International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

From Requirements to Resilience: Architecting a Digital Thread Across Engineering and Supply Chain Using MBSE and PLM

Jasleen Singh Saini

Submitted:03/02/2026 **Revised**: 13/03/2026 **Accepted:** 22/03/2026

Abstract: Modern engineering enterprises invest heavily in CAD environments and PLM platforms, yet supply chains continue to fail at the point where design decisions meet operational execution. The root cause is rarely a logistics breakdown — it is an architectural one. Most enterprises begin the digital thread in CAD, after system intent has already been established informally, without structured traceability. The absence of Model-Based Systems Engineering (MBSE) at the origin of this thread means that requirements, functional allocations, and supply chain constraints never enter the product lifecycle in a machine-readable, queryable form. By the time geometry is committed, the decisions behind it are invisible to any governance mechanism. This paper proposes an enterprise architecture blueprint that repositions MBSE as the authoritative anchor of the digital thread, establishes a formal Semantic Mapping Framework to bridge the logical-to-physical boundary between MBSE and CAD, and uses PLM as the backbone that synchronizes both layers across the full product lifecycle. A RACI-based governance model enforces data ownership at every thread boundary. A five-level Digital Thread Maturity Model provides a structured adoption roadmap. The central argument is that supply chain resilience cannot be achieved operationally when it has not first been built architecturally — and that architecture begins in MBSE, not in CAD.

Keywords: *Digital Thread, MBSE, PLM, SysML, Supply Chain Resilience, BOM Transformation, Semantic Mapping, Digital Twin*

1. Why Digital Thread Must Start in MBSE

1.1 The Problem with Starting in CAD

For decades, the standard practice in engineering enterprises has been to treat CAD as the origin of the digital thread. Requirements live in Word documents or spreadsheets, design intent is captured in 3D models, and the supply chain receives a BOM export at the end of the process. This approach is understandable — it reflects how engineering tools were historically adopted — but it is fundamentally reactive. By the time the thread begins, the most important decisions have already been made without any structural traceability connecting them to downstream operations.

The consequences of this gap show up predictably. A voltage requirement changes late in a program, and no one immediately knows how many components are affected. A supplier is disqualified, and the team has to manually trace which assemblies depend on that part. A mass budget is quietly exceeded in CAD, and no one catches it until verification. These are not random failures — they are the direct result of starting the thread too late, after the architecture has already been committed without a queryable record of the decisions behind it [1].

1.2 Establishing the Authoritative Source of Truth

The right starting point for the digital thread is the logical system architecture, built in MBSE before physical constraints are introduced. Tools like Cameo Systems Modeler and IBM Rhapsody allow

---

Independent Researcher, USA

engineering teams to define what a system must do — and what constraints it must satisfy — in a structured, traceable, machine-readable form. This logical model becomes the Authoritative Source of Truth (ASoT): the single record against which every downstream design decision is measured [1].

What makes this powerful is traceability. In SysML, engineers build explicit chains linking Requirements → Functions → Logical Components. This means that if a "Power Consumption" requirement is tightened, every functional block that draws from that budget is immediately visible in the model. The impact is not reconstructed through manual review — it is queried directly from the architecture [3]. This is not a documentation improvement. It is a structural change in how engineering knowledge is held and interrogated.

1.3 Embedding Supply Chain Thinking Early

One of the more underappreciated capabilities of MBSE is the ability to embed supply chain metadata directly into SysML blocks at the logical design stage. Attributes like "Preferred Supplier," "Target Unit Cost," and "Approved Manufacturer List" can be encoded as tagged values within the model before any geometry exists. The model then functions as a design filter — it flags configurations that the supply chain cannot support before an engineer has spent time detailing a part that will never be procurable [3].

This is a meaningful shift in when supply chain intelligence enters the engineering process. Rather than being consulted after design is complete, sourcing constraints become part of the design space from the beginning.

1.4 Finding Failure Modes Before Parts Exist

Functional Hazard Analysis (FHA) and Failure Mode and Effects Analysis (FMEA) conducted at the logical block level identify downstream failure risks before a single component is modeled in 3D. When failures are mapped to functional blocks rather than physical parts, the design team retains the flexibility to resolve them through architectural choices — adding redundancy, rerouting interfaces, restructuring subsystem boundaries — rather than being forced into late-stage component substitutions that disrupt both design stability and supply chain planning [1].

2. Translating System Intent to CAD Realization

2.1 Where Data Integrity Most Often Breaks Down

The transition from the logical domain (MBSE) to the physical domain (CAD) is, in practice, the most fragile point in the entire digital thread. On one side of this boundary, system intent is expressed in SysML with formal semantics — structured, queryable, governed. On the other side sits a CAD environment that understands geometry, mass, and material, but has no native concept of a system requirement or a functional block. Without a formal translation mechanism, data integrity degrades quickly. Engineers manually re-enter specifications. Constraints drift. The link between what the system was supposed to do and what the CAD model actually represents becomes invisible to any automated governance tool [5].

This boundary is referred to in the literature as the "Semantic Gap," and it requires a formal bridging framework — not a better file export format, but a structured semantic translation mechanism [4].

2.2 The Semantic Mapping Framework

The Semantic Mapping Framework proposed here operates through four linked mechanisms, each targeting a specific failure mode at the logical-to-physical boundary.

1. **Logical-to-Physical (L2P) Link:** Every SysML Block is assigned a Globally Unique Identifier (GUID) at creation. This GUID is pushed into the corresponding CAD file's metadata field, creating a persistent machine-readable link between the logical specification and its physical realization. Critically, this link survives model revisions, file migrations, and tool upgrades — as long as the GUID is treated as an immutable system identifier rather than a convenience label [4].
2. **Automated Attribute Inheritance:** Technical specifications defined in MBSE — such as OperatingTemperature < 85°C, MaxMass = 2.4 kg, or OperatingVoltage = 24V — are automatically written into CAD file properties when the L2P link is instantiated. This eliminates the need for manual re-entry and removes the most common source of specification drift between the engineering and physical layers [5].
3. **Architecture Drift Governance:** If a CAD designer increases component mass beyond the "Mass Budget" defined in the MBSE model, a Compliance Flag is automatically triggered in PLM.

The change is not blocked outright — engineering judgment is preserved — but it cannot be silently committed. The variance is formally logged and must be either approved through an Architectural Variance process or resolved through design correction.

4. **150% to 100% BOM Resolution:** MBSE defines the full configuration space of the product — the 150% BOM — which captures all possible variants, optional configurations, and conditional assemblies. CAD realizes a specific, context-appropriate instance — the 100% BOM — for a defined configuration [1]. This distinction preserves configurability at the architecture level while enabling precise execution at the physical level.

2.3 MBSE-to-CAD Attribute Translation

Mapping Layer	Source Attribute (MBSE)	Target Attribute (CAD)
Thermal Compliance	OperatingTemperature < 85°C	CAD Property: Max_Op_Temp
Mass Budget	MaxMass = 2.4 kg	CAD Property: Allocated_Mass
Voltage Specification	OperatingVoltage = 24V	CAD Property: Input_Voltage
Supplier Constraint	PreferredSupplier = [AML List]	CAD Property: AML_Reference
Configuration Scope	150% Variant Rule	100% Config Instance ID

Table 1: Semantic Mapping Framework — MBSE-to-CAD Attribute Translation

3. PLM as the Digital Thread Backbone

3.1 More Than a File Repository

PLM systems are frequently treated as little more than sophisticated file management platforms — a place to store CAD models, manage revisions, and route approvals. This framing undersells what PLM can and should do in a mature digital thread. When properly configured, PLM becomes the orchestration engine that manages the evolutionary state of the product across its entire lifecycle, maintaining the integrity of system intent as it transforms from engineering specification into manufacturing instruction into service record [2].

The key to this role is PLM's ability to maintain three simultaneous, synchronized views of the product structure — each one serving a distinct downstream consumer with a different organizational concern [6].

3.2 The Three BOM Views

The Engineering BOM (EBOM) is organized by functional system decomposition, derived directly from the MBSE/CAD layer. It reflects how engineers think about the product — by subsystem, by function, by interface. The Manufacturing BOM (MBOM) is reorganized by assembly process sequence to serve production planning. The same components appear, but in the order that manufacturing needs them. The Service BOM (SBOM) is restructured around spare parts logistics and field serviceability — organized the way a service technician or logistics planner needs to see the product.

Each of these transformations must be governed as a formal process. Treating them as informal spreadsheet exercises is one of the most common sources of downstream data inconsistency in engineering enterprises [2].

3.3 How Change Propagates Through the Backbone

The PLM backbone establishes a chain of linked data objects that make change propagation traceable and controllable. Every MBSE Model ID is linked to a PLM Part Master — the foundational record from which all downstream BOM structures are derived. When a change is committed in the MBSE layer, PLM automatically performs a "Where-Used" analysis across all linked CAD assemblies and BOM views. Every affected configuration is surfaced for review before the change is released. Nothing propagates silently [6].

3.4 BOM Transformation Governance

BOM Type	Organizational Basis	Primary Consumer	Governance Owner
EBOM	Functional system decomposition	Design engineering	MBSE Architect
MBOM	Assembly process sequence	Manufacturing planning	Manufacturing Engineer
SBOM	Spare parts and field logistics	Supply chain / service	SCM / After-Sales
xBOM (Variant)	Configuration rules from 150% model	Program / contracts	PLM Config. Manager
Regulatory BOM	Material compliance — RoHS / REACH	Legal / compliance	PLM Analyst

Table 2: BOM Transformation Governance — Structure, Ownership, and Consumption

4. Extending the Thread into the Supply Chain

4.1 From Data Delivery to Data Participation

There is a meaningful difference between a supply chain that receives engineering data and one that participates in engineering decisions. Most enterprises today operate in the first mode — the supply chain is notified of changes after design is complete, given a BOM to work from, and expected to execute. The closed-loop digital thread makes the second mode possible. When the thread extends from PLM into ERP systems and supplier portals, supply chain intelligence becomes available to design teams at the moment decisions are being made, not after they have been locked [7].

This shift — from data delivery to data participation — is arguably the highest-value outcome of a mature digital thread implementation [8]. It requires that supply chain systems consume structured attributes from the PLM Part Master in real time, rather than re-keying data from PDF drawings or email attachments.

4.2 What Automated Impact Analysis Actually Looks Like

The practical value of this integration is most clearly illustrated by a concrete example. Consider a program-level decision to switch from a 12V to a 24V electrical architecture — a seemingly straightforward requirement change that, in a disconnected enterprise, could take weeks to fully assess.

In a closed-loop digital thread, the process looks very different:

1. **Identification:** The thread traverses the MBSE model and identifies all functional blocks allocated to the 12V subsystem. This query propagates through L2P links to surface every associated CAD component and PLM part record automatically.
2. **Supply Chain Metrics Calculation:** For each affected component, the system computes the operational consequences in real time — Does the replacement 24V component carry a 20-week procurement lead time that threatens the program schedule? Is there existing inventory — say, $500,000 worth of 12V motors — that becomes obsolete under the new architecture? Is the new component available from only one qualified supplier, creating a supply concentration risk?
3. **Trade-off Support:** The design team receives this operational data before the architectural decision is finalized [7]. The supply chain impact becomes a design input rather than a downstream discovery.

4.3 Supply Chain Impact Analysis Framework

Change Event	Thread Layer Triggered	Supply Chain Metric Computed	Decision Output
Voltage Architecture Shift	MBSE → PLM → ERP	Lead time delta, inventory write-off	Architecture trade-off
Mass Budget Exceedance	CAD → PLM Compliance Flag	Material cost variance, supplier qualification	Variance approval
Supplier Disqualification	SCM → PLM Part Master	Affected assemblies count, re-qualification timeline	Sourcing re-plan
Requirement Tolerance Change	MBSE → L2P → CAD	Process capability delta, scrap rate forecast	Design or process change
Configuration Variant Addition	MBSE 150% BOM → EBOM	BOM complexity cost, kit proliferation risk	Program config. decision

Table 3: Supply Chain Impact Analysis — Change Events and Digital Thread Response

5. Governance and the RACI Model

5.1 Why Governance Cannot Be an Afterthought

A digital thread without governance is not a thread — it is a collection of loosely connected data artifacts that will gradually diverge from each other as the program progresses. Without clear authority assignment, the ASoT loses its authoritative status. Designers override requirements informally. Part records accumulate unsanctioned changes. The PLM backbone becomes a historical archive rather than a live governance mechanism [8].

The RACI model provides the structural solution. By formally assigning data ownership across the four primary roles in the digital thread enterprise — the Systems Architect (MBSE), the Mechanical Designer (CAD), the PLM Analyst, and the Supply Chain Manager (SCM) — every data object in the thread has a clear owner, a clear set of consulted parties, and a clear boundary around who can change what [9].

5.2 Data Ownership Matrix

Data Object	MBSE (Architect)	CAD (Designer)	PLM (Analyst)	SCM (Buyer)
System Requirement	A/R	C	I	I
Logical Interface	A/R	C	I	I
Physical Geometry	I	A/R	C	I
Part Material / Spec	C	R	A	C
Sourcing Status	I	I	C	A/R

Table 4: Data Ownership Matrix — RACI Assignments Across Digital Thread Roles (A = Accountable, R = Responsible, C = Consulted, I = Informed)

5.3 The Fundamental Governance Rule

The governance architecture enforces one non-negotiable rule: no change to Physical Geometry that violates a System Requirement can be committed without a formal Architectural Variance Approval. This rule is enforced at the PLM workflow level through automated compliance checking against the MBSE model at the point of

CAD check-in [9]. It is not a policy that relies on engineers remembering to follow a process — it is a technical control embedded in the thread's data infrastructure.

This matters because the most damaging governance failures in engineering enterprises are not the ones where someone deliberately bypasses a requirement. They are the quiet ones — an informal decision that seemed reasonable at the time, made without full awareness of the system constraint it violated, discovered only during verification or, worse, after delivery.

6. Digital Thread Maturity Model

6.1 Maturity Is Not Binary

Organizations do not jump from disconnected silos to a fully cognitive digital thread in a single program. Maturity progresses through identifiable capability levels, each requiring specific technical investments and — more importantly — specific organizational changes [10]. The five-level model presented here is intended as both an assessment tool and a planning framework. Understanding where an organization currently sits on this scale is the prerequisite for making targeted, realistic investments in advancing it.

6.2 The Five Maturity Levels

Level	Name	Technical Capability
1	CAD-Centric	Disconnected silos; manual BOM transfers; requirements stored in documents
2	PLM-Controlled	Linked CAD and EBOM; electronic change workflows; basic part master governance
3	MBSE-Connected	MBSE logic drives CAD and PLM; digital requirements with traceability; formal L2P links
4	Closed-Loop	Supply chain data (cost, lead time, inventory) feeds design trade-off analysis in real time
5	Cognitive Thread	AI predicts supply disruptions, flags requirement-design drift, and proposes architecture alternatives

Table 5: Five Maturity levels

6.3 What the Transitions Actually Require

Most industrial enterprises currently operate at Level 1 or Level 2. The move from Level 2 to Level 3 is often where programs stall — not because the toolchain integration is technically impossible, but because it requires a fundamental reorientation of engineering authority. Systems architects must take on the role of data stewards, not just analysts. Requirements must be treated as live, queryable objects rather than documents that are read once and filed [10].

The move from Level 4 to Level 5 is a different kind of challenge. It depends on AI models trained on enterprise-specific product and supply chain data — a capability that is genuinely emergent across the industry today. Organizations at Level 4 are already achieving substantial value; Level 5 represents the horizon that current research and toolchain development are actively working toward [10].

Conclusion

The phrase "from requirements to resilience" is not a metaphor — it is a description of a traceable data path that either exists in an enterprise or does not. When it does not exist, supply chain resilience is an outcome that has to be achieved reactively, through firefighting, expediting, and late-stage redesign. When it does exist, resilience is an architectural property of the system — built in from the first requirement, maintained through every design decision, and queryable at any point in the lifecycle.

The framework presented in this paper argues that achieving this requires three things to be true simultaneously. The digital thread must be anchored in MBSE, so that system intent is held in a structured, traceable form before physical constraints are introduced. The Semantic Mapping Framework must govern the logical-to-physical boundary, so that data integrity is preserved rather than assumed. And governance must be embedded in the thread's data infrastructure, so that authority is enforced rather than hoped for.

When these conditions are met, a requirement change does not trigger a manual impact assessment. It triggers an automated query that surfaces affected components, computes supply chain consequences, and places the relevant trade-off data in front of the engineering team before the decision is finalized. The supply chain stops being a downstream victim of design decisions and becomes an integrated participant in making them. That is what resilience as an architectural property actually looks like in practice.

References

[1] Sanford Friedenthal, Alan Moore and Rick Steiner, “A Practical Guide to SysML: The Systems Modeling Language, 3rd edn., Amsterdam, Netherlands: Elsevier/Morgan Kaufmann, 2015. [Online]. Available: https://www.sciencedirect.com/book/monograph/9780128002025/a-practical-guide-to-sysml

[2] John Stark, “Product Lifecycle Management (Volume 1,” 21st Century Paradigm for Product Realisation, 5th Edn., Springer, 2022. [Online]. Available: https://link.springer.com/book/10.1007/978-3-030-98578-3

[3] Victor Singh and Karen E. Willcox, "Engineering Design with Digital Thread," American Institute of Aeronautics and Astronautics. [Online]. Available: https://kiwi.oden.utexas.edu/papers/Engineering-design-digital-thread-Singh-Willcox.pdf

[4] Angira Sharma, et al., "Digital Twins: State of the art theory and practice, challenges, and open research questions," Journal of Industrial Information Integration, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2452414X22000516

[5] Roberto Minerva, et al., "Digital Twin in the IoT Context: A Survey on Technical Features, Scenarios, and Architectural Models," Proceedings of the IEEE, 2020. [Online]. Available: https://ieeexplore.ieee.org/document/9120192

[6] Michale Riesener, et al., "The Digital Shadow as Enabler for Data Analytics in Product Life Cycle Management," Procedia CIRP,2019. [Online]. Available: https://d-nb.info/1194239307/34

[7] Aidan Fuller, et al., "Digital Twin: Enabling Technologies, Challenges and Open Research," IEEE Access, 2020. [Online]. Available: https://ieeexplore.ieee.org/document/9103025

[8] Igiri Onaji, et al., "Digital twin in manufacturing: conceptual framework and case studies," International Journal of Computer Integrated Manufacturing, 2022. [Online]. Available: https://www.tandfonline.com/doi/epdf/10.1080/0951192X.2022.2027014?needAccess=true

[9] Jiewu Leng, et al., "Digital twin-driven manufacturing cyber-physical system for parallel controlling of smart workshop," Journal of Ambient Intelligence and Humanized Computing, 2019. [Online]. Available: https://link.springer.com/article/10.1007/s12652-018-0881-5

[10] Shouxuan Wu, et al., "Cognitive digital thread tool-chain for model versioning in model-based systems engineering," Advanced Engineering Informatics, 2025. [Online]. Available: https://www.sciencedirect.com/science/article/abs/pii/S1474034625003830