

Analysis for an End-to-End MBSE Operational Architecture

Joshua Adelabu¹, Bhushan Lohar¹, John T. Wade¹, Sean Walker¹,
and Carlos Montalvo²

¹Department of Systems Engineering, University of South Alabama, Mobile, AL 36688, USA

²Department of Mechanical, Aerospace, and Biomedical Engineering, University of South Alabama, Mobile, AL 36688, USA

ABSTRACT

Model-Based Systems Engineering (MBSE) has shifted systems engineering toward digitally integrated architectures. However, methodological fragmentation among frameworks like UAF, ARCADIA, and MOFLT persists, characterized by ontological gaps and restricted interoperability. These constraints impede seamless End-to-End (E2E) lifecycle continuity across the SE “Vee” model. This research addresses these limitations by extending the E2E MBSE Framework, specifically formalizing the Operational Architecture (Ops-Arch) phase. Using a rigorous decomposition and reconstruction methodology, the Ops-Arch is synthesized by leveraging NASA standards and extracting high-fidelity artifacts from legacy frameworks. The development employs comparative analysis, reverse engineering of architectural patterns, and meta-model engineering to identify functional gaps and circularities. The resulting synthesis produces a comprehensive reference architecture that harmonizes disparate modeling blocks into a unified construct. This approach reduces practitioner cognitive load and facilitates the transition from abstract intent to concrete structural models. Key contributions include a cross-framework meta-model alignment, a semantic gap map, and a formalized E2E MBSE construct package anchored in a reference ontology. By bridging methodological heterogeneity, this framework ensures consistency in large-scale system design and enhances cross-disciplinary collaboration. This work demonstrates that dissimilar operational views can be systematically harmonized into a singular, precise architecture, improving both technical accuracy and human comprehension. Future research will evolve these constructs into executable architectures applied to a System of Interest (Sol) case study for integrated Verification and Validation (V&V).

Keywords: End-to-End model-based systems engineering, Architectural framework, Digital thread, Traceability, Model governance

INTRODUCTION

The International Council on Systems Engineering (INCOSE) Vision 2035 posits that the future of Systems Engineering (SE) “is model-based, leveraging next generation modeling, simulation, and visualization environments powered by the global digital transformation, to specify, analyze, design, and verify systems” (INCOSE, 2022). MBSE has fundamentally transitioned SE from document-centric paradigms toward digitally integrated architectures,

ensuring interface traceability and rigorous V&V (Adelabu et al., 2025). MBSE enhances complexity management, accelerates information sharing, and improves transparency among disciplines and stakeholders, establishing a strong foundation of knowledge throughout the system lifecycle (Khandoker et al., 2026). A critical component of early-phase development is the Operational Architecture (Ops-Arch) also termed Operational Analysis in frameworks such as the Unified Architecture Framework (UAF) (Object Management Group (OMG), 2021), ARChitecture Analysis and Design Integrated Approach (ARCADIA) (Voirin, 2023), and Mission Operation Functional Logical Technical (MOFLT) (Faudou, 2022). Despite the maturation of these frameworks, the landscape remains characterized by methodological fragmentation. Existing approaches exhibit semantic inconsistencies, ontological gaps, and restricted interoperability, impeding the realization of a seamless End-to-End (E2E) lifecycle. These constraints are particularly pronounced when maintaining continuity across the SE V-Model in complex developments, as “MBSE frameworks often lack adaptability, constraining cross-domain application, hindering model reuse, and lacking interfaces from the left side to the right side” (Adelabu et al., 2025).

Analysis confirms the hypothesis that traditional frameworks are fragmented from an E2E perspective relative to the SE V-Model. “Many MBSE frameworks lack adequate support for specific domains, hindering their effectiveness in those areas,” and where frameworks exist, there is “a significant lack of standardization... leading to inconsistencies and confusion” (Adelabu & Lohar, 2025a). Furthermore, interdependencies between modeling phases vary by domain, confirming that traditional frameworks are tailored to specific niches. This specificity limits cross-domain application and creates a distinct demand to incorporate identified commonalities into a unified package capable of supporting multi-domain applications and use cases. Key challenges include steep learning curves, potential for impasse, overwhelming complexity, and a lack of clarity regarding the Right-Side (Realization/Verification) of the V-Model (Adelabu & Lohar, 2025b).

Consequently, a comprehensive analysis of the existing framework landscape has driven the continuous development of the E2E MBSE Framework. The resultant twenty-four packages have been streamlined and aligned to provide a continuous Digital Thread, facilitating the development of any SoI from Evolution to Extinction. These twenty-four packages provide solutions for traditional, emerging, and complex systems development. It is applicable to core SE concepts, including human systems integration, design methodologies, and multi-method modeling and analysis. Furthermore, it supports Systems-of-Systems (SoS) development, cyber-physical systems, and socio-technical systems.

As a result, the E2E MBSE Framework enables the realization of INCOSE Vision 2035, allowing “systems engineers to evaluate and assess an order of magnitude more alternative designs more quickly and thoroughly than can be done on a single design today” (INCOSE, 2022). Designed to guide users through the entire SE V-Model, from the left-side definition, across

the bottom gap, to the right-side realization the framework comprises twenty-four packages (Adelabu & Lohar, 2025c). The following table delineates the packages, divided between those explicating the left side (Definition) and those explicating the right side (Realization/Verification). This modular structure allows systems engineers and practitioners to tailor the framework by adding or deleting packages to fit their specific domain or application.

Table 1: End-to-End MBSE framework 24 packages from evolution to extinction.

Pkg	E2E MBSE Framework (Left Side of V)	Pkg	E2E MBSE Framework (Right Side of V)
1	Evolution	13	Implementation
2	Mission Analysis	14	Component or Unit Integration and Testing
3	Stakeholder Analysis	15	Subsystem Integration, Verification, and Testing
4	Requirement Analysis	16	System Integration, Verification, and Testing
5	System Analysis	17	System Validation, and Operational Testing
6	Operational Analysis	18	Digital Twin
7	Use Case	19	Qualification, Acceptance, and Compliance Testing
8	Functional Analysis	20	Deployment, Operations, and Sustainment Testing
9	Logical Architecture	21	Post-Deployment Verification, and Validation
10	Requirement Allocation Architecture	22	Production, Scalability, and Modification
11	Physical Architecture	23	In-Service-Maintenance (Operation)
12	Detailed Design (Virtual Twin)	24	Extinction (Sustainable Disposition)

In this paper, an Operational Architecture (Ops-Arch) is selected from the twenty-four primary packages as an exemplar to demonstrate the analytical and developmental process of the E2E MBSE Framework. This Ops-Arch nomenclature is utilized to distinguish the architectural synthesis from the broader process of Operational Analysis. Which originates in the early lifecycle phases and maintains bi-directional traceability across the entire system lifecycle. Systems engineers and architects must model the Ops-Arch with downstream concurrency, ensuring early-stage artifacts directly inform the intermediate and terminal phases. In the context of the SE V-Model, this research addresses the persistent semantic gap between Left-Side and Right-Side. The transition from document-centric to model-based engineering has exacerbated this gap, revealing a fragmentation trap within extant frameworks. This study mitigates these limitations by extending the E2E MBSE Framework, specifically formalizing the Ops-Arch as a critical semantic juncture among the twenty-four essential lifecycle phases.

Rationale for Operational Architecture

Within the E2E MBSE Framework, Ops-Arch serves as a critical semantic juncture in the digital thread. Because, this phase facilitates the transition from abstract stakeholder intent to formalized system behavior, establishing the conceptual and contextual baseline for subsequent Functional and Logical Architectures. As the primary foundation for downstream architectural development, the precision of this translation is paramount. Any misalignment during this phase introduces a fundamental intent gap between stakeholder requirements and the SoI, which propagates through all subsequent lifecycle phases.

LITERATURE REVIEW

“The ontological rigidity of established architectural frameworks (e.g., DoDAF, TOGAF) and the metamodel extensibility limitations of SysML impede domain-specific adaptation and model federation” (Adelabu et al., 2025). Currently, MBSE application “remains largely confined to the early-design phase—particularly requirements and architecture modeling—while later lifecycle phases are underrepresented or modeled in isolation” (Khandoker et al., 2026; Akundi & Lopez, 2021). Consequently, SoI “are often prematurely architected, hindering optimization at the architectural level. Furthermore, many system architectures are not sufficiently adaptable to changing requirements” (Zhang et al., 2025). Adaptability is further constrained by architectural complexity. For instance, while UAF’s seventy-two views provide comprehensive MBSE support, the steep learning curve poses significant challenges for practitioners. This often leads to the development of fragmented, domain-specific frameworks, resulting in extensibility and interoperability deficits. As a result, “MBSE is often performed within very localized contexts, creating models that inform their immediate stakeholders but are not necessarily suitable for broader use or reuse” (Noguchi, 2025). These silo modeling efforts minimize the scalability of MBSE across frameworks and domains. The following sections expound upon the theoretical, structural, and ontological foundations of these traditional frameworks.

Existing Frameworks’ Operational Phase

An analysis of the operational phase across legacy frameworks reveals significant taxonomical, semantic, and ontological variances. Figures 1 – 3 below represent three traditional frameworks’ Ops-Arch showing some commonalities and differences, making it difficult to match the metamodel. Some of the concepts in these frameworks have the same name but different ontological definitions.

ARCADIA: Comprises six core elements: Operational Mission, Operational Capability, Operational Process, Operational Activity Scenario, Operational Interaction, and Operational Activity (Voirin, 2023).

UAF: Defines its Operational Architecture through eight primary constructs: Capability, Actual Enduring Task, Operational Activity, Operational Activity Action, Operational Agent, Operational Role, Operational Port, and Operational Interface (OMG, 2021).

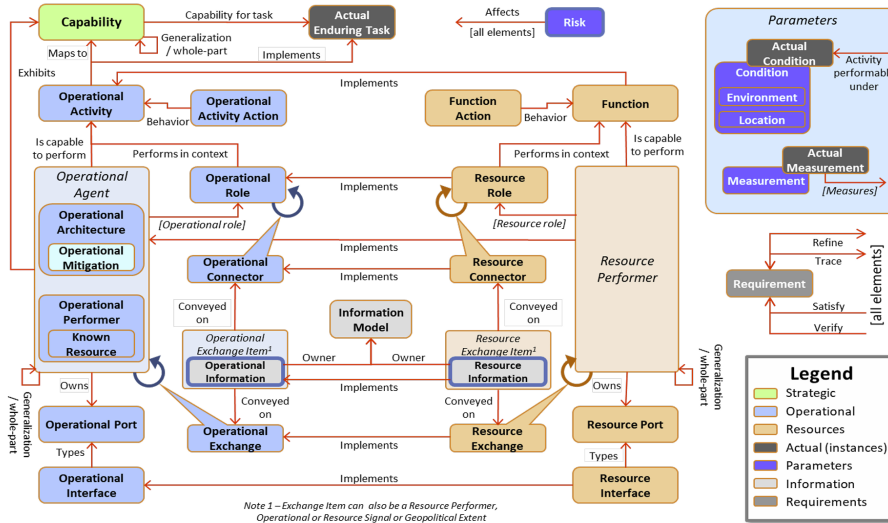


Figure 4:18 - Conceptual Schema for Operational Architectures

Figure 3: UAF conceptual schema for operational architectures (OMG, 2021).

Ontological Foundation and Semantics in MBSE

“An ontology is an explicit specification of a conceptualisation and provides the foundation for MBSE and architecture descriptions. Stronger foundations support longer-lived models that work better together” (Hampton et al., 2025). System modeling necessitates the formalization of concepts, terminology, and relationships; therefore, the constituent elements of a framework define its domain ontology. Given the increasing complexities of modern SoI (Adelabu & Lohar, 2025c), it is imperative to “put the O [Ontology] in model” (Hampton et al., 2025). Ontologies facilitate stakeholder concurrency, risk mitigation, and seamless architectural integration (Hampton et al., 2025). This ontological robustness underpins the development of an E2E MBSE framework designed to enhance usability and minimize cognitive load.

Integrating modeling languages, tools, and methodologies presents significant challenges for data interoperability. Semantic MBSE mitigates these issues by enhancing precision and expressiveness through a language founded on semantic specifications rather than relying solely on graphical notations (Lu et al., 2022). While graphical notations offer effective communication dexterity, a hybrid approach integrating graphical and semantic MBSE enables superior consistency and tool-chain integration across the system lifecycle. Consequently, combining ontological and semantic

graphical representations with associated relationship matrices facilitates a standardized taxonomy and a cross-framework conversion engine or library. This approach addresses a primary challenge: disparate user definitions and semantic misalignment.

Heterogeneity across frameworks, such as DoDAF, MODAF, NAF, MOFLT, and ARCADIA impedes metamodel alignment due to semantic variances, including homonyms, synonyms, and complex architectural mapping combinations. For instance, the UAF Domain Meta Model (DMM) is founded upon the IDEAS Ontology, necessitating that all entities incorporate four-dimensional (spatiotemporal) extensions, subtypes, and measures (OMG Standards Development Organization, 2022). Developing and maintaining cross-framework mappings requires significant resource allocation from systems architects, academia, and agencies such as DoD/DoW, NASA and ESA. Subsequently, there is a critical requirement for a formal methodology for semantic consolidation, relational matrices, or an interoperability framework. Ops-Arch addresses these systemic challenges, providing a solution for modelers, engineers, and strategic decision-makers.

METHODOLOGY

The Ops-Arch is synthesized via a rigorous decomposition and reconstruction methodology, leveraging NASA-defined standards and the extraction of high-fidelity modeling artifacts from legacy frameworks. The development process employs comparative framework analysis, architectural pattern reverse engineering, and meta-model engineering. NASA's transition toward Digital Systems Engineering (DSE) facilitates the decomposition of the 24 essential lifecycle phases within an E2E MBSE Framework into machine-readable modeling artifacts, establishing the authoritative foundation for operational architecture synthesis. This approach ensures methodological alignment for systems engineers and architects, promoting rapid adoption through familiarity with extant practices. The process involves reconstructing mined blocks through the specification of relationships and the analysis of similarities, variances, circularities, and gaps (Lohar, 2022). New artifact elements are postulated to mitigate these gaps, resulting in the synthesis of mined and novel constructs into a comprehensive, E2E MBSE Ops-Arch.

A multifaceted comparative analysis is executed to mitigate the fragmentation trap inherent in legacy frameworks. This research advocates a transition in meta-model engineering from purely syntactic mapping to the reconciliation of ontological gaps through rigorous decomposition and the establishment of a common taxonomy for framework alignment and interoperability. By analyzing the varied definitions of Ops-Arch constituents, this work postulates novel artifact elements required to maintain a continuous Digital Thread.

Reconstruction and Cross-Framework Analysis

While the E2E MBSE Framework remains tool-agnostic, a physical implementation was executed for architectural representation. Each packages was reconstructed using CATIA (No Magic) Systems of Systems Architect (2026x) via Block Definition Diagrams (BDD). Architectural elements were modeled as blocks (class) and assigned specific stereotypes (e.g., UAF, ARCADIA, MOFLT, and E2E). Relationships were mapped and replicated based on established SE semantics. A comprehensive model was developed to extract all blocks, stereotypes, and relationships, assigning Unique Identifiers (UIDs) to ensure traceability. Conceptual definitions were audited to establish classes and higher-level taxonomies, facilitating cross-framework synchronization and gateway connections. Following the isolation of variances and the retention of invariant concepts, identified functional gaps were addressed by postulating new blocks stereotyped as E2E, resulting in the synthesized architecture in subsequent section.

RESULTS

The following section details the relational representation of cross-framework semantics and the synthesized ontological constructs for Ops-Arch exemplar. The Ops-Arch architectural spine is comprised of seven postulated main blocks. The taxonomy (classification structure) of these blocks, connected by high-fidelity semantic links, is detailed below:

1. **Operational Intent and Capability:** Defines system purpose and mission effects. Contains: Operational Mission, System of Interest (SoI), Operational Capability, Capability, Actual Enduring Task.
2. **Operational Concept, Context, and System Usage:** Defines operational environment, conditions, and resources. Contains: Operational Concept, Context, System Usage, System, Entity Usage, Domain, Design Domain, Known Resource.
3. **Operational Behavior and Execution:** Defines temporal sequencing and actions. Contains: Operational Process, Activity Scenario, Activity, Activity Action, Phase, Phase Behavior, Task, Task Invocation.
4. **Operational Actors, Agency, and Roles:** Defines performers and organizational structure. Contains: Operational Performer, Agent, Architecture, Mitigation, Agentic Performer, Role, Actors Structure, Operational AI Agent.
5. **Operational Interactions, Interfaces, Ports, and Flows:** Defines connectivity and data exchanges. Contains: Operational Interaction, Port, Interface, Flow, Flow Connection, Flow Interface Type, Flow Type, Flow Property, Flow Type Category.
6. **Operational Virtual Physical Entity and Structural Representation:** Defines operational environment entities and supports Digital Twin realization. Contains: Entity, Entity Class, Entity Instance, Physical Entity, Virtual Entity.

- Operational Intelligence and Interoperability: Defines reasoning, semantic coherence, and cross-system integration. Contains: Semantic Ontology, Interoperability Descriptor, Operational Informational Link.

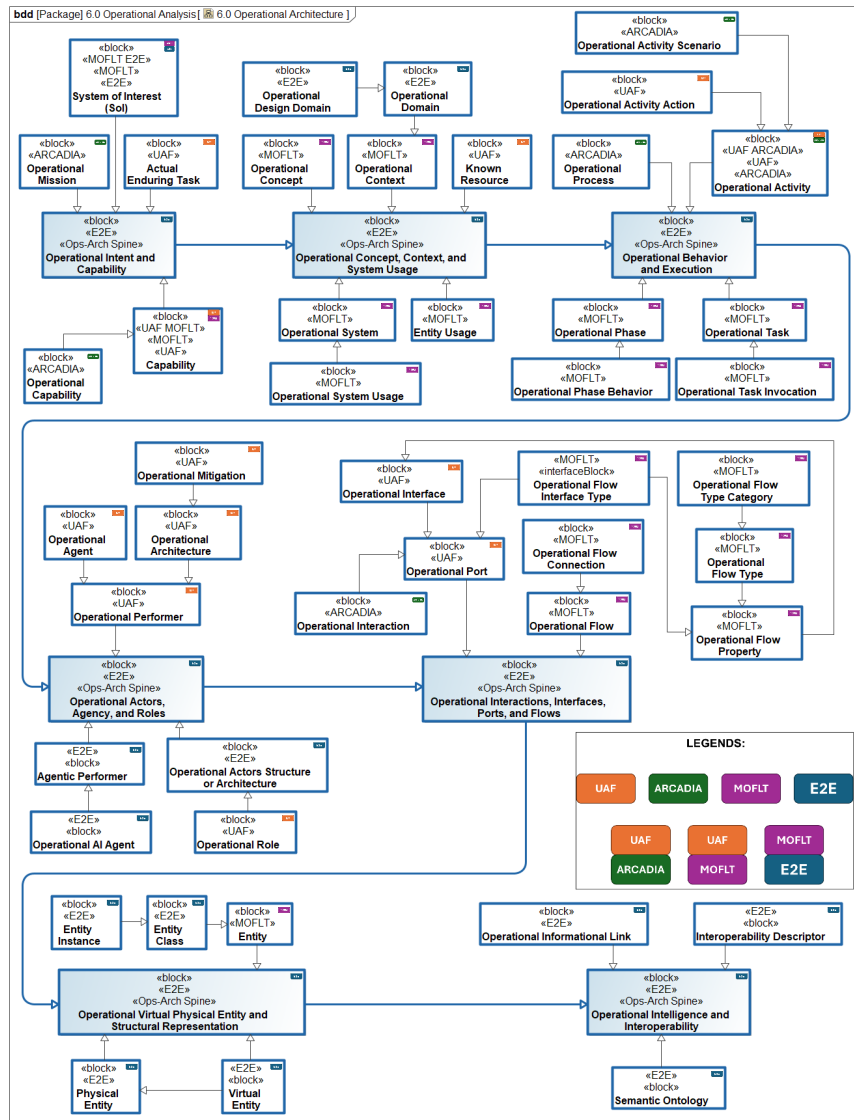


Figure 5: E2E MBSE framework operational architecture.

Architectural Spine Grouping Objectives

- Semantic Separation: Decouples intent, behavior, interaction, agency, and context.
- Core Composition: Elevates AI and E2E semantics to a fundamental operational spine component, constituting the primary structural bone and recipe.

- Framework Optimization: Utilizes ARCADIA, UAF, and MOFLT elements as a springboard for next-generation framework synthesis, rather than treating them as dogmatic constraints.
- Operational Decomposition: Facilitates clean decomposition of the Ops-Arch.

CONCLUSION AND FUTURE WORK

Ontological intersection between ARCADIA and UAF is identified through the Operational Activity construct, while a tri-framework intersection between UAF, ARCADIA, and MOFLT is established via the Capability construct, with Operational Capability (ARCADIA) mapped as a subset of the broader Capability (UAF/MOFLT) definition. Consequently, the E2E MBSE Framework aligns with the INCOSE Vision 2035, facilitating “high fidelity models, advanced visualization, and highly integrated, multi-disciplinary simulations” to enhance design space exploration (INCOSE, 2022).

The resulting Ops-Arch provides a robust foundation for tool-agnostic analysis, semantic interoperability, and lifecycle traceability, functioning as a critical semantic juncture in the Digital Thread. By elevating AI and E2E semantics to a core operational spine, this framework bridges methodological heterogeneity, transforming abstract stakeholder intent into formalized system behavior while mitigating the fragmentation trap inherent in legacy architectures. Key contributions include a cross-framework meta-model alignment and traceability matrix, a semantic gap map derived from E2E stereotypes, and a reusable construct package anchored in a formal reference ontology. This work demonstrates that dissimilar operational views can be systematically harmonized into a singular, precise architecture that improves both technical accuracy and human comprehension. Future research will focus on the evolution of these constructs into executable architectures and their application to a SoI case study for integrated V&V.

REFERENCES

- Adelabu, J., & Lohar, B. (2025a). End-to-End (E2E) Model-Based Systems Engineering (MBSE) Framework. *Shelby Hall Graduate Research Forum Presentations*, 1(1).
- Adelabu, J., & Lohar, B. (2025b, March 27). End-to-End (E2E) MBSE Framework. *Shelby Hall Graduate Research Forum Posters*.
- Adelabu, J., & Lohar, B. (2025c, September 15). End-to-End (E2E) Model-Based Systems Engineering (MBSE) Framework for Complex Systems Life Cycle Design. *Advanced Systems Research Laboratory (ASRL) Day (1st Founding Anniversary)*.
- Adelabu, J., Lohar, B., T Wade, J., Walker, S., & Montalvo, C. (2025). Building Blocks for Effective Modeling and Simulation of Systems Lifecycle using an End-to-End Model-Based Systems Engineering Framework. *AHFE 2025 Hawaii International Conference on Human Factors in Design, Engineering, and Computing (AHFE 2025 Hawaii Edition)*. <https://doi.org/10.54941/ahfe1007053>
- Akundi, A., & Lopez, V. (2021). A Review on Application of Model Based Systems Engineering to Manufacturing and Production Engineering Systems. *Procedia Computer Science*, 185, 101–108. <https://doi.org/10.1016/j.procs.2021.05.011>

- Ashfaq, M., Sadik, A. R., Das, T., Waseem, M., Makitalo, N., & Mikkonen, T. (2025). *Runtime Composition in Dynamic System of Systems: A Systematic Review of Challenges, Solutions, Tools, and Evaluation Methods*.
- de Sá, D., Schmiedel, C., & Lopes, C. P. (2026). *Liquid Interfaces: A Dynamic Ontology for the Interoperability of Autonomous Systems*.
- De Sousa, J., Faudou, R., Dahl, I., Deredempt, M.-H., & Marty, J.-L. (2022). *Application of MBSE to reverse-engineer OPS-SAT and prepare future IOD missions (including OPS-SAT2)*. https://nebula.esa.int/sites/default/files/neb_study/2614/C4000134652FP.pdf
- Faudou, R. (2022, December 15). Reverse Engineering of a CubeSat with MBSE. *Complex Systems Design & Management (CSD&M) Conference*. https://cesam.community/wp-content/uploads/2023/01/15h30-15_1530_CSDM2022_Faudou.pdf
- Hampton, P., Powley MEng, S., Banham BEng CEng MIET, D., & Qnx, B. (2025, November 25). Putting the O in Model-Ontologies for MBSE Categorisation. *2025 Institute for Systems Engineering (IfSE) Annual Systems Engineering Conference (ASEC)*.
- International Council on Systems Engineering. (2022, January). *Systems Engineering Vision 2035*. Resource Publications Technical Product Systems Engineering Vision 2035. <https://www.incose.org/resources-publications/technical-products/se-vision-2035/>
- Khandoker, A., Sint, S., Gessl, G., & Zeman, K. (2026). Toward Demystifying the Missing Links in Model-Based Systems Engineering (MBSE). *Systems*, 14(2), 158. <https://doi.org/10.3390/systems14020158>
- Lohar, B. R. (2022). Development of new space systems architecture in Sysml using model-based pattern language (Doctoral dissertation, University of South Alabama) https://jagworks.southalabama.edu/theses_diss/75
- Lu, J., Wang, G., Yan, Y., Cameron, D., Lei, F., Zhang, Y., Tang, X., Wang, J., Tang, Junjie, Tang, Jian, Zhang, H., Reniers, M., Törngren, M., Ma, J., Keraron, Y., Wu, S., & Kiritsis, D. (2022). Semantic Model-based Systems Engineering based on KARMA: A Research and Practice Roadmap 2025. *INCOSE International Symposium*, 32(1), 706–720. <https://doi.org/10.1002/iis2.12959>
- Noguchi, R. (2025). Standards Gaps for Enabling Model Interoperability for MBSE in a Digital Engineering Context. *INCOSE International Symposium*, 35(1), 427–443. <https://doi.org/10.1002/iis2.70030>
- Object Management Group (OMG). (2021). *Enterprise Architecture Guide for UAF (Informative) Appendix C*. <https://www.omg.org/spec/UAF/20211201/UAF.xmi>
- OMG Standards Development Organization. (2022, July). *About the Unified Architecture Framework Specification Version 1.3 beta*. <https://www.omg.org/spec/UAF>
- Voirin, J.-L. (2023). *Arcadia Language Reference: Meta Model Arcadia Language Concepts in details*.
- Zhang, L., Chen, Z., Laili, Y., Ren, L., Deen, M. J., Cai, W., Zhang, Y., Zeng, Y., & Gu, P. (2025). MBSE 2.0: Toward More Integrated, Comprehensive, and Intelligent MBSE. *Systems*, 13(7), 584. <https://doi.org/10.3390/systems13070584>.