owl2-syntax/.

ABOUT THE AUTHORS

Dr. Joe Gregory is a postdoctoral research associate at the University of Arizona. His research interests include digital engineering, model-based systems engineering, and semantic web technologies. His current research focuses on the development of the digital engineering factory, and the development of ontologies to support digital test and evaluation. He is the co-chair of the Digital Engineering Information Exchange (DEIX) Ontology Working Group.

Dr. James S. Wheaton is a principal systems engineer and the digital engineering champion at Terumo Blood and Cell Technologies. He is an INCOSE member in the Colorado Front Range chapter, and lead contributor in the INCOSE DEIX Ontology Working Group developing the DEIX ontology using the Web Ontology Language. His former career was in software engineering of ecommerce, big data, AI, and blockchain systems. James interned at NASA Jet Propulsion Laboratory from 2023-2024, where he developed model-based system architectures and requirements for two Mars sample return projects. James holds a BS in mechanical engineering from Purdue University (2011), a PhD in systems engineering from Colorado State University (2025), and OMG-certified systems modeling language professional (OCSMP) and associate systems engineering professional (ASEP) certifications.

Clarence ‘Moe’ Moreland. For more than twenty years of his 40 plus years of experience, Mr. Moreland has been the most senior authority and architect on over 100 programs and projects, the many of which are black, guiding numerous teams in sizes from four to fifty or more software, systems, and electrical engineers. Mr. Moreland is the co-chair of the Digital Engineering Information Exchange (DEIX) Ontology Working Group.

Celia Tseng is the director of Aerospace and Defense Consulting at Dassault Systems, with 21 years of systems engineering experience from Raytheon Technologies, Lockheed Martin, and SAIC. Celia is a co-chair of the joint INCOSE/ NDIA/ OMG Digital Engineering Information Exchange Working group and INCOSE standard liaison in ISO/IEC JTC 1 SC 7 digital engineering. She has a masters in systems engineering from Cornell University, a certified systems engineering professional (CSEP), a certified system modeling professional (OMG OCSMP) and certified agile scrum master (SAFe).

OCTOBER 2025 VOLUME 28/ ISSUE 5

OCTOBER 2025
VOLUME 28/ ISSUE 5

Embedding Digital Engineering into the Classroom

Joe Gregory, joegregory@arizona.edu; and **Alejandro Salado,** alejandrosalado@arizona.edu

ABSTRACT
The digital thread concerns the integration of engineering data across a system's lifecycle and has emerged as a cornerstone of modern digital engineering. One promising application is in engineering education, where it can expose students to authentic, connected workflows across multiple domains. The digital engineering factory (DEF), developed at the University of Arizona, is a web-based platform designed to support systems and software engineering students by providing integrated access to tools for project management, requirements, modeling, analysis, verification, and test planning. These tools are linked through Violet, a central hub that aggregates data into a comprehensive, semantically structured database. Using the ontological modeling language (OML) and the University of Arizona ontology stack (UAOS), the DEF supports reasoning, validation, and querying. These capabilities enable dynamic dashboards to guide students and assist instructors with assessment. In a classroom deployment, students worked in role-defined teams using the DEF to manage their project data and participate in model-based design and review. The DEF provided students with a clear view of the full engineering lifecycle and enabled automated grading based on traceable, semantically validated data. Ongoing development is focused on improving automated workflows and enhancing the student and instructor experience.

KEYWORDS: education, digital engineering, model-based engineering, ontologies

INTRODUCTION

As systems become increasingly complex and software-intensive, there is a growing need for integrated, traceable, and interoperable engineering processes. Industry has responded with the concept of the digital thread. The 'digital thread' refers to the continuous linking of engineering data across the entire system lifecycle, from concept to disposal (AIAA 2023, Dertien and Hastings 2021). This approach promises not only improved efficiency and error detection, but also deeper insight into trade-offs, requirements compliance, verification status, and system behavior.

Digital engineering has the potential to overcome the fragmentation that has historically plagued system development efforts. Engineering data is often dispersed across specialized tools, leading to duplication of effort, inconsistencies in representation, and barriers to traceability (Bone et al. 2019). A key motivation for the deployment of digital threads is to eliminate the need for manual replication of information across tools by enabling seamless data integration – allowing each artifact to exist once, in its authoritative source, while still being accessible in context with other lifecycle data (Singh and Willcox 2018). Achieving this vision also requires a common semantic foundation to ensure that data from different tools can be interpreted consistently. Without a shared vocabulary and structure, integration remains superficial, and automated validation or reasoning becomes impractical. Finally, for digital thread data to be usable in practice, it must be delivered through intuitive interfaces (such as dashboards) that enable stakeholders to consume the information they need regardless of the tools in which that information was originally authored. These capabilities collectively underpin a robust and actionable digital thread.

While digital engineering is rapidly gaining traction in aerospace, defense, and other technology sectors, educational programs often lag behind. Students are typically introduced to engineering tools and models in isolation, without a coherent view of how their work relates to the broader system context. Consequently, many graduate with limited exposure to the digital thread, model-based practices, or the semantic underpinnings that enable machine-interpretable integration (Gregory and Salado 2024d). To address this gap, the University of Arizona is developing the digital engineering factory (DEF) – a web-based platform that integrates tools, models, and datasets into a unified environment for systems and software engineering education (Gregory and Salado 2024a). Unlike traditional course management platforms, the DEF emphasizes semantic consistency, automated reasoning, and data-driven feedback. It allows students to experience the end-to-end engineering process through

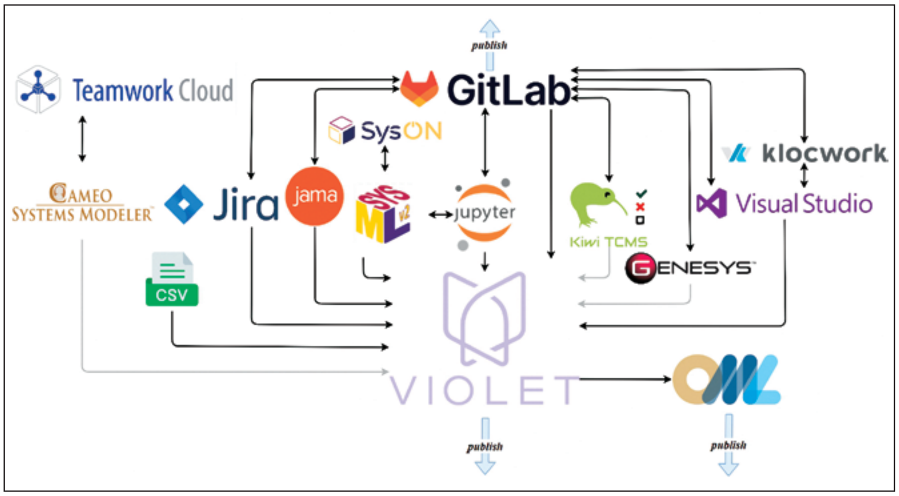


Figure 1. Digital engineering factory (DEF) architecture

authentic, interconnected workflows. This article describes the DEF, its architecture, ontological foundation, deployment in the classroom, and future directions.

WHAT IS THE DIGITAL ENGINEERING FACTORY?

The DEF is designed around a hub-and-spoke architecture (Guntupalli 2023) that connects a suite of engineering tools to a central semantic integration hub. The tools ("spokes") cover various aspects of systems and software engineering, including requirements management, architecture modeling, task tracking, testing, simulation, and analysis. Tools currently integrated include Jama (requirements), Jupyter (SysML v2 modeling), Jira (task management), GENESYS (system architecting), Python-based analyses, and GitLab (version control and CI/CD). These are not simply available in parallel; they are connected through a digital backbone that enables cross-domain relationships and lifecycle traceability.

At the center of the DEF is Violet (the "hub") (Violet Labs 2025). Violet aggregates information from each spoke via APIs, validates it, and transforms it into a semantically rich dataset using the ontological modeling language (OML), developed at the Jet Propulsion Laboratory (JPL) (OpenCAESAR 2025). The result is a knowledge graph that can be queried, reasoned over, and visualized. The DEF also provides users with access to customizable web-based dashboards. This means that users do not have to interact directly with the OML representation of the dataset. Instead, they can interact with the dashboards to display query results in a convenient manner (Gregory, Iyer, and Salado 2025).

This architecture, displayed in Figure 1, allows the DEF to function as more than a collection of tools. It becomes an interoperability layer, ensuring consistency and coherence across engineering domains. A requirement authored in Jama, for instance, can be traced to an architectural component that has been defined using SysML v2, verified by a test procedure in GENESYS, and linked to Jira tasks and Git commits — all semantically grounded and dynamically updated.

ONTOLOGIES AS A FOUNDATION FOR INTEROPERABILITY

The DEF's integration capabilities are underpinned by the University of Arizona ontology stack (UAOS) (Gregory and Salado 2024e). The UAOS is a modular, layered set of ontologies built using OML. It is presented in Figure 2. The stack follows a formal architecture grounded in best practices from ontology engineering:

A **top-level ontology** (TLO), or foundation ontology, provides a domain-independent foundation of general categories and relations (like object, process, part-of) to ensure consistency and interoperability across more specific ontologies (ISO/IEC 2021). By aligning diverse domain models under a shared framework, it supports integration, automated reasoning, and precise communication. The TLO in the UAOS is based on the basic formal ontology (BFO), which defines general categories like continuants, occurrents, and roles (ISO 2021).

Core ontologies sit between top-level ontologies and domain ontologies. They provide reusable, domain-neutral models for common concepts shared across multiple domains – such as events, organizations, measurements, or physical artifacts. The UAOS contains five core ontologies. Four are derived from the common core ontologies (CCO) (Rudnicki 2019) (information, event, agent, measurement), and the provenance ontology is based on the PROV-N notation (Moreau et al. 2013).

Domain ontologies define concepts, relationships, and rules specific to a particular area of knowledge or practice – such as aerospace, medicine, or finance. They are built on top of top-level and/or core ontologies to ensure consistency but focus on capturing the detailed semantics relevant to that domain. The UAOS currently comprises 18 domain ontologies that cover domains such systems architecture, software, project management, and orbital mechanics. These generally conform to existing standards – but in some cases (e.g., verification) we are working to develop our own mathematically rigorous foundations on which to build the ontologies. The modular nature of the UAOS means that we can easily and regularly add other domain ontologies to the UAOS and revise existing ones, depending on what is required by any particular use case.

Each layer builds on the one below it, ensuring that all terms used in DEF

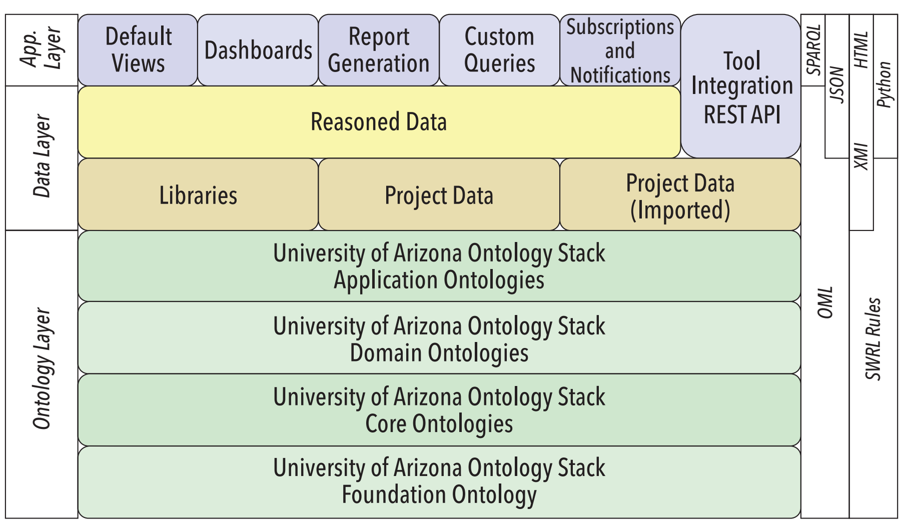


Figure 2. University of Arizona ontology stack (UAOS)

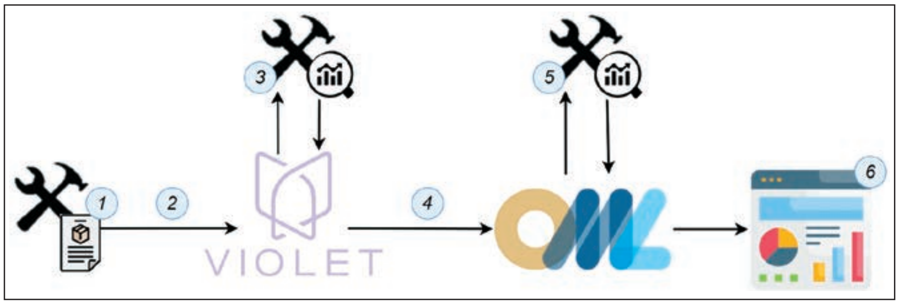


Figure 3. Digital engineering factory (DEF) general workflow. The numbers in blue circles correspond to the steps defined in this section.

datasets are precisely defined and logically coherent. This approach enables automated validation, semantic querying, and inference. These capabilities are critical when managing complex engineering data (Dunbar et al. 2023). For example, a test case is a *subclass* of a planned process, can *verify* a requirement, and can be *performed* by a project team member. Domain ontologies are built on standards wherever possible. The systems architecture ontology, for example, is based on ISO 42010 (ISO/IEC/IEEE 2022), and we are developing a test ontology based on the UML Testing Profile (UTP) (Object Management Group (OMG) 2025). The UAOS ensures that these relationships are logically valid and enforceable. Mistakes, such as assigning a requirement to a test procedure instead of a test case, are caught during validation.

HOW IT WORKS: FROM MODELS TO DASHBOARDS

A typical DEF workflow (displayed in Figure 3) consists of six key stages:

(1) Data Creation
Students create artifacts in various tools: e.g., requirements in Jama, system models in Cameo, tasks in Jira, analyses in Python. Each tool exports data via APIs.

(2) Data Aggregation via Violet
Violet periodically pulls data from each tool and stores it in a structured SQL database. The database includes metadata (timestamps, authorship), relationships (e.g., which requirement verifies which component), and content (e.g., model elements).

(3) Analysis
Users can define analyses to operate on specific subsets of the aggregated dataset (e.g., design parameters, test results) to perform calculations and simulations. These analyses, often written in Python, generate new data products such as performance metrics, compliance scores, or derived attributes. The results are then fed back into the dataset, where they can be traced, queried, and visualized alongside the original data.

(4) Semantic Transformation
Users can then use Violet to perform a complete mapping of the dataset to a graph representation, written in OML and structured in compliance with the UAOS. If any of the data contained in the dataset is not consistent with the UAOS (e.g., a user has accidentally stated that a verification activity is verified by a requirement, instead of the other way around), the reasoner will automatically detect this.

(5) Reasoning and Querying
Using the graph query language SPARQL, users can perform complex queries such as "Which requirements are not satisfied?," "Which components have no verification plan?," or "Which Jira tasks are overdue?" Logical inferences (e.g., transitive satisfaction or derived dependencies) are also applied.

(6) Visualization via Dashboards
A streamlit-based dashboard displays the processed data in user-friendly formats. Dashboards can include tables of requirements, Gantt charts of test schedules, graphs of verification coverage, or interactive rubrics for grading. Users can create and customize their own dashboards by selecting relevant tabs, with each tab displaying the results of a set of predefined queries.

USE CASES: BRINGING DIGITAL ENGINEERING TO LIFE

Systems Engineering Class Projects

As part of the course 'An Introduction to Systems Engineering,' students designed robotic rovers for trash collection (Gregory and Salado 2024b). Each team comprised six members with assigned roles. Students used Jama, SysML v2, Jira, and GitLab to manage their work. The DEF provided automated grading dashboards, which run SPARQL queries based on the course rubric. These queries checked for task completion, requirement coverage, architectural traceability, and verification status. The results gave instructors an objective, up-to-date view of team progress. The resulting dashboard is displayed in Figure 4.

System Design and Analysis

At the University of Arizona, the DEF is not just used in the classroom. The DEF allows researchers to explore novel approaches to digital engineering. One of the features we have demonstrated is the ability to link architecture models to Python-based analyses. In Gregory et al. (2025), for example, we demonstrated how we can use the DEF to connect spacecraft architecture models, orbital models and requirements to simulate the orbital decay of the spacecraft and verify a mission requirement. Data from all tools was harmonized through Violet and rendered as a unified knowledge graph. The dashboard visualized verification status,

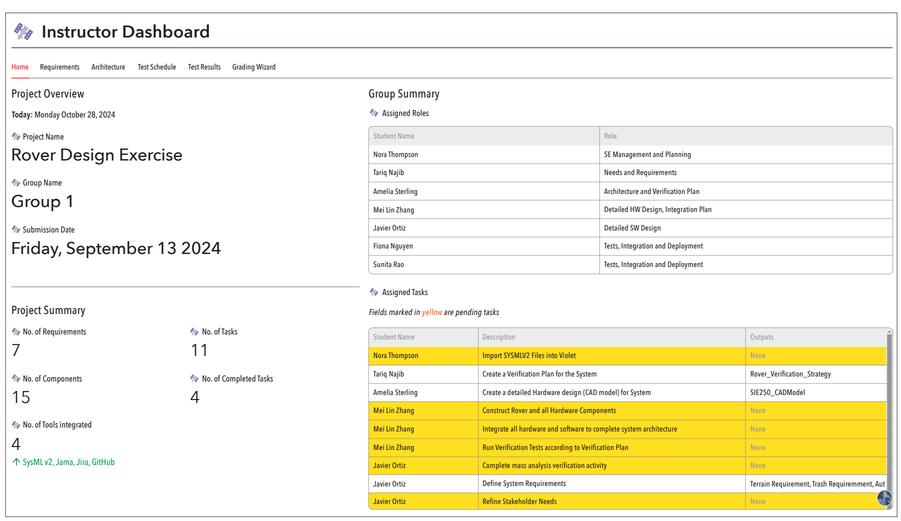


Figure 4. Instructor dashboard – illustrating summary of student project submitted as part of 'An Introduction to Systems Engineering'

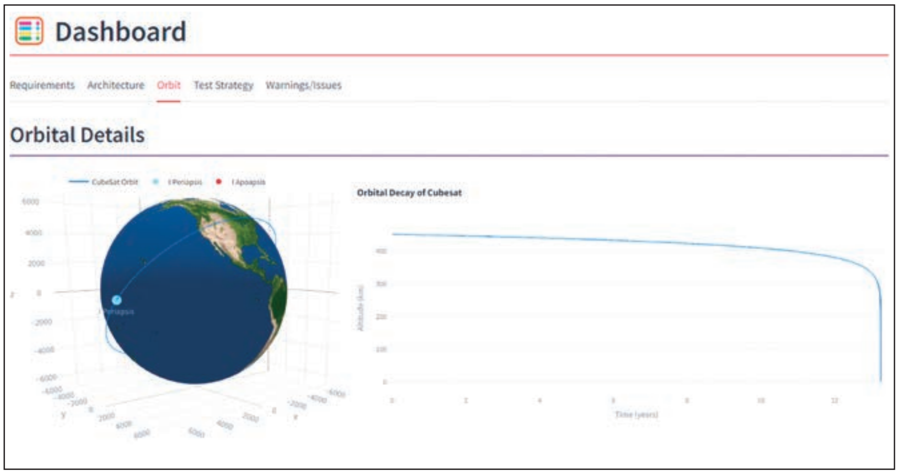


Figure 5. 'Orbit' tab on DEF dashboard – illustrating results of deorbit analysis. Reproduced from Gregory et al. (2025)

mass allocation, and unmet requirements in real time, and is presented in Figure 5.

Digital Test and Evaluation

As researchers, we also use the DEF and the UAOS to explore novel approaches to digital test and evaluation (T&E). For example, we have leveraged the reasoning and querying capabilities afforded by the ontologies to generate Bayesian network representations of test strategies (Gregory, Jackson, and Salado 2025). We have also used the DEF to model a Department of Defense-style test and evaluation master plan (TEMP) and check for resource conflicts, overlapping events, and missing preconditions (Gregory and Salado 2024c). In this example, the user imported definitions of test cases, equipment, etc. into Violet from SysML v2 models. The user then specified the order in which these 12 test cases were to be carried out. The total duration of the test strategy was then calculated, and a summary of the test strategy was presented on the dashboard. The information presented to the user highlighted some issues with the proposed test strategy – some of the necessary equipment was not available for two of the test cases, and the duration of the test strategy would exceed the upper limit of 60 days. The user then updated the SysML models to include the necessary equipment and updated the test case schedule in Violet. The updated dashboard shows that these changes resolved the issues that had been detected and presented in the dashboard. The corresponding dashboards are displayed in Figure 6. This exercise demonstrated how semantic reasoning can be applied to logistical and operational problems.

Figure 6. 'Test Strategy' tab on DEF dashboard. Top: The dashboard detects issues with the test strategy. Bottom: The dashboard shows updated results after the user corrected the issues in SysML v2 and Violet. Reproduced from Gregory et al. (2025)

LESSONS LEARNED AND FUTURE PLANS

The DEF has proven valuable from the perspective of students, instructors, and researchers – but work in this area has also revealed areas for improvement:

Students benefit most when they don't see the ontology. The power of the UAOS is in what it enables: real-time feedback, consistency checks, and intelligent dashboards. The complexity that it introduces is best kept behind the scenes, particularly for students. The dashboard design must abstract away semantic formalisms while preserving the benefits of formal reasoning.

Push-back capabilities are essential. Currently, Violet pulls data into a semantic representation, but writing changes back into tools (e.g., updating a Jira ticket or modifying a requirement in Jama) is only partially supported. Developing and integrating these "write" paths into the workflow is a key objective for future iterations.

Setup must be smoother. Tool integration currently involves some manual configuration, which can be a barrier for new users. Future versions of the DEF will focus on containerization, automated setup scripts, and a centralized user intereface (UI) for managing connections and mappings.

Scalability is within reach. The DEF has already been tested in classrooms with up to 40 students. With improved automation and cloud hosting, it can support larger deployments and even multi-institution collaborations – but this needs to sufficiently tested.

CONCLUSION

The digital engineering factory (DEF) represents a new approach to engineering education: one that aligns with modern industry practices, emphasizes semantic integration, and empowers both students and instructors through data. By unifying disparate tools under a common ontological framework, the DEF offers a powerful demonstration of what the digital thread can look like in the classroom.

More than a proof-of-concept, the DEF is a working platform. It is currently used in real courses, by real students, producing real engineering artifacts. It teaches not just tool usage, but systems thinking, model-based reasoning, and the value of data integrity. As digital engineering becomes the norm in complex system development, platforms like the DEF will be critical in preparing the next generation of systems engineers. ■

REFERENCES

- AIAA. 2023. "Digital Thread: Definition, Value and Reference Model." *American Institute of Aeronautics and Astronautics (AIAA) Digital Engineering Integration Committee (DEIC)*, no. June.
- Bone, Mary A., Mark R. Blackburn, Donna H. Rhodes, David N. Cohen, and Jaime A. Guerrero. 2019. "Transforming Systems Engineering through Digital Engineering." *Journal of Defense Modeling and Simulation* 16 (4): 339–55. https://doi.org/10.1177/1548512917751873.
- Dertien, Steve, and Will Hastings. 2021. "The State of Digital Thread." *PTC White Paper*.
- Dunbar, Daniel, Thomas Hagedorn, Mark Blackburn, John Dzielski, Steven Hespelt, Benjamin Kruse, Dinesh Verma, and Zhongyuan Yu. 2023. "Driving Digital Engineering Integration and Interoperability through Semantic Integration of Models with Ontologies." *Systems Engineering*, no. January: 1–14. https://doi.org/10.1002/sys.21662.
- Gregory, Joe, Visalakshi Iyer, Andres Galvez, Alejandro Salado, Lucy Hoag, Alisha Zute, Mike Marmar, and Barry Jones. 2025. "Digital Engineering for Cubesats with Semantic Web Technologies." In *AIAA SciTech*, 1–13. Orlando, US-FL. https://doi.org/10.2514/6.2025-1656.
- Gregory, Joe, Visalakshi Iyer, and Alejandro Salado. 2025. "Semantically Enabled Dashboards to Support Systems Engineers." In *INCOSE International Symposium*, 1–17. Ottawa, CA.
- Gregory, Joe, Bennett Jackson, and Alejandro Salado. 2025. "Automated Generation of Bayesian Verification Strategies Using Semantic Web Technologies." In *Conference on Systems Engineering Research (CSER)*, 1–10. Long Beach, US-CA.
- Gregory, Joe, and Alejandro Salado. 2024a. "A Digital Engineering Factory for Students." In *Conference on Systems Engineering Research (CSER)*. Tucson, US-AZ.
- ———. 2024b. "Digital Engineering Factory (DEF): Connecting Systems Engineering Artifacts in the Classroom." In *SECESA*, 25–27.
- ———. 2024c. "DTEMP: From Digitizing to Modeling the Test and Evaluation Master Plan." *Naval Engineers Journal* 136 (1): 134–46.
- ———. 2024d. "Quantifying the Ability of the Digital Engineering Factory to Address the Digital Engineering Competency Framework." In *American Society for Engineering Education (ASEE) Conference*. Portland, US-OR.
- ———. 2024e. "Towards a Systems Engineering Ontology Stack." In *INCOSE International Symposium*. Dublin, IE: INCOSE.
- Guntupalli, Bhavitha. 2023. "ETL Architecture Patterns: Hub-and-Spoke, Lambda, and More." *International Journal of AI, BigData, Computational and Management Studies* 4 (3): 61–71. https://doi.org/10.63282/3050-9416.ijaibdcms-v4i3p107.
- ISO/IEC/IEEE. 2022. "ISO/IEC/IEEE 42010:2022 Software, Systems and Enterprise — Architecture Description." https://cdn.standards.iteh.ai/samples/74393/fc7b7f103d8446a4b87a3261e31370d3/ISO-IEC-IEEE-42010-2022.pdf.
- ISO/IEC. 2021. "Information Technology — Top-Level Ontologies (TLO) — Part 1: Requirements" 2021: 23. https://www.iso.org/standard/71954.html.
- ISO. 2021. "ISO 21838-2: Information Technology — Top-Level Ontologies (TLO) — Part 2: Basic Formal Ontology (BFO)" 2010.
- Moreau, Luc, Paolo Missier, James Cheney, and Stian Soiland-Reyes. 2013. "PROV-N: The Provenance Notation." *W3C Recommendation*, no. April: 1–34. http://www.w3.org/TR/2013/REC-prov-n-20130430/.
- Object Management Group (OMG). 2025. "UTP2 - UML Testing Profile 2." https://www.omg.org/spec/UTP2.
- OpenCAESAR. 2025. "Ontological Modeling Language 2.0.0." https://www.opencaesar.io/oml/.
- Rudnicki, Ron. 2019. "An Overview of the Common Core Ontologies." *CUBRC Report*.
- Singh, Victor, and Karen E. Willcox. 2018. "Engineering Design with Digital Thread." In *AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, 2018*, 1–21. https://doi.org/10.2514/6.2018-0569.
- Violet Labs. 2025. "Violet." 2025. https://www.violetlabs.com/.

ABOUT THE AUTHORS

Dr. Joe Gregory is a postdoctoral research associate at the University of Arizona. His research interests include digital engineering, model-based systems engineering, and semantic web technologies. His current research focuses on the development of the digital engineering factory, and the development of ontologies to support digital test and evaluation. He is the co-chair of the Digital Engineering Information Exchange (DEIX) Ontology Working Group.

Dr. Alejandro Salado has over 15 years of experience as a systems engineer, consultant, researcher, and instructor. He is currently an associate professor of systems engineering with the department of systems and industrial engineering at the University of Arizona and the director of its systems engineering programs. Dr. Salado holds a BS/MS in electrical and computer engineering from the Polytechnic University of Valencia, a MS in project management and a MS in electronics engineering from the Polytechnic University of Catalonia, the SpaceTech MEng in space systems engineering from the Technical University of Delft, and a PhD in systems engineering from the Stevens Institute of Technology.

Accelerating Digital Engineering Adoption: A Comprehensive Example Using MBSE and Digital Twin with a Portable Robotic Arm

Saulius Pavalkis, Saulius.Pavalkis@3ds.com; and **Mariah Otte,** Mariah.Otte@3ds.com

■ **ABSTRACT**
Digital engineering (DE) promises faster, more reliable product development—but only when its benefits are tangible to practitioners. This concise 5-page paper distills a years-long applied research effort at Dassault Systèmes into a portable showcase: a five-axis Arduino-based robotic arm modelled, simulated, manufactured, and verified through a single digital thread. Using SysML-based model-based systems engineering (MBSE) (MagicGrid method), Modelica multiphysics, robotic simulation, FMI co-simulation, and MQTT-enabled hardware-in-the-loop, we demonstrate how requirements trace directly to architecture, mechanics, electronics and code. The result is a replicable template for universities and industry teams seeking to adopt digital engineering (DE) with minimal cost and maximum pedagogical impact.

INTRODUCTION

Complex products—from eVTOL aircraft to autonomous farm equipment—demand that multidisciplinary teams collaborate on authoritative models rather than brittle documents.

Yet many organizations still struggle to justify model-based systems engineering (MBSE) (SEI 2020) and digital twin investment because examples are either too abstract (toy problems) or too expensive (million-dollar testbeds). For example, Chris Schreiber, LMCO systems engineering modernization senior manager in 2019 said "... beyond simply capturing systems engineering design, to enabling a better way of engineering systems."

Our goal is therefore to create a portable, affordable (~$60 USD) demonstrator that covers the full V-model—from stakeholder need to operational diagnostics—within tools already familiar to aerospace, defense, and industrial clients. The portable robotic arm (Adeept 2023) offers enough typical modern systems characteristic: kinematic (5 DOF, serial servos, diverse loads), software, electrical, and mechanical design to exercise

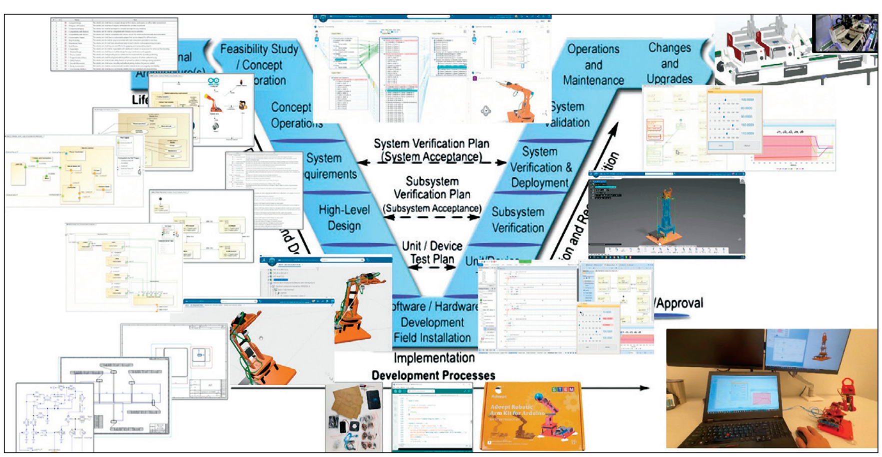


Figure 1. Full engineering lifecycle

real engineering trade-offs and other use cases while remaining classroom-friendly.

INTEGRATED ENGINEERING WORKFLOW

In the proposed project we integrate the following engineering disciplines and tasks in a model based way: requirements management, systems engineering, electrical schematics, mechanical design 3D CAD, 3D wire harness, model (OMG SysML 1.7) to code generation (C and target platform – Arduino), configuration management – baseline, traceability, hardware-in-the-loop, Multiphysics simulation, robotic simulation, manufacturing and assembly, co-simulation, verification and validation (V&V), operational monitoring and diagnostics. This demonstrates the engineering lifecycle covered with sample models and integrated through baseline, traceability, and co-simulation (Figure 1).

MBSE TO TRANSITION FROM STAKEHOLDER NEEDS TO SOLUTION ARCHITECTURE AND REQUIREMENTS FOR DESIGN

MBSE enables transitioning from stakeholder needs to system requirements, optimal solution architecture and requirements for design. We are applying OMG systems modeling language SysML, and MBSE OOSEM based method MagicGrid (MagicGrid). MagicGrid based on ISO 15288 standard (ISO/IEC/IEEE 15288) give predictable innovation driven transition from stakeholder needs to optimal solution architecture to physical requirements for design (Figure 2).

The system scenario starts with stakeholder needs which can come from external dedicated requirements tool or captured directly in system architecture model. From user requirements we transition to MBSE using SysML and MagicGrid.

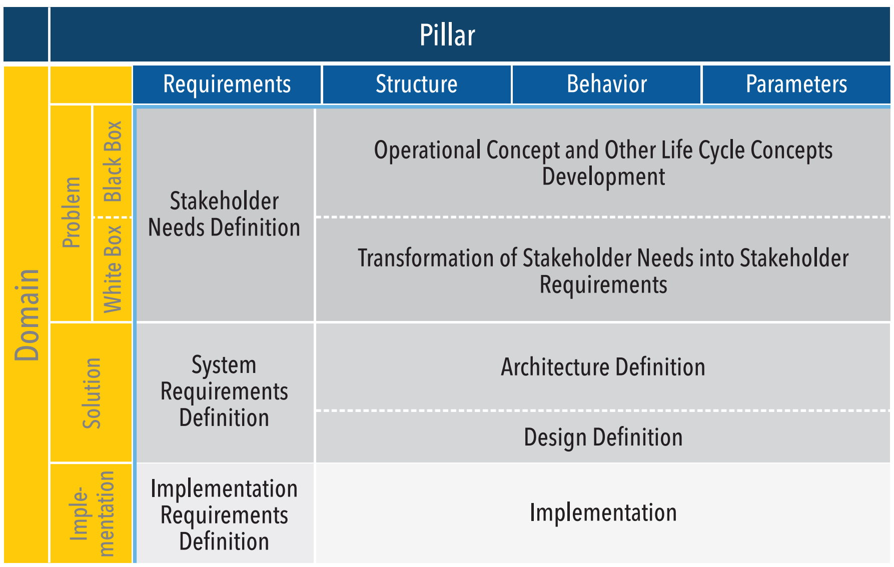


Figure 2. MBSE method–MagicGrid to transition from stakeholder needs to solution architecture to physical requirements for design

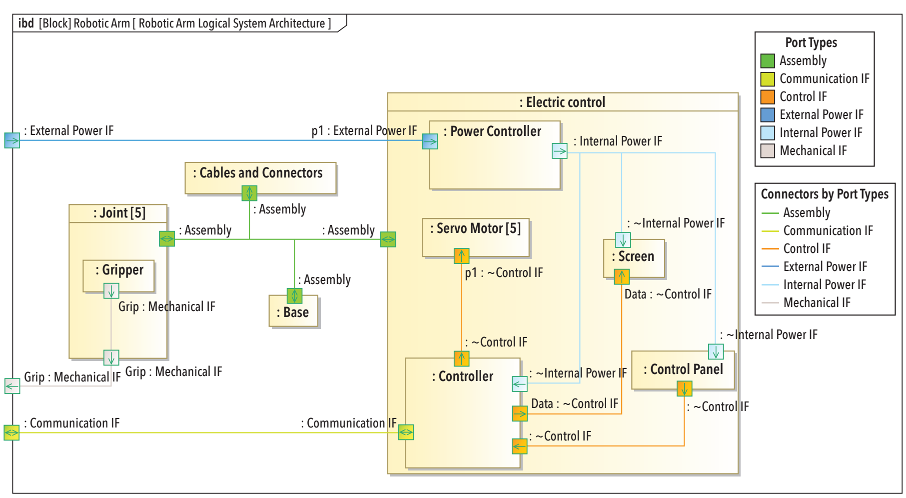


Figure 3. Logical system architecture

It is important to note, a large part of the system environment where the system exists is a digital environment where all models coexist as digital twins for rapid system change, configuration, testing, and shipment to market.

Logical system architecture (Figure 3) is grouping system functions as result of use case based functional system analysis. It is oriented to what is flowing and what is the system doing — not how. It is smallest single architecture of the system. Logical architecture is based on stakeholder needs, a precondition for clear optimal system requirements.

Solution architecture (Figure 4) is the result of trade study analysis based on functions, interfaces and optimized on measures of effectiveness (MoEs). Solution architecture is oriented into physical interfaces and components. It is the primary model to communicate with designers.

From optimal solution architecture we transition to design and requirements for software, mechanical, electrical design.

SYSTEM DESIGN

System design consists of co-engineering of discipline design around the solution architecture. It includes: electrical schematics, mechanical design–3D CAD, 3D wire harness, model to code generation (C and robotic target platform), configuration management–baseline, hardware in the loop, multiphysics simulation, robotic simulation, manufacturing and assembly, co-simulation, V&V, operational monitoring and diagnostics. The key objective is to create a complete digital thread with traceability links to connect design artifacts from stakeholder needs to detailed implementation design. Systems traceability (Figure 7) contributes to systems engineering providing clear stakeholder needs impact on system design, rapid system reengineering, and change impact analysis.

Mechanical design (Figure 5) is a precondition to: structural analysis, 3D wire harness, manufacturing, styling, and many other design disciplines. System structural, behavioral, and parametric requirements have a direct impact on mechanical design.

The requirements, functional, logical, and physical layers (RFLP) approach provides a metamodel and common method to how design data is organized in the product lifecycle management (PLM) based application. RFLP was introduced in CATIA V6 for digital continuity (Kleiner2013). Requirements, functional, logical, and physical layers provide a comprehensive structure for system design and implementation. RFLP roots are in systems engineering; they originated to manage complexity and ensure traceability.

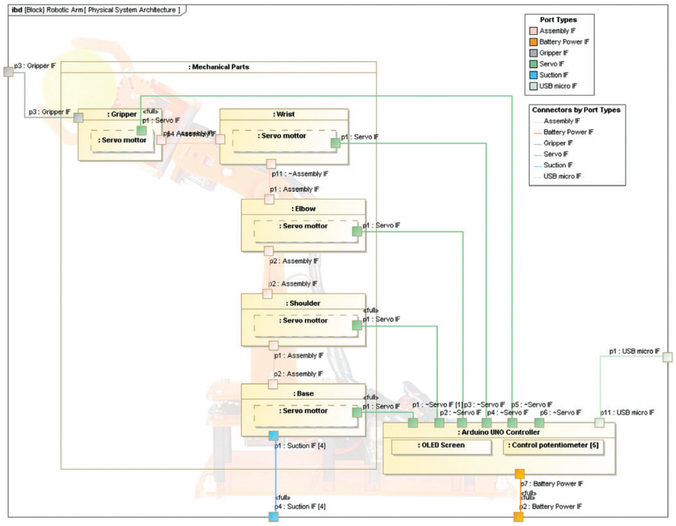


Figure 4. Solution system architecture

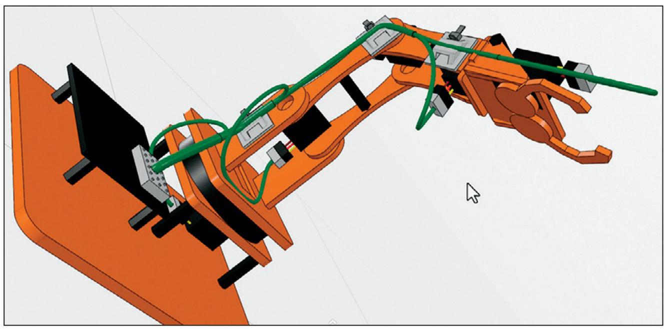


Figure 5. Mechanical design–3D CAD

Electrical Design – Schematics and 3D Wire Harness Design. The main purpose of electrical design is to design network schematics connecting equipment at the pin level and define signals flowing from one pin to another. MBSE provides power and precision requirements and also test cases.

The electrical designer transitions from schematics to 3D wire harness design by creating routed harness segments in 3D covering: electrical 3D design, electrical 3D part design, and electrical manufacturing preparation. The electrical and manufacturing engineer performs flattening of the 3D wire harness for manufacturing Figure 6, generating absolute or algorithmic scale full product specification.

DIGITAL THREAD, TRACEABILITY, AND BASELINE

For the data to be integrated we ensure configuration and baseline management. We need to provide lifecycle management, revision tracking, and access control for data integrity. We need enable access to design data and support real-time updates by multiple users, ensuring data consistency.

To ensure systems traceability we need to connect design artifacts from stakeholder needs to detailed implementation design. Create a complete digital thread with traceability links for change impact analysis. Integrate diverse applications like Catia Magic (Cameo), MATLAB/Simulink, 3D CAD (MultiCAD), MS Office, PDF, DOORS, and Jama Software. Establish a single source of truth accessible across disciplines. As result we get: full traceability enabling cross design and analysis digital thread, change impact analysis, review and collaboration.

In our case, the digital thread (AIAA 2023) includes: requirements, system architecture, electrical schematics, mechanical design–3D CAD, 3D wire harness, model to code generation (C and robotic target platform), hardware-in-the-loop, multiphysics simulation, robotic simulation, manufacturing and assembly, co-simulation, V&V, operational monitoring, and diagnostics. Traceability (Figure 7) contributes to systems engineering providing clear stakeholder needs impact on system design, rapid system reengineering, and change impact.

DIGITAL VERIFICATION AND VALIDATION THROUGH CO-SIMULATION

Co-simulation (Figure 8) leverages open standards: SysML, Modelica, FMI, MQTT, and other OOTB connections integrating various application in a modular way.

- SysML – system architecture, logic, requirements. Execution of system behavior profile, set limits on joints configuration, track requirements compliance. System logic in SysML simulation and requirements verification is performed using SysML execution.
- Modelica – Multiphysics simulation. Keep track of each joint voltage, current, torque. Multiphysics analysis of system behavior is performed based on custom and predefined libraries for system sizing, what if scenarios, and system verification.
- Visual robot simulation. This includes design, control, and virtual robot testing. It is performed to predefine robot behavior and ergonomics analysis, identify boundaries, and perform system V&V.
- Hardware-in-the-Loop – physical prototyping, monitoring.

MODEL BASED DIGITAL ENGINEERING ROI

This project allowed us to demonstrate the major **return on investment (ROI)** of model based digital engineering – rapid system update. Changes in stakeholder needs for longer arm reached was evaluated for impact and updated in final specifications in minutes. This included:

- Impact analysis using traceability.
- Collaboration creating issues in digital environment.
- Automatic parametrized models update from requirement to mechanical design, to recalculated wire harness, to produced

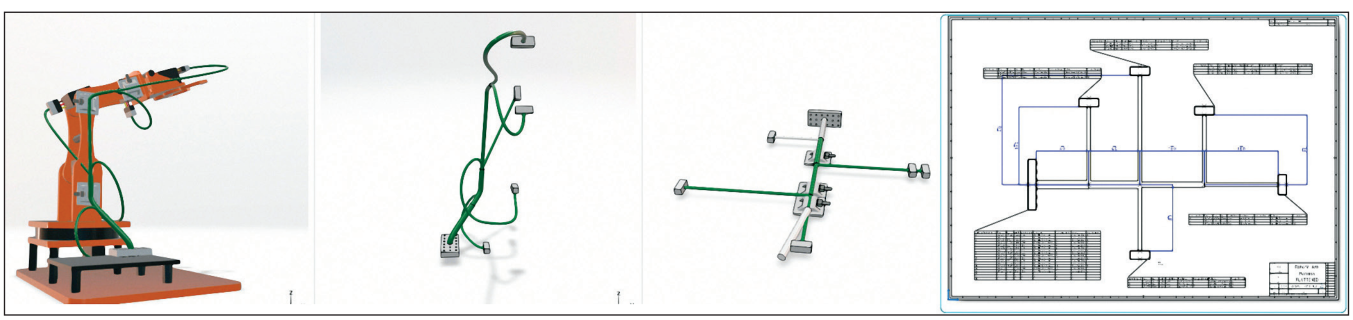


Figure 6. Transition from 3D wire harness design to manufacturing

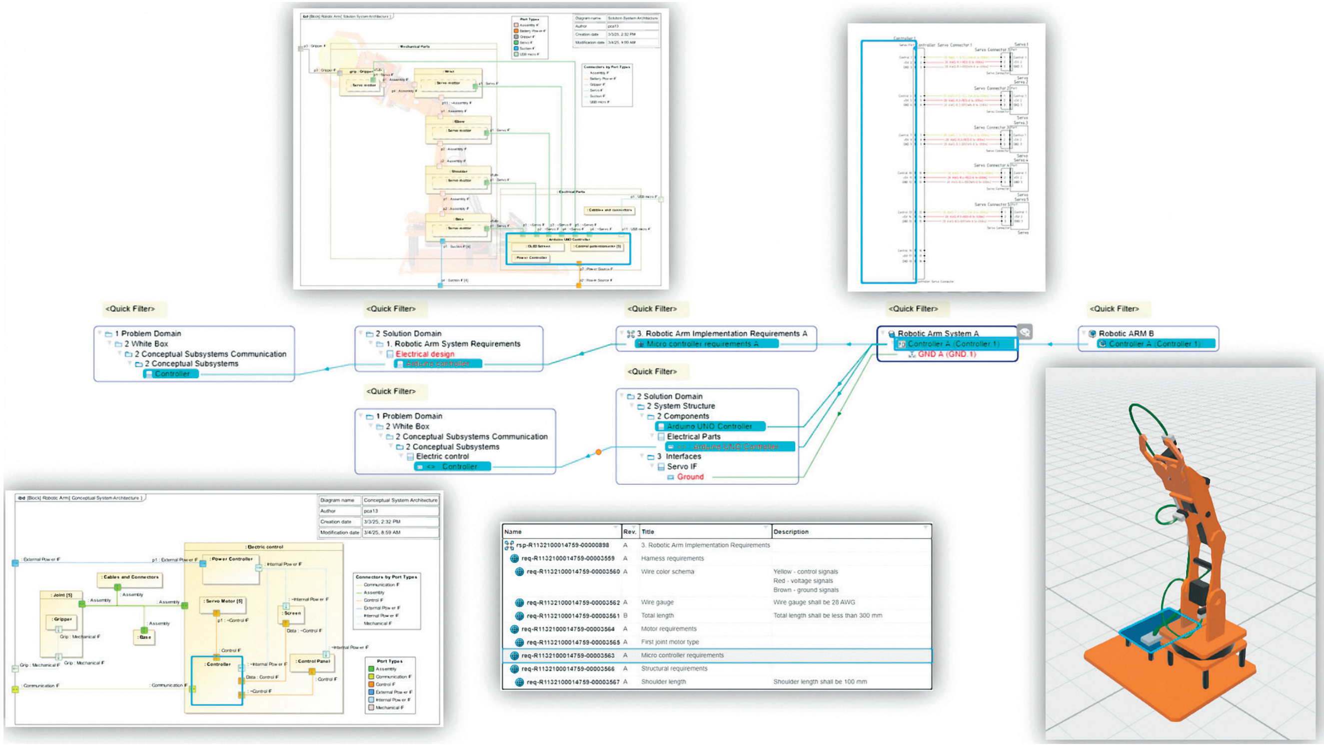


Figure 7. End-to-end traceability from stakeholder needs to system architecture, design, manufacturing, analysis, and V&V

final drawing for manufacturing.

- V&V through co-simulation testing updated arm reach covering power usage testing with Dymola, collisions detection with Robot Simulation by DELMIA.

CONCLUSIONS

This article demonstrates an example for model based digital engineering adoption. It is a compact project that highlights major design tasks and connectivity through a digital thread, traceability, and co-simulation. The project leverages open standards SysML, Modelica, FMI, IoT – MQTT. It is highly modular enabling to connect new solutions and replace existing ones. We believe that adoption of such data enables the most rapid digital engineering adoption by academia and industry. This project allowed us to demonstrate the major **ROI** of model based digital engineering — rapid system update. ■

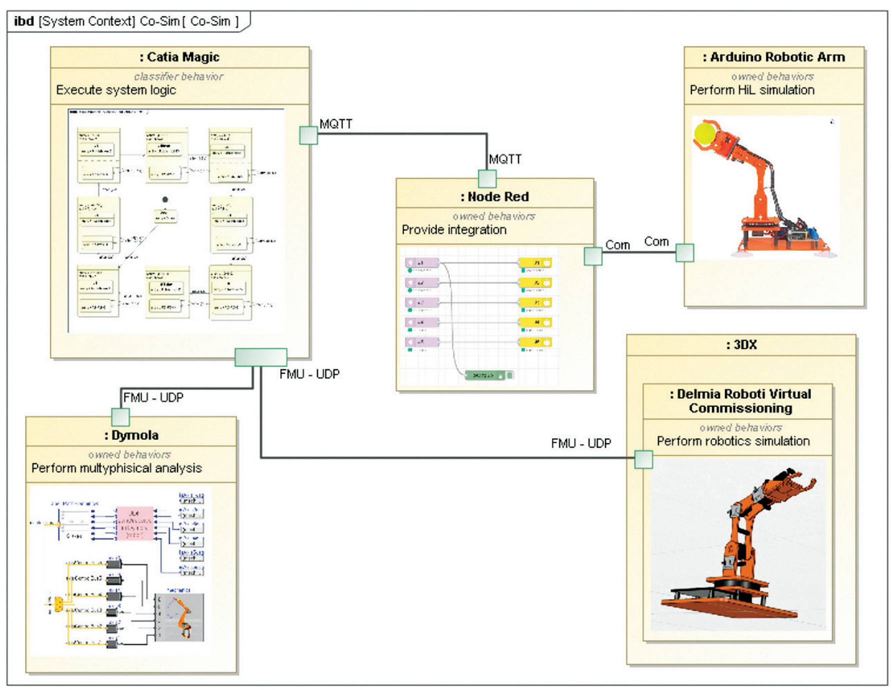


Figure 8. Real time co-simulation: robotic, multiphysics system

REFERENCES

- Adeept. 2023. "5-DOF Robotic Arm Kit Compatible with Arduino IDE – Programmable DIY Coding STEM Educational 5-Axis Robot Arm with OLED Display & Processing Code." https://www.adeept.com/adeept-5-dof-robotic-arm-kit-compatible-with-arduino-ide-programmable-diy-coding-stem-educational-5-axis-robot-arm-with-oled-display-processing-code_p0118.html/. (Accessed November 2024).
- AIAA. 2023. "Digital Thread: Definition, Value and Reference Model." AIAA, AIA, NAFEMS.
- DAU (Defense Acquisition University) n.d. https://www.dau.edu/glossary/digital-engineering?utm_source=chatgpt.com.
- ISO/IEC/IEEE 15288. 2023. Systems and Software Engineering System Life Cycle Processes.

- Kleiner, Sven, and Christoph Kramer. 2013. "Model-Based Design with Systems Engineering based on RFLP using V6." Smart Product Engineering: Proceedings of the 23rd CIRP Design Conference, Bochum, DE: Springer Berlin Heidelberg, 11-13 March.
- MagicGrid. n.d. MagicGrid Body of Knowledge. https://discover.3ds.com/magicgrid-book-of-knowledge.
- OMG SysML1.7. n.d. System Modeling Language specification version 1.7. Object Management Group. https://www.omg.org/spec/SysML/1.7.
- SEI. 2020. "An introduction to Model-Based Systems Engineering (MBSE)." Software Engineering Institute, Carnegie Mellon University, Pitsburgh, US-PA. Nataliya Shevchenko. https://insights.sei.cmu.edu/blog/introduction-model-based-systems-engineering-mbse/.

ABOUT THE AUTHORS

Saulius Pavalkis, ESEP, PhD is global MBSE ecosystem senior director. He has 20+ years of MBSE solutions experience, as MBSE transformation leader, product owner, and analyst at Cameo core team, chief solutions architect, consultant, and trainer. He contributed to MBSE SysML based MagicGrid method and framework and, more recently, *Agile MBSE Cookbook* by Bruce Douglas. Saulius guides 6000+ subscribers, the largest SysML systems simulation community (youtube.com/c/MBSEExecution).

Mariah Otte is a solution architect on the aerospace and defense industry team at Dassault Systèmes, specializing in finding the right solutions for aerospace customers looking to leverage virtual twins.

She began her career at Dassault as a robot simulation and programming expert, successfully integrating multiple digital twins with real-world work cells. Her focus then expanded to digital manufacturing, exploring how virtual factories can model and plan production processes. Currently, Mariah defines comprehensive solutions tailored to customer needs, with a focus on manufacturing and aftermarket processes.

Helping Organizations Adopt Digital Engineering in a Mature and Sustainable Way with DE CMAF

Michael Shearin, michael.shearin@gtri.gatech.edu; **Dr. Valerie Sitterle,** vsitterl@stevens.edu; and **Dr. Owen Eslinger,** owen.j.eslinger.civ@army.mil

ABSTRACT
Organizations seeking to transition to digital engineering (DE) practice face a significant challenge in their need to understand what capabilities are necessary to achieve their DE goals and how these capabilities should mature over time. Further, an organization's DE goals are driven by their specific organizational responsibilities and gaps they may have across multiple dimensions: DE environments and infrastructure, workforce development and skills, workflows, DE practice, and time. The DE capability maturity and assessment framework (CMAF) helps an organization identify desired capabilities, required maturity levels on a specified timeline, and any gaps across all five DE capability areas. The DE CMAF was developed for and through support of the US Army, intended generally for US Department of Defense (DoD) organizations, and used thus far by the US Army, Navy, and Air Force. However, its design and content are extensible and applicable to all organizations.

KEYWORDS: DE; digital engineering; maturity; assessment; adoption; workforce; development; environment; ecosystem

1. OVERVIEW

To achieve a digital engineering (DE) transformation, organizations need to understand what capabilities are necessary to achieve their DE goals, what gaps they have with respect to those capabilities, and how maturation of these capabilities can close those gaps. Each must be understood in the context of their organization's responsibilities and will span across DE environments and infrastructure, workforce development and skills, workflows, DE practice, and time. Organizations face the problem of knowing what to do and where to focus efforts to begin or mature their transformation to full DE practice. The DE capability maturity and assessment framework (CMAF) was created for the Assistant Secretary of the Army (Acquisition, Logistics and Technology) (ASA(ALT)) to help address this challenge for organizations of any size or level. It serves as an organizing, integrating framework that provides guidance for organizations to evaluate and advance their practice of DE. Derived from DoD guidance (DoD Strategy 2018) (DoD DE Ecosystem Requirements 2022) and using the conceptual structure of INCOSE's model-based capability matrix (INCOSE 2020) as a foundation, the DE CMAF captures very specific elements of DE practice, including technical and systems engineering concerns. Capabilities are expressed to a level of detail that organizations can use the framework to create a roadmap for maturation. The framework is inherently tailorable, allowing organizations using the framework to customize requirements, strategy, and even process capabilities as needed.

2. MOTIVATION: USE CASES DRIVING REUSE AND DISCOVERABILITY

2.1. Problem Statement

The US DoD aims for transformation of technical and systems engineering practices to a digital, model-based form across all components. This transformation is not, however, binary. Consider the dimensions at play in *DoDI 5000.97 – Digital Engineering* (DoD 2023), which describes a DE environment in the context of DAU's definition of a DE ecosystem:

> *"A digital engineering ecosystem may include, but is not limited to, government-to-government, contractor-to-government, and contractor-to-supplier digital collaboration. These collaborative digital environments are key to involving all stakeholders in developing models,*

Source	Reason for use	Concepts of policy or guidance activities or capabilities needed at enterprise level	Concepts of maturity (i.e., to what level are capabilities being achieved)	Explicit DE ecosystem requirements from tooling, interoperability, and infrastructure or services perspectives
2018 DoD Digital Engineering Strategy	**The "why"**: Primary source for direction and action	Strong coverage	Minimal or no coverage	Moderate coverage
INCOSE Model-Based Capability Matrix (MBCM)	**The "what"**: Workflow and organization concepts. **The "how well"**: Primary source for maturity concepts.	Moderate coverage	Strong coverage	Moderate coverage
OUSD(R&E) DE Ecosystem Requirements	**The "how"**: Only source with solid focus on tooling and infrastructure (does not cover concepts outside of the DE Environment).	Minimal or no coverage	Minimal or no coverage	Strong coverage

Strong coverage · Moderate coverage · Minimal or no coverage

Figure 1. DE CMAF foundation sources and what they cover

executing simulations, and performing analysis and optimizations for the digital models or digital twins. In some instances, customers, regulators, contractors, suppliers, or operators must be integrated into the digital engineering ecosystem to complete the digital thread."

Organizations seeking to transition to DE must clearly understand (a) where they are currently, (b) where they need to focus in terms of DE capability maturation, and (c) how well they are doing as they embark on this journey across these multiple dimensions.

2.2. Use cases – Helping Organizations Evaluate and Plan their DE Transformation

The DE CMAF provides a meaningful tool for assessment of DE capabilities grounded in guidance. During its development, the team identified several use cases highly relevant to organizations seeing DE adoption and transformation:

1. Begin a DE transformation – Understand what capabilities to consider and in a way that allows the organization to prioritize in line with their unique role and objectives;
2. Assess a DE transformation – Determine where the organization is in a DE transformation, discover existing capability gaps, and determine where to focus next;
3. Define a DE transformation roadmap – Define the organization's custom DE capability categories and levels of maturity needed to accomplish business objectives:
    a. Structure and identify priorities for capture in a request for proposal (RFP),
    b. Evaluate profiles for vendor response to an RFP or define content for a performance work statement (PWS).

3. BUILDING BLOCKS OF MATURITY ASSESSMENT: STRATEGY, PROCESSES, AND REQUIREMENTS

3.1. Overview of Sources

Three cross cutting concerns gave structure to the framework: concepts of policy or guidance for capabilities, concepts of maturity, and high-level technical requirements. Accordingly, the DE CMAF is derived from three types of sources that together address these concerns as shown in : (i) the 2018 DoD DE strategy guidance for the strategic "why" aspect of organizational DE strategy as well as workforce training and a culture of practice, (ii) INCOSE's MBCM for the process-based "what" and fundamental maturity concepts to plan or assess in a structured way and measure progress, and (iii) DoD DE ecosystem requirements for the technical and logical environment structure necessary, including general tooling and interoperability concepts. When taken together, these sources and cross cutting concerns resulted in a collection of specific, actionable elements that can be commonly assessed for maturity and can be used to create a roadmap for implementation.

4. IMPORTANT CONCEPTS FOR EFFECTIVE USE

The DE CMAF is not an all or nothing or an immutable framework. It contains specific activity descriptions, sorted and grouped according to main areas of emphasis, but not all activity lines may be important to a given organization or its responsibilities. It is structured so an organization can customize, selecting and assessing only activities important to its needs. It is tailorable by design as not all organizations need to reach the highest maturity level in any given activity and certainly not in all of them. DE CMAF activity lines do not state how an organization should meet goals, only what types of things must be accomplished to reach maturity levels for each. In this way, it aims to be actionably specific and comprehensive as a capability guidance and maturity assessment framework for figuring out where you are as an organization, where to go based on organization priorities, and how well you are doing (i.e., maturity assessment) with respect to critical elements of DE practice. The DE CMAF is designed for clarity and intuitive ease of use to create actionable insight (Figure 1).

5. EXCEL WORKBOOK-BASED CMAF: HOW IT'S ORGANIZED

The Excel workbook version of the DE CMAF is a lightweight interface created to facilitate ease of use across a wide assortment of users. The workbook sheets and their respective purposes are summarized in Table 1. All three DE CMAF sources are included in their entirety for reference for detail about context as needed activity.

5.1. DoD DE Strategy Decomposition leading to Specific Activity Description

Each DE activity sheet starts by defining a three-part organizational structure, which makes DE CMAF entries sortable and discoverable across 3 levels of granularity:

1. High-level capability category, aligned to one of the 5 DoD DE strategy main goals.
2. Focus area, aligned to one of the 14 DoD DE strategy focus areas (goal components).

Table 1. DE CMAF Excel workbook sheets and their purposes

Sheet	Purpose
Explanation	High-level overview of the DE CMAF, its sources, structure, and usage intent
Dashboard	Complete set of capability profile charts from the main assessment groups derived from the 2018 DoD DE Strategy (i.e., corresponding to goals 1 through 5)
Gs1-2 DE DEV AND EXEC	DE Activity assessment worksheet: Focused on DE capability development and execution (2018 DoD goals 1 and 2)
G3 TECH INNOVATION	DE Activity assessment worksheet: Focused on technological innovation for advancing DE practice (DoD Goal 3)
G4 DE ECOSYS INFR	DE Activity assessment worksheet: Focused on DE Ecosystem infrastructure and capability needs to support effective DE practice (DoD goal 4)
G5 WORKFORCE DEV	DE Activity assessment worksheet: Focused on workforce development, including identifying needs, training, measuring, and collaborating (DoD goal 5)
G5 ORG TRANSFORM	DE Activity assessment worksheet: Focused on organizational transformation, especially higher-level leadership, policy, or guidance needs to support effective DE practice (DoD goal 5)
BASE-2018 DoD DE Strategy-14pts	The complete set of 2018 DoD DE Strategy goals decomposed: 5 main goals, 14 goal components, specified sub-components
BASE-INCOSE MBCM-SORT ID	The complete INCOSE MBCM
BASE-OUSD(R&E) Ecosys Reqts	The complete DoD OUSD(R&E) DE Ecosystem Requirements list (12/2022 draft)

Table 2. Example selection from DE CMAF showing organizational structure

Major Goal Area of Applicability (Aligned with DoD DE Strategy top-level goal, i.e., goals 1 thru 5)	*(Aligned with Goal Focus Areas as numbered in DoD DE Strategy as Goal.n — e.g., 5.1)*	*(Aligned with Focus Area Components as described in text under each numbered Focus Area in the DoD DE Strategy)*	*This is the "What you need to do" description. Here, this is based on the sub-component activity description from the cited DoD Strategy portion.*
Capability Category	**Focus Area**	**Sub-Component (Activity)**	**CMAF Activity Description**
Development, Integration, and Use of Models	**Plan for and Implement Model Use**	**Orchestration and Integration of Work Activities**	**Develop and implement an approach to represent the system of interest and/or its components through data sets, models, and digital artifacts that support use and reuse in engineering and business activities.**

3. Sub-components, derived from the additional detail under each of the 14 focus areas in the DoD DE strategy.

An activity description is defined for each 3-part combination as illustrated in Table 2. It focuses on the subcomponent level of description, to explicitly define what an organization needs to accomplish for a given DE capability. Activities are intended to be as specific as possible while still being relevant across multiple organization types.

Maturity Levels

Each activity description is expanded with maturity levels 0 through 4, derived from INCOSE MBCM maturity stages to the maximum extent possible. An example maturity level definition structure for the activity description shown in Table 2 is provided in Table 3. The top row in each table clarifies the nature of the entries in that column. For the maturity levels, *the same pattern captured in the top row holds for all maturity levels in every workbook tab for all activity descriptions; the mental model of what a maturity level means is consistent throughout the entire DE CMAF.*

6. EXCEL WORKBOOK-BASE CMAF: HOW IT'S USED

6.1. Starting an Assessment

An assessment team should plan for 4 to 5 two-hour sessions, aligned with the 5 assessment areas, approachable in any order. A team should start on a given activity sheet, selecting activities within it as important to their organization. It is acceptable and expected if only some activities on a sheet are identified as valuable. On each activity sheet, the left-most columns are the assessment columns as shown in Figure 2. Each activity deemed important should be assigned a current, short-term goal, and long-term goal maturity level. An organization may define its own short-term and long-term timeframes, but 2 and 5 years are recommended starting points respectively. When making a maturity determination, round down; sub-levels, such as "2.5," are not needed in the framework. Organizations should select current and goal levels based on ease of use and understanding of what those mean for their organization. Finally, assessors should assign a responsible party for each activity of interest. There

Table 3. Example Maturity Level build for Activity Description shown in Table 2

This is the "What you need to do" description.	*"The thing" does not exist or is ad hoc.*	*The organization has identified "the thing"*	*The organization has communicated and has started implementing "the thing"*	*The organization is refining and maturing "the thing"*	*The organization is mature in its digital practice regarding "the thing"*
CMAF Activity Description	**Maturity Level 0**	**Maturity Level 1**	**Maturity Level 2**	**Maturity Level 3**	**Maturity Level 4**
Develop and implement an approach to represent the system of interest and/or its components through data sets, models, and digital artifacts that support use and reuse in engineering and business activities.	**There are no formal plans for model use across technical engineering and systems engineering activities relevant to the business needs of the organization. Specifically, organization approaches to technical use, requirements traceability, and structured use of data sets, models, and simulations are ad hoc.**	**Types of data, models, simulations, and other digital artifacts needed to meet organization responsibilities across engineering and business activities are identified.**	**Maturity Level 1 + Specific DE (inclusive of MBSE) processes that describe how data, models, simulations, and digital artifacts will be used to guide business practices and technical decisions are defined. The organization is beginning to implement these processes.**	**Maturity Level 2 + Data sets, models, and simulations are actively used to represent the system of interest or its components across the organization. Initial processes for feedback on performance of these activities to support the applicable phase(s) of acquisition are in place.**	**Maturity Level 3 + Structured DE practices are in place that allow the organization to vet potential requirements prior to Request for Proposal release, assess engineering change orders or program upgrades, etc. Modeling and simulation activities are used as applicable to assess and optimize resource usage, examine process changes, support supply-chain management routing and inventory quantities, business decisions, etc. Metrics are defined and continuously analyzed to guide consistent and effective digital engineering activities and implementation of model-based practices.**

is no need to rank prioritize DE activities in the DE CMAF. Percent completion is calculated as an equal weighting of those DE capability activities identified as important. This is intentional as a separate rank prioritization of numerous activities (DE CMAF has 57) would put too much emphasis on yet another layer of analytical evaluation (ranking), take away from the intent, and still not tell a user how far they needed to go.

6.2. Assessment Completed. Now what?

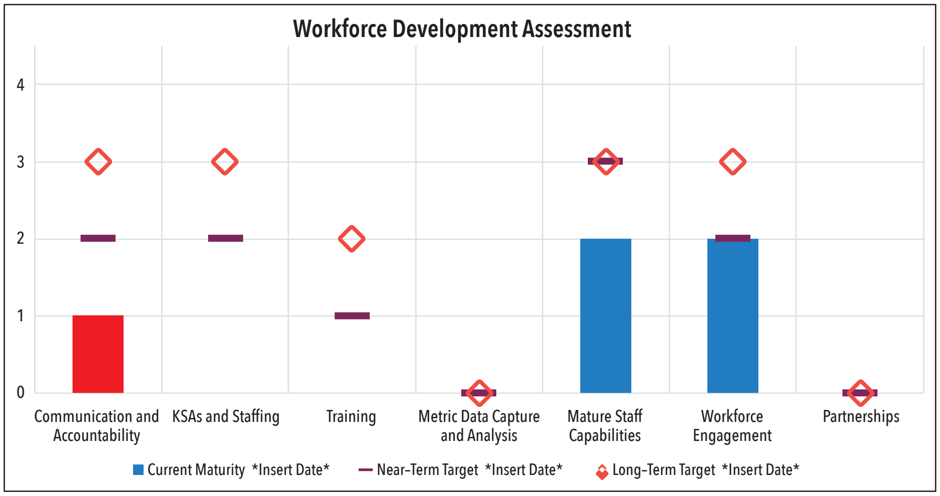
Capability and maturity profiles generated by the DE CMAF for each area will be representative of the organization conducting the assessment at that moment in time, created from activity lines important to an organization plus the current maturity level and shorter- and longer-term maturity goals for each. Figure 3 depicts an example chart for a notional assessment of workforce development. Text to the right tells a user (i) how far the organization needs to go to reach its goal, and (ii) what percent of that journey is complete.

6.3. Using the Dashboard Sheet to See the Overview

The "dashboard," depicted in Figure 4, serves as a quick-look to assess relative progress across the goal-grouped activity-specific sheets in one location. The top two charts show roll-ups for an organization's current and near-term DE capability maturity levels overall (at left) and aggregated for each of the 5 goal-grouped activity-specific areas (at right). These values are currently averaged percent completions from the grouped capability areas on each activity sheet. The bottom charts show detail for each of the goal-grouped activity-specific areas, as shown in Figure 3. Organizations should focus on what this reveals about where to put efforts and

Short Name	*Is this important to your organization right now or in the forseeable future?*	*Your organization's current maturity level — (0, 1, 2, 3, 4)*	*Your shorter-term maturity level goal — (0, 1, 2, 3, 4)*	*Your longer-term maturity level goal — (0, 1, 2, 3, 4)*	*Who will be responsible?*	*Major Goal Area of Applicability (Aligned with DoD DE Strategy top-level goal, i.e., goals 1 thru 5)*	*(Aligned with Goal Focus Areas as numbered in DoD DE Strategy as Goal.n — e.g., 5.1)*	*(Aligned with Focus Area Components as described in text under each numbered Focus Area in the DoD DE Strategy*	*This is the "What you need to do" description. Here, this is based on the sub-component activity description from the cited DoD Strategy portion.*
Short Name	Important	Current Maturity *Insert Date*	Near-Term Target *Insert Date*	Long-Term Target *Insert Date*	Responsible Party?	Capability Category	Focus Area	Sub-Component (Activity)	CMAF Activity Description

Figure 2. Excel worksheet DE CMAF excerpt showing left-most assessment columns



Near Term: Accomplished a sum total of 5 of the targeted 10 Maturity Levels across 6 Activities (50% complete)

Long Term: Accomplished a sum total of 5 of the targeted 14 Maturity Levels across 6 Activities (36% complete)

Number of Activities Designated Important	6
Sum Total Near Term Current Maturity	5
Sum Total Long Term Current Maturity	5
Sum Total Near Term Target Maturity Levels	10
Sum Total Long Term Target Maturity Levels	14
Near Term Percentage Complete	50%
Long Term Percentage Complete	36%

Figure 3. Example DE CMAF profile for workforce development

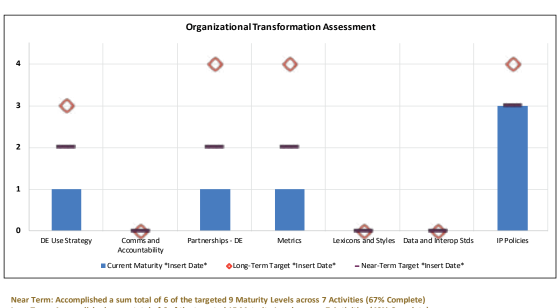


Figure 4. DE CMAF dashboard example

resources and progress toward achieving these goals over time. Similarly, higher-level organizations should recognize the great utility of customization and that individual assessments from different organizations are likely not directly comparable.

7. FUTURE WORK AND NEXT STEPS

The DE CMAF team is working on creating a descriptive model using the unified architecture framework modeling language (UAFML) so that organizations can more seamlessly use it in conjunction with their own enterprise architecture descriptions. The team is also exploring more complex DE transition planning use cases that will tie in other approaches and tools available or under development. One such flow would use a DE CMAF as input into a request for information (RFI), both of which would serve as inputs into the US DoD's DETECT tool, which would then serve as the basis for creating and conducting an analysis of alternatives (AoA). An example using these four high-level steps would be valuable to organization who are implementing a portion of their DE transition. ■

REFERENCES

- Hoheb, A., and J. Hale. 2019. Model-Based Enterprise Capabilities Matrix, INCOSE (No. M19-7475-1).
- Office of the Deputy Assistant Secretary of Defense for Systems Engineering. 2018. Digital Engineering Strategy.
- Shearin, M. E., V. Sitterle, and Z. Connor. 2025. "Overview for the C-MAF." Presented at the Thirty-Fourth Annual International Symposium of INCOSE, Dublin, IE, 5 July.
- United States Department of Defense, Research and Engineering. 2022. "Digital Engineering Ecosystem General Requirements." https://www.cto.mil/wp-content/uploads/2023/06/Dig-Eng-Ecosystem-Reqs.pdf.
- United States Department of Defense. 2023. "DoDI 5000.97 – Digital Engineering."

ABOUT THE AUTHORS

Dr. Valerie Sitterle is the deputy chief technology officer for the Systems Engineering Research Center led by Stevens Institute of Technology. She has over 25 years of experience in analysis of technical and use characteristics of defense systems in operational environments, with a current focus in systems engineering research and its impact on DoD capability development. She was previously a principal research engineer and chief scientist for the systems engineering research division, Georgia Tech Research Institute.

Mr. Mike Shearin is a senior researcher engineer at the Georgia Tech Research Institute and has 21 years of combined industry and research experience in systems engineering and digital engineering. He is currently head of the DE tool and environments branch at GTRI. Mr. Shearin also is the co-lead of the OMG model-based acquisition managed community and the OMG DE Working Group. He holds a BS in computer engineering and an MS in applied systems engineering, both from Georgia Tech.

Dr. Owen Eslinger is a distinguished program manager at the US Army ASA(ALT), specializing in computational and applied mathematics. With a PhD from The University of Texas at Austin, his expertise spans digital engineering, high-performance computing, and the acquisition of defense weapons systems. Dr. Eslinger has managed several Department of Defense programs, authored numerous peer-reviewed publications, and contributed to advancing digital engineering. He is an adjunct professor, mentor, and recipient of multiple awards, including the ERDC Vogel Award and Army R&D Achievement Award.

Systems Engineering: The Journal of The International Council on Systems Engineering

Call for Papers

The *Systems Engineering* journal is intended to be a primary source of multidisciplinary information for the systems engineering and management of products and services, and processes of all types. Systems engineering activities involve the technologies and system management approaches needed for

- definition of systems, including identification of user requirements and technological specifications;
- development of systems, including conceptual architectures, tradeoff of design concepts, configuration management during system development, integration of new systems with legacy systems, integrated product and process development; and
- deployment of systems, including operational test and evaluation, maintenance over an extended life-cycle, and re-engineering.

Systems Engineering is the archival journal of, and exists to serve the following objectives of, the International Council on Systems Engineering (INCOSE):

- To provide a focal point for dissemination of systems engineering knowledge
- To promote collaboration in systems engineering education and research
- To encourage and assure establishment of professional standards for integrity in the practice of systems engineering
- To improve the professional status of all those engaged in the practice of systems engineering
- To encourage governmental and industrial support for research and educational programs that will improve the systems engineering process and its practice

The journal supports these goals by providing a continuing, respected publication of peer-reviewed results from research and development in the area of systems engineering. Systems engineering is defined broadly in this context as an interdisciplinary approach and means to enable the realization of successful systems that are of high quality, cost-effective, and trustworthy in meeting customer requirements.

The *Systems Engineering* journal is dedicated to all aspects of the engineering of systems: technical, management, economic, and social. It focuses on the life-cycle processes needed to create trustworthy and high-quality systems. It will also emphasize the systems management efforts needed to define, develop, and deploy trustworthy and high quality processes for the production of systems. Within this, *Systems Engineering* is especially concerned with evaluation of the efficiency and effectiveness of systems management, technical direction, and integration of systems. *Systems Engineering* is also very concerned with the engineering of systems that support sustainable development. Modern systems, including both products and services, are often very knowledge-intensive, and are found in both the public and private sectors. The journal emphasizes strategic and program management of these, and the information and knowledge base for knowledge principles, knowledge practices, and knowledge perspectives for the engineering of systems. Definitive case studies involving systems engineering practice are especially welcome.

The journal is a primary source of information for the systems engineering of products and services that are generally large in scale, scope, and complexity. *Systems Engineering* will be especially concerned with process- or product-line–related efforts needed to produce products that are trustworthy and of high quality, and that are cost effective in meeting user needs. A major component of this is system cost and operational effectiveness determination, and the development of processes that ensure that products are cost effective. This requires the integration of a number of engineering disciplines necessary for the definition, development, and deployment of complex systems. It also requires attention to the lifecycle process used to produce systems, and the integration of systems, including legacy systems, at various architectural levels. In addition, appropriate systems management of information and knowledge across technologies, organizations, and environments is also needed to insure a sustainable world.

The journal will accept and review submissions in English from any author, in any global locality, whether or not the author is an INCOSE member. A body of international peers will review all submissions, and the reviewers will suggest potential revisions to the author, with the intent to achieve published papers that

- relate to the field of systems engineering;
- represent new, previously unpublished work;
- advance the state of knowledge of the field; and
- conform to a high standard of scholarly presentation.

Editorial selection of works for publication will be made based on content, without regard to the stature of the authors. Selections will include a wide variety of international works, recognizing and supporting the essential breadth and universality of the field. Final selection of papers for publication, and the form of publication, shall rest with the editor.

Submission of quality papers for review is strongly encouraged. The review process is estimated to take three months, occasionally longer for hard-copy manuscript.

Systems Engineering operates an online submission and peer review system that allows authors to submit articles online and track their progress, throughout the peer-review process, via a web interface. All papers submitted to *Systems Engineering*, including revisions or resubmissions of prior manuscripts, must be made through the online system. Contributions sent through regular mail on paper or emails with attachments will not be reviewed or acknowledged.

All manuscripts must be submitted online to *Systems Engineering* at ScholarOne Manuscripts, located at:

https://mc.manuscriptcentral.com/SYS

Full instructions and support are available on the site, and a user ID and password can be obtained on the first visit.