

The Collapse of MBSE and the Collateral Damage to Systems Engineering

Art Villanueva, DEng, ESEP, Phronos
art@phronos.com

February 2026

Abstract

Model-Based Systems Engineering (MBSE) has increasingly been treated as synonymous with systems engineering, yet in practice it has often devolved into documentation rather than decision support. System models are produced, governed, and maintained, but rarely serve as load-bearing inputs to consequential engineering decisions. As a result, the limitations of MBSE practice are frequently misattributed to systems engineering itself. This paper argues that the issue is not disciplinary failure but a category error: descriptive modeling techniques have been mis-scoped and asked to substitute for analytical reasoning and judgment. Tool constraints, workflow separation, and organizational role definitions have reinforced this substitution, shifting systems engineering from a decision-making discipline to an artifact-producing function and eroding its authority. While advances such as SysML v2 and automation raise the technical ceiling of MBSE, they do not resolve the underlying misalignment. The paper concludes that systems engineering must be reasserted as a load-bearing decision function, with MBSE properly positioned as supporting infrastructure for representation and coordination rather than as its defining practice.

1. The Problem: Why MBSE Feels Unsatisfying in Practice

Across many organizations, systems engineers report a similar experience. Significant effort is invested in MBSE tools, training, governance, and compliance. System models grow in size and detail. Reviews are held, artifacts are approved, and traceability matrices are complete.

Yet when consequential design decisions are made, the system model is rarely the decisive input. Decisions are driven instead by simulations, spreadsheets, prototypes, domain-specific analyses, or expert judgment. The model is updated afterward to reflect what has already been decided.

Over time, this creates a persistent sense that modeling work is disconnected from engineering work. What makes this discomfort difficult to articulate is that nothing is obviously broken. The tools function as intended, the models

are internally consistent, and the prescribed process is followed. From a compliance perspective, success is visible and defensible. Yet the value proposition feels thin.

This paper treats that discomfort as a signal that suggests that the dissatisfaction arises from a structural mismatch between what MBSE is being asked to do and what it is capable of doing in most organizational contexts.

2. A Category Error: Systems Engineering Is Not MBSE

2.1 Systems Engineering as a Discipline

Systems engineering is fundamentally a way of reasoning about complicated and complex systems across their lifecycle². Its core activities include framing problems, evaluating trade-offs, defining interfaces, identifying and managing risk, planning integration, and reasoning about lifecycle behavior under uncertainty.

These activities are inherently cognitive and social. They rely on judgment, negotiation, synthesis, and the ability to integrate perspectives across technical and organizational boundaries. Historically, systems engineering has been practiced using whatever tools best supported thinking at the time: whiteboards, spreadsheets, simulations, prototypes, structured discussions, and domain-specific analyses.

Systems engineering has never been defined by a particular representational medium.

2.2 MBSE as a Technique

MBSE is a set of practices and techniques for representing and managing system information using models, independent of any specific language or tool³. SysML has been the most common realization of MBSE in practice, but it is not synonymous with MBSE. Its original promise was to provide a shared representational foundation that could improve consistency, communication, and traceability across disciplines.

MBSE was meant to help engineers reason together. It was not meant to stand in for that reasoning.

2.3 How Substitution is Occurring

Over time, MBSE is being treated as synonymous with systems engineering. Training programs, standards, organizational processes, and role definitions are actively reinforcing this association. In many environments, “doing systems engineering” came to mean producing SysML models in approved tools.

As this substitution is occurring, the performance of MBSE is beginning to define the perceived performance of systems engineering itself. When models fail to influence decisions, the discipline, not the technique, is being judged to have failed.

2.4 A Practical Comparison

<i>Dimension</i>	Systems Engineering	MBSE
<i>What it is</i>	A discipline for reasoning about complicated/complex systems	A set of tools and techniques for representing system information
<i>Primary purpose</i>	Support sound technical and organizational decisions	Organize, communicate, and maintain system descriptions
<i>Relationship to decisions</i>	Directly frames, evaluates, and constrains decisions	Often documents decisions after they are made
<i>Generativity</i>	Produces insights, trade-offs, and risk understanding	Produces artifacts and traceability
<i>Analytical capability</i>	Draws on analysis, simulation, experiments, and judgment	Relies on external tools for analysis
<i>Load-bearing role</i>	Removing it degrades system outcomes	Removing it usually causes inconvenience
<i>Dependence on tools</i>	Tool-agnostic	Tool-dependent
<i>Treatment of uncertainty</i>	Actively reasons about unknowns	Tends to represent assumed structure
<i>Failure mode</i>	Poor decisions and unmanaged risk	Stale or compliance-driven models

This distinction is central. Systems engineering is load-bearing. MBSE, as commonly practiced, is not.

3. The Load-Bearing Test: Why MBSE Became Documentation

A useful way to evaluate any engineering practice is to ask a simple question: *what breaks if this practice is removed?*

If structural analysis is removed, structures fail.

If control theory is removed, systems become unstable.

If software execution is removed, nothing runs.

These practices are load-bearing. Their absence produces immediate and observable failure.

In most organizations, the removal of MBSE artifacts produces a different outcome. Engineering work continues. Interfaces are still negotiated. Designs are still analyzed. Systems are still built, integrated, and tested. The loss is felt primarily in coordination, traceability, and review preparation, not in the fundamental ability to engineer the system.

This does not imply that MBSE has no value; it implies that it is rarely carrying the primary engineering load.

3.1 Tool Limits and Expectation Mismatch

SysML v1.x was designed as a descriptive modeling language. It provides no standardized execution semantics, no native treatment of time or continuous behavior, and no built-in mechanisms for uncertainty or probabilistic reasoning⁴. Requirements are represented as text with attributes and links, with satisfaction asserted through traceability rather than demonstrated through evidence. These are the boundaries.

As MBSE adoption expanded, expectations grew faster than tool capability. SysML models were increasingly treated as if they could function as analytical system representations rather than descriptive ones. When models are expected to inform decisions without participating in analysis, modeling effort grows while insight does not.

The result is a widening gap between effort and value. Documentation becomes the one role the tools can reliably fulfill.

3.2 Structural Bias and Workflow Separation

MBSE tools make some activities easy — static structure, hierarchy, allocation — and others difficult — dynamic behavior, uncertainty, cross-domain causality. Engineers naturally model what the tools handle well. Over time, the model becomes complete in form while remaining thin in explanatory power.

Most MBSE environments also require focused, tool-centric interaction. Modeling becomes a distinct activity performed before reviews or after decisions rather than during analysis and synthesis. When modeling is decoupled from decision-making, it becomes retrospective.

MBSE tools also impose nontrivial cognitive and operational overhead. Their complexity, interaction models, and governance requirements often make exploratory reasoning, rapid iteration, and informal analysis more difficult than in lighter-weight analytical environments. As a result, engineers naturally shift substantive thinking to tools that better support analysis and synthesis, while relegating MBSE tools to formalization and documentation activities. When this pattern becomes established, MBSE effectively collapses into documentation.

4. Consequences: The Collateral Damage to Systems Engineering

As MBSE converges on documentation, systems engineering becomes increasingly associated with artifact production rather than with engineering judgment. The discipline is reframed as overhead instead of as a source of insight. Because MBSE artifacts are visible, enumerable, and auditable, they become proxies for systems engineering itself, even when they play no decisive role in shaping outcomes.

When those artifacts fail to influence consequential decisions, confidence in the discipline erodes. Stakeholders begin to equate the limited impact of models with limited engineering value, and systems engineering is judged accordingly.

The deeper consequence is a loss of epistemic authority. When system models do not predict behavior, reveal failure modes, or constrain decisions, systems engineering is no longer consulted as a primary source of understanding. Decisions migrate toward groups that can demonstrate predictive or analytical power — analysis teams, software organizations, test organizations, or domain specialists operating outside the systems function.

Once systems engineering is excluded from decision-making forums, it loses the opportunity to demonstrate value. This exclusion is self-reinforcing: diminished authority leads to reduced influence, which further diminishes perceived relevance. The failure of a modeling toolchain is thus mistaken for the failure of a discipline.

As MBSE fails to become load-bearing in decision-making, some organizations are adapting by redefining the role of modeling itself. Rather than restoring engineering authority to the systems function, MBSE responsibilities were increasingly assigned to dedicated modelers whose primary task was to construct and maintain system models, not to make or defend technical decisions.

In this configuration, modelers function much like draftsmen for civil, mechanical, and electrical engineers. They translate architectural intent, design decisions, and requirements, developed elsewhere, into formal representations suitable for review, coordination, and compliance. The model records what has been decided, but does not participate meaningfully in deciding it.

This role separation is often efficient and, in some cases, intentional⁵. By decoupling modeling from decision authority, organizations reduce friction between modeling activities and ongoing engineering work. However, this separation also formalizes the distance between system models and engineering judgment. Modelers may possess deep familiarity with the system representation, yet lack both the mandate and the forum to influence trade-offs, assess risk, or challenge assumptions.

As a result, MBSE becomes a production activity rather than an engineering one. System models are evaluated on completeness, consistency, and traceability rather than on their ability to expose uncertainty, invalidate assumptions, or constrain design choices. The authority to shape the system migrates away from the systems function, while modeling becomes associated with documentation, compliance, and tool proficiency.

This organizational adaptation reinforces the perception that systems engineering is overhead. When models are produced by personnel who function as draftsmen rather than decision-makers, their limited influence appears intrinsic to the discipline rather than structural in origin. The underlying misalignment between modeling effort and engineering responsibility is obscured, and the cycle of diminished authority continues.

5. Boundaries, Evidence, and the SysML Question

5.1 SysML v1.x Was Never an Analytical Tool

SysML v1.x was designed as a **descriptive modeling language**, not as a simulation, optimization, or execution environment. It provides a structured way to represent system structure, interfaces, requirements, and relationships, with the explicit goal of improving communication and consistency across disciplines.

It was not intended to replace domain-specific analytical tools. Continuous-time simulation, control design, optimization, and numerical analysis are better served by environments such as MATLAB, Simulink, Modelica, or other specialized platforms. Expecting SysML to perform these functions is a category error.

This distinction matters. Critiques of MBSE often incorrectly frame the issue as a shortfall in technical capability. But in reality, the tools were being evaluated against expectations they were never designed to meet.

5.2 How Descriptive Models Became Overloaded

The problem arose when SysML models were treated not merely as coordination artifacts, but as *surrogates for system understanding*. As MBSE became conflated with systems engineering, the presence of a model was increasingly interpreted as evidence that the system had been analyzed, even when the analytical work occurred elsewhere.

In most organizations, simulations were still performed in MATLAB, spreadsheets, or bespoke tools. However, the results of those analyses were not structurally integrated into the system model in a way that made the model load-bearing. Instead, SysML artifacts often recorded assumptions, allocations, or outcomes after the fact.

This created a subtle but consequential inversion: the model appeared authoritative while remaining analytically passive.

5.3 Why Better Tools Does Not Fix the Problem

The availability of more powerful simulation and automation tools does not prevent MBSE from collapsing into documentation because the issue is not technical capability but role definition. SysML models are positioned as central

artifacts without being structurally connected to the analyses that generate engineering insight.

As descriptive fidelity increases, modeling effort grows. But if analytical authority remains elsewhere, decisional impact does not. The model becomes more complete and more polished while remaining optional to the core engineering workflow.

Improving the representational layer does not automatically make it load-bearing.

5.4 Evidence, Attribution, and the MBSE Value Narrative

Published accounts frequently report improved outcomes following MBSE adoption. However, these claims often conflate the presence of a system model with the analytical work performed in separate tools. Improvements driven by simulation, prototyping, or architectural trade studies are attributed to MBSE even when the model played no causal role.

This attribution error inflates expectations and obscures the true source of value. When MBSE is later expected to deliver similar results independently, it predictably falls short, not because the tools failed, but because they were credited with work they did not perform.

5.5 SysML v2 and Automation Raise the Ceiling, but Do Not Redefine the Role

SysML v2 significantly improves formal semantics, extensibility, and integration potential⁶. These advances make it easier to connect descriptive system models to analytical and executable toolchains. However, SysML v2 does not change the fundamental distinction between representing a system and reasoning about its behavior. Without deliberate changes to workflow, authority, and accountability, more capable modeling languages risk reinforcing the same pattern: better models, produced earlier, that still do not drive decisions.

A similar observation applies to the emerging use of agentic AI within MBSE environments. Agentic AI can reduce the friction associated with creating, maintaining, and navigating system models by automating consistency management, impact analysis, view generation, and traceability maintenance. In doing so, it can improve the usefulness of models as coordination artifacts and lower the cost of keeping them aligned with ongoing engineering work.

These advances do not alter the category distinction at the heart of this paper. Automating representational labor, even when highly sophisticated, does not substitute for analytical reasoning, trade-off evaluation, or decision-making under uncertainty. If AI outputs are treated as authoritative rather than as aids to human judgment, the same substitution error that burdened MBSE risks being repeated in automated form.

This dynamic can also be understood through the lens of calm technologies. Calm technologies recede into the background of a user's attention, allowing primary cognitive effort to remain focused on the task itself rather than on tool operation⁷. In contrast, most MBSE environments require sustained attention to notation, structure, and tool state. Effective use therefore demands specialization, shifting effort from system reasoning toward model management. Over time, this encourages organizational separation between those who operate the tools and those who perform analysis and make decisions.

Both SysML v2 and agentic AI expand what is technically possible within MBSE. Neither resolves the underlying misalignment between modeling effort and engineering authority. Without actively restoring systems engineering's role as a decision-making discipline, these advances will primarily improve the efficiency and polish of documentation rather than its decisional impact.

6. A Practical Path Forward: Using MBSE for What It Was Designed to Do

The problems described in this paper do not arise because MBSE is inherently flawed, nor because analytical tools are inadequate. They arise because MBSE has been asked to perform functions it was never designed to support. The solution is not more modeling, but clearer scope discipline.

MBSE should be used for representation, coordination, and traceability. Analysis, simulation, optimization, and evidence generation should remain in the tools built for those purposes.

6.1 Define Clear Boundaries Between Modeling and Analysis

Model-based systems engineering was created to provide a shared, structured representation of system intent: architecture, interfaces, requirements, and relationships. It was not designed to replace numerical simulation, optimization, control analysis, or statistical reasoning.

Analytical work should continue to be performed in environments such as MATLAB, Python, Modelica, spreadsheets, test rigs, and domain-specific tools. Requirements evidence and verification logic should continue to reside in tools such as Jama or equivalent systems.

Attempting to force these activities into MBSE environments increases effort without improving insight.

Management action: Explicitly define MBSE as a representational and coordination layer, not an analytical or simulation environment.

6.2 Use MBSE to Coordinate, Not to Compute

MBSE adds value when it provides a stable framework that connects analyses, requirements, interfaces, and decisions across disciplines. Its role is to reference, contextualize, and relate analytical results — not to reproduce them.

A system model should identify:

- Which analyses exist
- What assumptions they rely on
- Which decisions they inform
- Where authoritative results are maintained

The model should point to analytical artifacts, not attempt to subsume them.

Management action: Require MBSE artifacts to reference authoritative analyses rather than duplicate or re-implement them.

6.3 Keep Simulations and Evidence in the Tools Built for Them

Simulation environments and scripting languages exist because they provide capabilities MBSE tools intentionally do not: execution semantics, numerical stability, uncertainty handling, and rapid iteration.

Forcing simulation into MBSE tools creates brittle models that are expensive to maintain and rarely trusted. Allowing each tool to do what it does best improves both credibility and efficiency.

Management action: Protect analytical workflows from being absorbed into MBSE tools under the banner of “model-based.”

6.4 Make MBSE Support Decisions Without Owning Them

MBSE should support decision-making by making assumptions explicit, surfacing dependencies, and revealing interface impacts. It should not be treated as the source of analytical truth.

Systems engineers must retain responsibility for synthesizing analytical results from MATLAB, Python, test data, and other sources into coherent decisions. MBSE provides the map; analysis provides the terrain.

Management action: Hold engineers accountable for decisions, and models accountable for clarity and coherence.

6.5 Measure MBSE Success by Coordination Quality, Not Analytical Power

The value of MBSE should be evaluated by its effectiveness in:

- Reducing miscommunication
- Improving consistency across teams
- Exposing interface conflicts early
- Making assumptions and dependencies visible

Expecting MBSE to replace simulation or analysis sets it up to fail.

Management action: Evaluate MBSE on coordination and traceability outcomes, not on computational capability.

6.6 Treat Modelers as Engineering Support, Not Surrogate Engineers

It may appear contradictory to criticize the separation of modeling from engineering earlier in this paper and then recommend deliberate role separation here. The distinction is not between separation and integration, but between authority and support. The earlier failure mode emerged when modeling was separated from decision authority and allowed to define systems engineering by artifact production. In that configuration, models recorded decisions made elsewhere, and the discipline lost influence. The separation proposed here is different. It preserves decision authority within the systems engineering function while clarifying modeling as a representational and coordination role. Specialization is not the problem; displacement of engineering judgment is. When engineers own trade-offs and modelers maintain coherence, the boundary strengthens both roles rather than hollowing one out.

Dedicated modelers can provide real value when their role is clearly defined. Like draftsmen, they translate engineering intent into formal representations, maintain consistency, and support coordination across teams. However, they should not be expected to compensate for the absence of engineering authority or analytical capability. This distinction elevates both roles by allowing each to specialize rather than asking one to substitute for the other.

Many organizations have attempted to collapse these roles by treating modeling proficiency as equivalent to systems engineering authority. In practice, this approach has rarely succeeded. Instead of strengthening decision-making, it has typically increased modeling effort without increasing engineering influence.

Modelers should work in close proximity to decision-making engineers, with frequent interaction and well-defined interfaces that enable feedback, challenge, and iteration without creating dependency or role entanglement. Models should be updated during analysis and synthesis, not only after decisions are made.

The more effective approach is deliberate role separation with strong coordination: engineers own decisions and trade-offs; modelers maintain the coherence and fidelity of system representations.

Management action: Clarify role boundaries: modelers maintain representations; engineers own decisions. Measure success accordingly.

7. Implications for Practice

Systems engineering should be defined by the role it plays in shaping decisions, not by the artifacts it produces. Its value lies in framing problems, exposing trade-offs, managing uncertainty, and integrating perspectives across domains.

MBSE should be treated as supporting infrastructure. Where models directly contribute to analysis, verification, or decision support, they should be invested in and maintained. Where they primarily serve documentation or compliance needs, expectations should be scaled accordingly.

Organizations should evaluate value by asking:

- Did the model change the decision, or merely record it?
- Did it invalidate assumptions or reveal new constraints?
- Would removing it have altered the outcome?

If the answer is consistently negative, the problem is not tool proficiency but role alignment.

For systems engineering to function as intended, responsibility, incentives, and authority must align. Engineers accountable for system-level risk must have standing in decision forums and access to analytical resources beyond representational tools.

8. Conclusion

Systems engineering has not collapsed. What is collapsing is the expectation that Model-Based Systems Engineering, as commonly practiced, can serve as its replacement. Most organizations may not have mis-invested in MBSE, but they have overextended it beyond its intended role.

MBSE provides valuable coordination and representation capabilities, and emerging standards such as SysML v2 expand what is technically possible. However, no modeling language can substitute for judgment, synthesis, and decision-making under uncertainty.

The future of systems engineering depends on restoring the discipline's authority to frame problems and shape outcomes. MBSE should support that work where it adds value, and step aside where it does not.

By disentangling systems engineering from its proxy, organizations can recover the discipline's purpose, credibility, and impact without abandoning the tools that were meant to serve it.

9. References

1. Estefan JA, Weilkiens T. MBSE Methodologies. In: Madni AM, Augustine N, Sievers M, eds. *Handbook of Model-Based Systems Engineering*. Springer International Publishing; 2023:47-85. doi:10.1007/978-3-030-93582-5_12
2. Walden DD, International Council on Systems Engineering,. *INCOSE Systems Engineering Handbook*. Fifth.; 2023.
3. Friedenthal Sanford, Moore A, Steiner R. *A Practical Guide to SysML: The Systems Modeling Language*. Third.; 2014.

4. OMG Systems Modeling Language (SysML) Version 1.6. Object Management Group (OMG). 2019. Accessed February 16, 2026. <https://www.omg.org/spec/SysML/1.6/PDF>
5. Henderson K, McDermott T, Salado A. MBSE adoption experiences in organizations: Lessons learned. *Syst Engin.* 2024;27(1):214-239. doi:10.1002/sys.21717
6. OMG. Systems Modeling Language (SysML) v2 Specification. 2025. Accessed February 15, 2026. <https://www.omg.org/spec/UML/2.5.1/PDF>
7. The Coming of Age of Calm Technology. 1996. Accessed February 15, 2026. <https://www.kostakos.org/courses/ubicomp10S/papers/visions/weiser-96.pdf>