INCOSE PUBLICATIONS

CHAPTER CONFERENCE

Approach of Product of Product Lines for Complex Aerospace Systems Development

INCOSE MBSE SUMMIT · 2025

AUTHORS
Sandesh Puranik, Mohan Mishra, Dinesh Kushwaha, and Masood Ahmed

CHAPTER
India Chapter

PUBLICATION DATE
July, 2026

SUBMISSION #
36

INCOSE INDIA MBSE Summit 2025
14-15 June 2025

Approach of Product of Product Lines for Complex Aerospace Systems Development.

Sandesh Puranik
Collins Aerospace
North Gate office, Yelahanka, Bangalore - 560064
7019818190
Sandesh.puranik@collins.com

Mohan Mishra
Collins Aerospace
North Gate office, Yelahanka, Bangalore
560064
9885389942
mohan.mishra@collins.com

Dinesh Kushwaha
Collins Aerospace
North Gate office, Yelahanka, Bangalore -
560064
9845871310
dinesh.kumar.kushwaha@collins.com

Masood Ahmed
Collins Aerospace
DLF, Gachibowli, Hyderabad - 500032
7893898081
masood.ahmed.mohammed@collins.com

Abstract. As we often refer to term "System of Systems" (SoS) which encompasses multiple independent systems, similarly there of multiple Large and complex systems, such as aircraft which is made up of multiple intricate subsystems. Within an organization and/or across organizations, various development teams work on these subsystems, and each subsystem typically has its own product line to support different OEM requirements. However, the final decision on which Subsystem Configurations will be used is made at a higher system level. This creates a challenge where multiple product lines must be managed together, ensuring compatibility, consistency, and integration across the entire system, the traditional Product Line Engineering (PLE) approach is effective for managing configurations within an individual subsystem. However, when several subsystem product lines must work together within a broader system context, traditional PLE approaches fall short. Misalignment in interfaces, redundant functionalities, or inconsistent configuration decisions across teams can introduce integration delays and increase system complexity. Therefore, a new approach is needed to manage the collective behavior and coordination of these interconnected product lines. This paper explores these challenges and proposes a **Product Line of Product Lines (PoPL)** approach to systematically manage and coordinate multiple interconnected product lines in large-scale aerospace system development.

Keywords. Product Line Engineering (PLE), Product of Product Lines (PoPL), Subsystem Configuration, Multiple Product Lines.

This document does not contain any Export Controlled Technical Data.

Introduction

Modern aerospace systems are composed of multiple subsystems, each of which is complex and evolves independently to meet diverse operational and regulatory requirements. The subsystems often have their own product lines, offering multiple configurations to serve different OEM requirements and mission needs [4][5]. What makes this landscape even more intricate is the structure of the aerospace ecosystem itself, which is rarely centralized. In reality, the development of these subsystems spans across multiple organizations, suppliers, and geographically distributed teams, each operating with its own design processes, tool chains, and delivery commitments [3][6].

A single system say, a Avionics may be the result of coordinated effort among different vendors providing the Aircraft management systems, Monitoring System, and Flight Control System, each as distinct product lines [2][4]. In practice, most aerospace giants do not supply just a single component or subsystem to their customers. These offerings are rarely isolated; they are intended to function cohesively as part of a broader platform [7][8]. As a result, internal product lines must be structured in a layered and interconnected fashion, allowing the organization to reuse capabilities, synchronize development cycles, and respond to changes efficiently [5].

For example, a company delivering multiple Sub-systems must ensure these are compatible not only with each other but also with the broader System suite deployed across different aircraft programs [4][7]. This necessitates well-defined internal architecture, where each product line maintains autonomy over its feature model and configuration management, but is also aligned to interface standards, common services, and system-level requirements [1][3]. Without such alignment, even organizations with strong individual product lines risk inefficiencies, duplicated effort, or integration delays when contributing to a shared system.

These teams do not function in isolation; they work within an interdependent network where decisions made at one level can ripple across the entire architecture. A configuration choice at the system level such as the selection of a specific control logic for a commercial aircraft variant can influence the configurations of downstream components [3][6]. At the same time, subsystem teams may consider the system they support, such as a specific aircraft platform, as merely one of several consumers of their product line, which must also cater to other aircraft types like helicopters or business jets [4][5]. The final decision on which configuration to integrate is typically made at a higher system level or based on customer-opted feature/function at aircraft level.

Product Line Engineering (PLE) helps in managing configurations within a single product line, but the growing demand for multiple configurations across different programs creates a more interconnected structure [1][5]. Each development team manages its own feature model, defining configurations based on its own requirements [2][3].

To address these challenges, this paper introduces a Product of Product Lines (PoPL) approach. This approach provides a structured method for managing multiple subsystem product lines while ensuring that different configurations align with overall system needs. The following sections explore the key challenges in managing multiple product lines, describe the PoPL approach, and outline how it can be applied to aerospace system development.

Methodology

To implement the Product of Product Lines (PoPL) approach effectively in complex aerospace programs, a structured methodology is essential. This methodology is centered around managing interdependencies and ensuring alignment across different layers of product lines, while maintaining the autonomy and flexibility of individual subsystem development. The following four key steps (as depicted in Figure 1) outline the practical execution of the PoPL strategy:

1. **Establish a Multi-Level Product Line Structure**

The foundation of PoPL begins with identifying and organizing product lines in a hierarchical structure starting from the system-level Avionics down to such as Aircraft management systems, Monitoring System & Flight Control System. This structure provides clarity on ownership, configuration scope, and interfaces at each level. It allows each team or organizational unit to evolve their product lines independently while staying aligned with higher-level system goals. For instance, the Avionics product line may span across various aircraft types, and its subsystems can be reused across those platforms but configured differently based on aircraft needs.

2. **Manage Dependencies Across Levels**

A critical enabler of PoPL is the identification and tracing of dependencies between product lines. Configuration decisions made at the system level must influence and constrain valid configurations at the subsystem level. For example, an Avionics configured for a commercial aircraft may require specific display formats and redundancy features that must be enforced within the corresponding Aircraft management systems and Monitoring System variants. These cross-level interactions must be explicitly modeled to avoid incompatible configurations and to ensure traceability across the architecture. This step enforces architectural coherence while allowing subsystem teams to stay responsive to multiple programs.

3. **Implement Configuration and Variation Management**

To support flexible yet controlled reuse, variation points and configuration rules must be defined using structured feature models at each level of the hierarchy. Each product line maintains its own feature model to capture permissible combinations, constraints, and options. These models must also support integration with system-level configurations. This layered variation management allows organizations to efficiently produce tailored solutions such as variants suited for the Customer 1, Customer 2 & Customer 3 without reinventing core subsystems. Importantly, this promotes scalable reuse and reduces duplication of engineering effort across programs.

4. **Enable Automation and Tooling Support**

Finally, to operationalize the above steps and ensure efficiency at scale, robust MBSE (Model-Based Systems Engineering) tooling is essential. These tools help automate synchronization across product lines, enforce configuration rules, and validate dependencies. Automation reduces human error, accelerates variant generation, and ensures that feature compatibility and architectural integrity are maintained even as configurations evolve. Especially in a multi-team environment where different product lines are developed concurrently, automation ensures a consistent and up-to-date integration flow.

Together, these steps form a cohesive methodology that supports the scalable and flexible management of multiple product lines in a modern aerospace ecosystem. By combining hierarchical structuring, dependency management, controlled variation, and tooling automation, the PoPL approach lays the groundwork for systems that are not only modular and reusable, but also integratable and reliable.

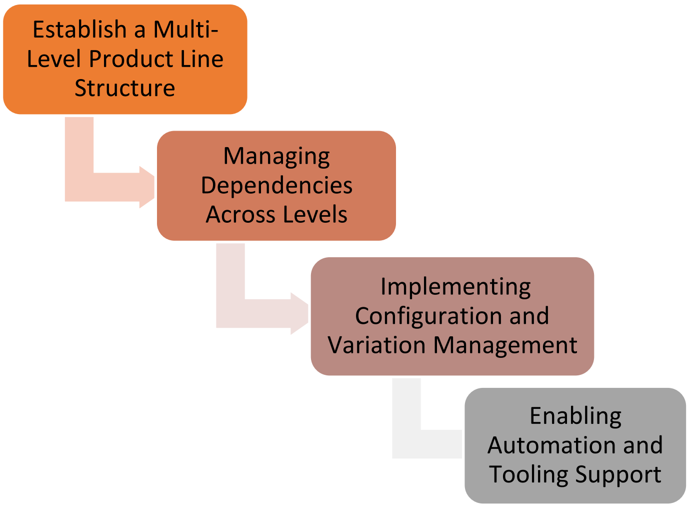


Figure 1. Methodology

Managing multiple product lines in complex aerospace systems requires a structured approach that extends beyond traditional Product Line Engineering (PLE). While PLE is effective for handling variability within a single product line, modern aerospace systems demand a more interconnected approach due to the increasing number of dynamic configurations. The Product of Product Lines (PoPL) approach helps address this challenge by managing dependencies across multiple product lines, ensuring seamless integration from components to full systems.

Each product line operates with its own feature model, defining the available configurations and constraints. At the system level, overarching configurations such as those for an Higher System dictate requirements for subsystems. These subsystems, in turn, define their own feature models, catering to multiple applications. For example, Aircraft management systems, may be designed to support both commercial aircraft and helicopters, but with variations in control laws, redundancy levels, or interface requirements. Elements such as Monitoring System are developed independently but must be compatible with multiple subsystems Configurations. This layered approach allows the same component or subsystem to be reused across different programs, reducing redundant development while maintaining flexibility. In practice, this means that reuse is not just technical, but strategic. When feature models and variation rules are well-integrated across levels, an organization can rapidly spin off new offerings say, adapting an existing military sub-system to a commercial platform or deploying a proven Subsystem variant into a new UAV system. Instead of re-engineering or maintaining dozens of independent baselines, PoPL allows for scalable reuse while still addressing platform-specific nuances.

Ultimately, the PoPL methodology is about balancing independence with alignment. Each team retains control over its own product line critical for innovation and responsiveness but does so within a governed framework that ensures cross-system compatibility. This methodology, when embedded within the organizational development process and supported by robust MBSE tooling, becomes a catalyst for both operational efficiency and system robustness in complex aerospace programs.

For example, To illustrate the application of the PoPL approach, consider a Avionics that serves as a high-level product line catering to different customer categories supporting different platforms. While these platforms share common Avionics capabilities, each variant imposes unique operational constraints and regulatory requirements. The Avionics in this scenario is composed of three major subsystems: Aircraft management systems, Monitoring System & Flight Control System.

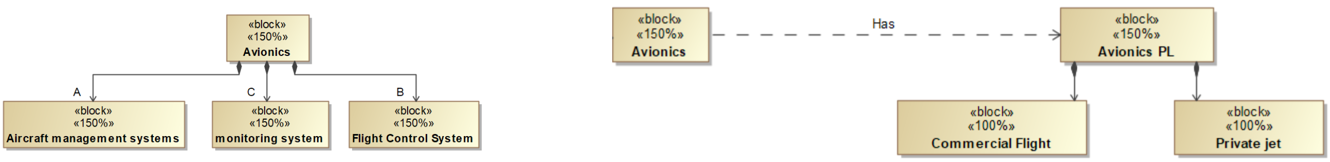


Figure 2. Example – Avionics PL

These subsystems might have their own product lines to supmultiple configurations are likely to be offered in order to meet the diverse requirements of various aircraft OEMs.

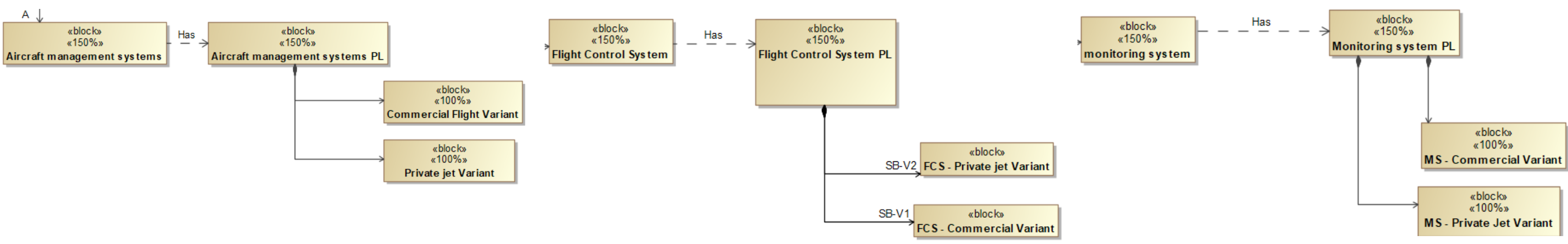


Figure 3. Different PL's of Various Sub-Systems

The below matrix shows the different configurations of System A namely: Commercial Flight & Private jet, each assembled using available configurations from the respective subsystem product lines. These configurations are tailored to meet performance, integration, and certification needs across different aircraft categories.

Table 1. Usage of Configurations of Avionics System

Legend: ↗ Usage		Aircraft management systems PL	SA-V1 : Commercial Flight Varian	SA-V2 : Private jet Variant	Flight Control System PL	SB-V1 : FCS - Commercial Varian	SB-V2 : FCS - Private jet Variant	Monitoring system PL	SC-V1 : MS - Commercial Variant	SC-V2 : MS - Private Jet Variant
For MBSE Summit			1	1		1	1		1	1
Commercial Flight	3	1	↗		1	↗		1	↗	
Private jet	3	1		↗	1		↗	1		↗

The matrix below shows deployment of the respective Avionics configurations to meet specific customer platform needs such as Customer 1, Customer 2, and Customer 3. In real-world implementation, each product line is typically managed in separate and dedicated models. However, the integrated representation shown here is a conceptual abstraction to illustrate the methodology, not a literal depiction of PLE/MBSE tool implementation.

Table 2. Avionics Configurations Satisfying Different Customer Needs

Legend
↗ Realization

	Customers	Customer 1	Customer 2	Customer 3
For MBSE Summit		1	1	1
Avionics PL				
V1 : Commercial Flight	2	↗	↗	
V2 : Private jet	1			↗

One of the key challenges in this structure is managing dependencies across different product lines. Since subsystems often serve multiple programs, their configurations cannot be dictated by a single system-level requirement. Instead, each subsystem team must maintain internal configurability to serve multiple customers. For instance, an Aircraft management systems team may treat an Avionics team as just one of its customers, while also supporting other aircraft programs that use different Avionics architectures. Similarly, at the component level, a component may be designed for multiple customers for different use cases, requiring flexible integration points.

"Conquering Complexity with Simplicity" – The approach works most effective when integrated with a Product Line Engineering (PLE) tool. By leveraging a factory configurator, this approach enables the generation of specific product instances based on selections made from a defined feature catalogue. In theory, the configurator processes the permissible permutations and combinations, applies predefined rules, and rapidly produces the desired product instance. This approach not only streamlines initial product generation but also significantly enhances the ability to accommodate customer-driven changes allowing for quick reconfiguration and regeneration of updated product instances within the defined constraints.

Benefits

The Product of Product Lines (PoPL) methodology brings numerous benefits to the development and management of complex aerospace systems:

1. **Scalability and Reuse**: By structuring product lines in a hierarchical manner, organizations can maximize reuse of components, subsystems, and even entire configurations across multiple platforms. For example, a Aircraft management systems tailored for a business jet can be adapted with minimal changes for a helicopter variant, leveraging the same core feature model.

2. **Reduced Time-to-Market**: Through the reuse of validated models and modular configurations, development teams can significantly cut down on engineering time for new customer programs. Instead of starting from scratch, they modify existing product line elements based on system-level needs.

3. **Improved Governance**: When dependencies between product lines are managed explicitly, teams gain better traceability from system-level features down to specific subsystem

implementations. This helps in maintaining compliance with safety regulations and certification requirements, which are critical in aerospace.

4. **Aligned Multi-Team Collaboration**: PoPL promotes transparency and coordination between teams working on different subsystems. A shared understanding of how configurations flow from system to component level supports better alignment across engineering functions, even across different organizations or suppliers.

5. **Configurable Solutions at Scale**: The ability to handle complex variability through structured feature models enables the organization to respond to diverse OEM and operator requirements without exponentially increasing development complexity.

Challenges

1. **Initial Setup Complexity**: Defining multi-level feature models and establishing clear interdependencies can be resource-intensive, especially for legacy programs that were not originally designed with modularity in mind.

2. **Tooling and Integration**: Realizing the PoPL methodology effectively requires robust tool support, including MBSE platforms that can handle multi-level modeling, automated variant generation, and rule enforcement. Many organizations still rely on fragmented toolchains, which can hinder seamless integration.

3. **Cultural and Organizational Barriers**: Teams are often accustomed to working in silos, especially when they serve different platforms or customers. Shifting to a PoPL mindset requires buy-in across departments and often a rethinking of ownership boundaries and communication protocols.

4. **Change Propagation and Impact Analysis**: When a change is made in a higher-level product line or system configuration, assessing its impact across all affected subsystems can be nontrivial. Without robust dependency tracing and impact analysis, the risk of unintended downstream effects remains high.

5. **Maintaining Configurability vs. Simplicity**: As configurability increases, so does model complexity. It becomes crucial to strike a balance between supporting variability and keeping the product line manageable and testable. Overly complex configurations may lead to brittle models and difficulty in validation.

Recommendations

1. **Think in Layers, Not Just Features:** Instead of treating each product line in isolation, organizations should design them as interconnected layers. This ensures that higher-level systems dictate configurations for subsystems and components while still allowing flexibility for different platforms like commercial aircraft, helicopters, and UAVs.
2. **Create a Common Language Across Teams**: Managing multiple product lines means different teams must work together without miscommunication. Standardizing how configurations are defined and tracked across teams will make integration smoother and reduce costly errors.
3. **Use MBSE for Better Visibility**: Model-Based Systems Engineering (MBSE) helps teams understand how one configuration choice impacts another. It allows engineers to see dependencies rather than just document them, making traceability more intuitive.
4. **Automate Where It Makes Sense**: Configuration management across multiple product lines can become overwhelming if done manually. Smart automation, especially with AI-driven tools, can help teams quickly assess viable configurations, reducing time spent on iterative testing and validation.
5. **Encourage Cross-Team Collaboration**: No single team owns an entire aircraft; it's a collective effort. Organizations should foster cross-functional collaboration, ensuring that different

teams understand how their choices impact the larger system. Governance structures should support, not restrict, innovation by enabling seamless decision-making.

Conclusion

The Product of Product Lines (PoPL) approach presents a structured methodology for managing complex aerospace systems where multiple subsystems and components have their own independent product lines. Traditional Product Line Engineering (PLE) methods efficiently handle configurations within a single product line, but as the demand for diverse configurations increases, organizations must manage interconnected product lines across different levels systems, subsystems, and components. The methodology outlined in this paper provides a systematic way to model and manage multi-tier product lines, reducing complexity while enhancing traceability and integration. Future research should explore automation tools and AI-driven decision-making to further streamline PoPL adoption in large-scale aerospace development programs. PoPL enables seamless configuration management across multiple layers.

References

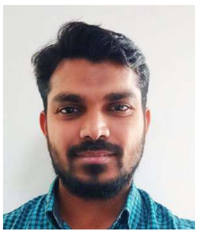
1. Gregg, S. P., Scharadin, R., & Clements, P. C. (2019). The best of both worlds: Agile development meets product line engineering at Lockheed Martin. *INSIGHT*, 22(2), 25–31. https://doi.org/10.1002/inst.12243
2. Hause, M., & Hummell, J. (2019). Model-based product line engineering–enabling product families with variants. *INSIGHT*, 22(2), 43–48. https://doi.org/10.1002/inst.12245INCOSE Online Library
3. Darbin, R., Pitz, R., Taylor, M., Teaff, J., Young, B., Wilson, B., Hartley, D., & Clements, P. (2021). Feature-based product line engineering: An essential ingredient in agile acquisition. *INCOSE International Symposium*, 31(1), 956–971. https://doi.org/10.1002/j.2334-5837.2021.00880.xINCOSE Online Library+1ResearchGate+1
4. Guo, C., Teng, J., Gao, M., & Jiang, Y. (2021). The research of aerospace software product line engineering. In Y. Wang, L. Xu, Y. Yan, & J. Zou (Eds.), *Signal and Information Processing, Networking and Computers* (pp. 357–364). Springer, Singapore. https://doi.org/10.1007/978-981-33-4102-9_44
5. Bilic, D., Sundmark, D., Afzal, W., Wallin, P., Causevic, A., Amlinger, C., & Barkah, D. (2020). Towards a model-driven product line engineering process: An industrial case study. *Proceedings of the 13th Innovations in Software Engineering Conference*, 1–8. https://doi.org/10.1145/3385032.3385043
6. Dove, R., Schindel, W. D., & Hartney, R. W. (2017). Case Study: Agile Hardware/Firmware/Software Product Line Engineering at Rockwell Collins. Presented at the 11th Annual IEEE International Systems Conference. Retrieved from https://rockwellcollinsthoughtleadership.wordpress.com/2017/04/29/case-study-agile-hardwarefirmwaresoftware-product-line-engineering-at-rockwell-collins-2/
7. MacKnight, E. (2021, May). *General Electric Aviation's use of product line engineering and model-based systems engineering*. *BigLever Software.* Retrieved from https://biglever.com/momentum-blog-series4/
8. Mohan, K., & Ramesh, B. (2020). Towards a model-driven product line engineering process: An industrial case study. In *Proceedings of the 13th Innovations in Software Engineering Conference* (pp. 1–10). ACM. https://doi.org/10.1145/3385032.3385043
9. Flores, R., Krueger, C. W., & Clements, P. C. (2012). Mega-scale product line engineering at General Motors. In *Proceedings of the 16th International Software Product Line Conference (SPLC 2012)* (pp. 259–269). ACM. https://doi.org/10.1145/2362536.2362571
10. Wikipedia contributors. (n.d.). *Avionics*. In *Wikipedia, The Free Encyclopedia*. Retrieved May 26, 2025, from https://en.wikipedia.org/wiki/Avionics

Biography

Sandesh Puranik is a Lead Engineer at Collins Aerospace, RTX, with over 3 years of experience in Systems Engineering and Model-Based Systems Engineering (MBSE) within the aerospace domain. He is also an INCOSE Certified Associate Systems Engineering Professional (ASEP). Over the course of career, he has worked with a range of MBSE, MBPLE tools and methodologies and has been actively involved across multiple phases of the system development lifecycle in alignment with the Vee model. He is a passionate Aerospace engineer and an active member of INCOSE.

Mohan Mishra is a Principal Engineer at Collins Aerospace, RTX, with 13 years of experience in product development, actively involved across all phases of the system development lifecycle. His core expertise includes Model-Based Engineering (MBE), Digital Engineering (DE), and Embedded Systems. He has designed and developed complex systems in both the aerospace and automotive domains and holds multiple patents and trade secrets. He is also an active member of industry bodies such as SAE, OMG, and INCOSE.

Dinesh Kumar Kushwaha is an Associate Director - Engineering at Collins Aerospace, RTX**.** He has 20 years of experience in guidance, navigation, aircraft performance prediction, system engineering using the MBSE approach, and machine learning/neural networks. He is an INCOSE-certified system engineering professional (CSEP) and has several patents, trade secrets, and conference papers. He is also part of industry bodies such as RTCA, SAE, and INCOSE**.**



Mohammed Masood Ahmed is a Principal Engineer at Collins Aerospace, RTX, with 15 years of experience in the Aerospace domain specializing in development of Flight Management System (FMS). He has contributed from systems engineering to software development leveraging Model-Based Systems Engineering (MBSE) and Model-Based Product Line Engineering (MBPLE) for Product Development.