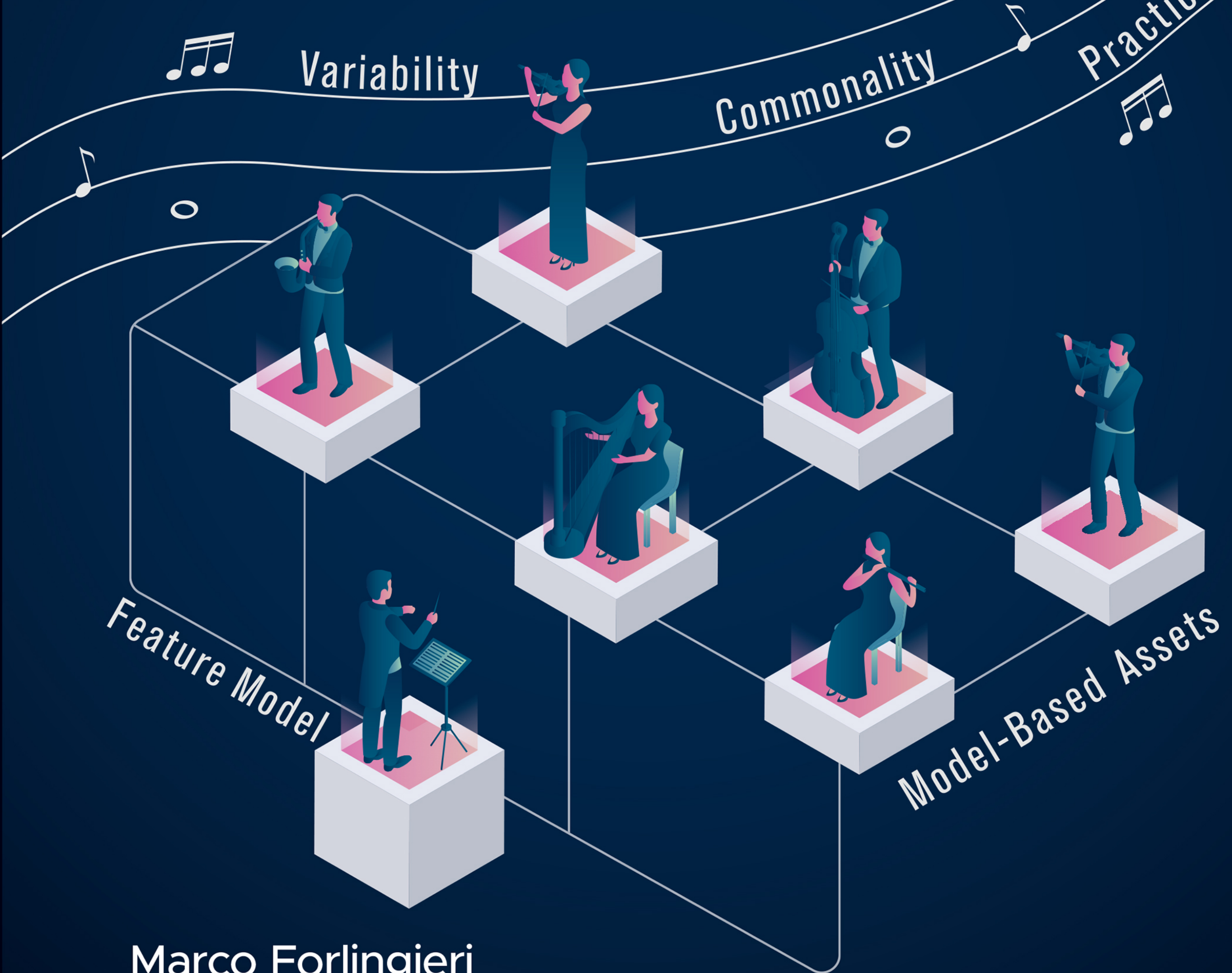
Model-Based Product Line Engineering (MBPLE)

The Feature-Based Path to Product Lines Success



Marco Forlingieri
Tim Weilkiens
Hugo Guillermo Chalé-Gongora

WILEY

Model-Based Product Line Engineering (MBPLE)

Model-Based Product Line Engineering (MBPLE)

The Feature-Based Path to Product Lines Success

Marco Forlingieri
Tim Weilkiens
Hugo Guillermo Chalé-Gongora

WILEY

Published by John Wiley & Sons, Inc., Hoboken, New Jersey.
Published simultaneously in Canada.

Library of Congress Cataloging-in-Publication Data: 2025902944

Print ISBN: 9781394204663
ePDF ISBN: 9781394204670
ePub ISBN: 9781394204687
oBook ISBN: 9781394204694

Cover Design: Wiley
Cover Image: Maria Izabel Thimoteo

Set in 9.5/12.5pt STIXTwoText by Lumina Datamatics

MBPLE book Dedications

To Prof. Dr. Danilo Beuche and Marco Ferrogalini, whose mentorship and inspiration shaped my career and introduced me to the world of Product Line Engineering.

To my grandparents, Angelo, Annamaria, and Maria, who remind me daily of the strength and wisdom of my Italian roots.

And to my wife, Maria, and my dog, Kyoto, my companions on every adventure around the world. Your support and love make every journey worthwhile.

Marco Forlingieri

To all the fantastic engineers who are committed to making the world a better place and to my family who make the world a great place for me every day.

Tim Weilkiens

To my parents and siblings, whose example, unconditional love, and support have led me to where I am and made me who I am.

To Geneviève and Anne-Nohemi, for the joy of seeing your smile every morning, and for setting me right whenever I'm wrong. Thanks for making each return home into the best moment of my day. You make life exciting.

Hugo Guillermo Chalé-Gongora

Contents

Foreword

Once upon a time, a unique trio got together: an Italian tool-vendor, a French-Mexican executive, and German subject matter expert (it sounds like the start of a joke, but this is serious stuff, folks!). We come from different backgrounds, possess diverse skills, and pursue distinct goals, but we share one undeniable passion called Model-Based Product Line Engineering (MBPLE).

Several years ago, while working at Renault, Guillermo, the executive, had an intuition: "What if we bring Model Based Systems Engineering (MBSE) and Product Line Engineering (PLE) together?" It was the origin of MBPLE, a concept Guillermo later applied at Alstom and Thales. Marco, now an executive tool vendor, read the work of Guillermo and decided to make PLE and MBSE his main expertise, applying it at Bombardier Transportation and later at Airbus. Tim, the subject matter expert, wrote VAMOS (Weilkiens 2016), a book about modeling the variability with SysML, which captured the attention of the other two, being the first and only source about the topic. The world of systems engineering is small and thanks to International Council on Systems Engineering (INCOSE) we got to know each other at the point that we, the trio, decided to put in writing all the knowledge and expertise gathered about the topic. We wanted to keep the concepts as simple as possible to reach managers, engineers, and students. This is how the idea of this book came to be.

In fact, only a few books exist about product lines, much less about their development, and nothing, so far, about MBPLE. This is because product lines are akin to mythological creatures – mysterious and elusive. Like the legendary Nessie in Loch Ness, the enigmatic UFO sightings or a politician keeping promises once elected, product lines are talked about by some who claim to have experienced them, but fail to prove their existence! And yet, paradoxically, you encounter them in your everyday life. They manifest when you purchase a car, wield a smartphone, drill the wall, dry your hands, board a plane, or even when you turn on the air conditioning.

What is the secret behind product lines? How are they developed? Can someone unveil the product line that, like Aristotle's "first motor," generates countless derived products? Such a revelation proves elusive! Just like a witness to an UFO sighting, the Loch Ness monster or a trustworthy politician, many speak of it, pretending to know the truth, but most, again, are unable to provide concrete evidence.

You might be skeptical, assuming that we, the authors of this book, are merely another group of individuals pretending to have witnessed something incredible without being able to prove it. But what about the five industrial cases on the concrete application of MBPLE from Airbus, Thales, MBDA, Raytheon and Belimo at the end of the book? Or the many references provided throughout this text?

We do not ask you to believe us, we extend an invitation, a summons to join us on this extraordinary expedition, where the mysteries of product lines and their secret recipe await your discovery. Together, let us unravel the enigma, peer behind the veil, and embark on a journey that will forever transform our perception of the way we carry out your business thanks to a deeper understanding of reuse, assets sharing and variability.

Welcome, fellow traveler, to this captivating odyssey and, paraphrasing Ulysses' words in Dante's Divine Comedy (Alighieri 1995), keep in mind that "you were not made to live as brutes, but to follow virtue and knowledge of MBPLE."

Marco Forlingieri
Tim Weilkiens
Hugo Guillermo Chalé-Gongora

References

Alighieri, D. (1995). The Divine Comedy. Originally published in 1320. New York: Knopf, distributed by Random House.

Weilkiens, T. (2016). Variant modeling with SysML. MBSE4U.

Preface

In the field of systems engineering, Product Line Engineering (PLE) has emerged as a transformative approach for managing the complexity of modern product families. First introduced in the 1990s, PLE laid the groundwork for systematically developing families of software and hardware. Over the years, it has evolved to meet the needs of increasingly complex industries, including automotive, aerospace, defense, and beyond.

In today's era of digital transformation, PLE is essential for maintaining competitiveness. The complexity of product development cycles now demands an integrated, model-based approach that enables organizations to achieve digital continuity and efficient lifecycle management. For decades, International Council on Systems Engineering has played a central role in the evolution of systems engineering, embracing emerging disciplines like PLE and adapting to model-based advancements.

Recognizing the importance of PLE in advancing systems engineering, the International Council on Systems Engineering (INCOSE) established the PLE Working Group in 2012. This group plays a critical role in standardizing PLE practices and fostering global collaboration across sectors. Aligned with INCOSE's mission to promote systems engineering, the PLE Working Group has contributed significantly to the evolution of PLE and, more recently, Model-Based Product Line Engineering (MBPLE).

The authors of this book are recognized international experts, bringing extensive experience in both the theory and practical application of MBPLE across industries. Their deep involvement in INCOSE and contributions to major organizations and international standards make them uniquely qualified to present this comprehensive guide.

The publication of ISO/IEC 26550:2015 marked a significant milestone for PLE, providing a foundational standard for product line engineering and management. This was further strengthened by ISO/IEC 26580:2021, which introduced feature-based methods for managing PLE across industries. INCOSE's PLE Working Group has played a vital role in formalizing these standards, ensuring a cohesive framework for PLE across sectors and fostering collaboration within the systems engineering community.

The combination of Model-Based Systems Engineering (MBSE) with PLE – now known as MBPLE – has become a cornerstone of innovation. By leveraging models, MBPLE enables engineers and organizations to manage product family variations and configurations with unprecedented precision and flexibility. This integrated, model-based approach has made it possible to address the complexities of today's product lifecycles, delivering improved digital continuity and competitive advantages.

While the evolution of MBPLE continues to advance, it is accompanied by diverse interpretations and emerging trends, creating a need for clarity and guidance. This book fills a critical gap by offering a comprehensive guide to MBPLE, combining strategic insights with foundational concepts and practical implementation techniques, particularly in Systems Modeling Language (SysML). Through real-world case studies and insights from PLE pioneers, it explores the transformative power of MBPLE, including the potential impact of Artificial Intelligence and its associated opportunities and risks. This book serves as an essential resource for anyone aiming to learn and leverage MBPLE in today's increasingly complex and competitive product development environments.

Ralf Hartmann, INCOSE President

About the Companion Website

This book is accompanied by a companion website.

www.wiley.com/go/forlingieri/mbple1e

This website includes:

- Author Images
- Website content and structure

1

Introduction

Model-based product line engineering (MBPLE) is a broad topic involving many stakeholders. For most companies, MBPLE's transformational nature concerns almost all engineering activities and operations. Leaders need to understand MBPLE to plan and implement it strategically, as well as to understand and frame its application. But MBPLE principles, concepts, and methods must also be understood by a wider population, including the following:

Process owners who create and adapt MBPLE methods and tools to meet organization-specific requirements.
Engineers who use MBPLE methods directly to create feature models and shared assets supersets.
Engineers and product owners who develop parts of the product line.
Students who learn systems engineering, product line engineering (PLE), and model-based approaches.

This book discusses MBPLE as a central topic, covering all its facets, including the enabling factors for a successful implementation of MBPLE in an organization. For readers who are new to the topic or who would like to acquire a broad understanding of MBPLE, reading the book following the order of the sections might be the best approach. However, readers who are interested in a particular topic or who have previous knowledge of PLE can browse freely through the different sections of the book.

Part I discusses the motivation of MBPLE, which is important for all MBPLE stakeholders. Like all engineering approaches, MBPLE is only a means to an end. Therefore, it is important and helpful to have a good understanding of the purpose of MBPLE.

Part II describes the fundamental concepts of MBPLE, independently of concrete implementations of modeling languages and tools. Like Part I, Part II is also an important reading for all MBPLE stakeholders. The chapters in Part II look at the historical development of model-based approaches and PLE, as well as relevant standards such as ISO/IEC 26550 (ISO/IEC 2015) and ISO/IEC 26580 (ISO/IEC 2021), which provide the framework wherein the MBPLE concepts are presented.

Part III explains the technical implementation of MBPLE and is of particular interest to those who use MBPLE directly. One focus is on the models with SysML, whereby both the widely used SysML v1 and the new SysML v2 are used.

Part IV explains the adoption of MBPLE in organizations, a significant topic for leadership roles. The chapters consider both the investment and return on investment of introducing MBPLE, the processes, the methods, the organization, the information model, and the tool chains. This part is

Model-Based Product Line Engineering (MBPLE): The Feature-Based Path to Product Lines Success, First Edition.
Marco Forlingieri et al.

rounded off by a chapter with an interview of two PLE pioneers, Dr. Danilo Beuche from PTC and Dr. Charles Krueger from Big Lever Software.

Part V intends to show that the book presents MBPLE not only in theory but also in real-life practice. Five chapters focus on industrial cases about adopting MBPLE from five different organizations.

In the appendix, a glossary lists the most important MBPLE terms for reference.

Acknowledgments

It takes a lot of people to write this book, many of whom are directly acknowledged in this book. Thanks to Ralf Hartmann, president of INCOSE, for writing the preface and his continued commitment to the systems engineering community. Many people have agreed to report on their MBPLE approaches within their companies and have written industrial cases in Part V. These contributors include Davi Henrique de Sousa Pinto from Airbus, Dieter Wagner from MBDA, Agnès Guiblin from Thales, James Teaff from Raytheon, and the Belimo's team, consisting of Manuel Pijorr, Alba Pennisi, Markus Hüppi, Daniel Messmer, Mariana Reyes Perez, Mitko Tanevski and Simon Hoffman. Thank you all for your work and the insights you provided in this book. The two PLE pioneers, Dr. Danilo Beuche and Dr. Charles Krueger, took the time to share their views on the world of PLE. Thank you very much for your valuable time and contribution; without your contribution, this book would have not been complete. Many ideas were discussed and matured in the INCOSE PLE Working Group (INCOSE PLE 2024) context. Marco Ferrogalini, who contributed his wealth of experience as Vice President and Head of Modeling and Simulations at Airbus, was involved in the book's initial ideas and structure.

References

INCOSE PLE (2024). INCOSE PLE Working Group. Webpage viewed on 15 July. https://www.incose.org/communities/working-groups-initiatives/product-lines.

ISO/IEC (2015). *ISO/IEC 26550:2015* Software and systems engineering – reference model for product line engineering and management. Geneva ISO/IEC.

ISO/IEC (2021). ISO/IEC 26580:2021 Software and systems engineering – methods and tools for the feature-based approach to software and systems product line engineering. Geneva ISO/IEC. https://www.iso.org/standard/43139.html.

Part I

Motivation

The first part of the book focuses almost solely on the purpose of Model-Based Product Line Engineering (MBPLE). This emphasis is intentional, based on the authors' experiences that company members often lose sight of the overall purpose of their initiatives. Frequently, they mistake tools and methods (such as digital transformation, modeling, agile practices, or digital twins, to name just a few) for end goals. Avoiding this confusion is crucial for MBPLE due to its transdisciplinary nature, the required investments, and the need to implement it across the whole company.

Chapter 2 begins by drawing a picture of the context in which today's products are developed. The chapter briefly explores complexity and variability as well as their implications on the methods needed to develop and deliver products that are adapted to the needs of different customers. This is the object of Chapter 3, which presents different reuse patterns along with their advantages and disadvantages. Special attention is given to the dangers of implementing unplanned reuse strategies and to the prerequisites that should be in place to ensure a successful implementation of a product line approach.

Chapter 4 explains the difference between document-based engineering and model-based engineering. The growth in the adoption of model-based engineering as well as the perspectives of its evolution are also explored. The chapter presents the benefits and challenges associated with model-based engineering. Chapter 5 is a logical complement to Chapter 4 as it proposes a similar approach to explain the differences between the delivery of single products and the delivery of products in a product-line-based approach. This chapter explains the advantages of transitioning to a programmatic approach to manage the products of a product line as a single entity, as opposed to managing a plethora of separate development projects.

Chapter 6 closes Part I with an analysis of digital threads and digital continuity as enablers to allow managing data during all the stages of the product line life cycle and across the different products of a product line in a consistent way. The chapter also underlines the importance of the openness of IT solutions and the need to connect these in integrated networks to support MBPLE in an effective manner.

2

Complexity and Variability

2.1 Introduction

As the discipline of systems engineering evolves, it must confront the growing complexity inherent in its expansive scope. This chapter first explores in Section 2.2 the dual drivers of complexity identified by the International Council on Systems Engineering (INCOSE) in its Vision 2035 (INCOSE 2022): technology-driven and scope-driven complexity. Technology-driven complexity is fueled by fast technological advancements, while scope-driven complexity emerges from the increasing size and interdependencies within the development scope. We explore deeper each driver, emphasizing their distinctive impacts on the field.

Section 2.3 addresses the concept of variability driven by customization in product development. Customization, while enhancing individuality and flexibility, often results in a diverse range of product variants.

The combination of complexity and variability are discussed in Section 2.4, two trends that are deeply interrelated and form the challenges in modern product and systems engineering, emerging as a key focus of the chapter. Through a few examples, it illustrates the difficulties of balancing high variability demands with the need to manage complexity effectively.

Ultimately, this chapter contextualizes these concepts within the broader framework of systems engineering, emphasizing the criticality of innovative development paradigms to address the combined challenges of increasing complexity and variability demands in the development of complex systems.

2.2 Complexity

In systems engineering, complexity has always been a key aspect to address. The field itself developed to better handle the increasing scale, interconnections, and complexity in systems development. Professor de Weck (Conservation of Complexity 2023) discussed the rising complexity in today's products and systems, noting that it is much higher than in the past, as highlighted in Figure 2.1. This complexity is intended to lead to enhanced performance and, possibly, increased resilience. It is a common understanding among systems engineers that complexity is escalating each year due to rapidly changing contexts, the growing interdependencies of systems, and more challenging projects taken on by organizations and governments. He also pointed out how monumental achievements like the lunar landing, the International Space Station, 20-hour transoceanic flights, or capturing images of the earliest proto-galaxies post-Big

Model-Based Product Line Engineering (MBPLE): The Feature-Based Path to Product Lines Success, First Edition.
Marco Forlingieri et al.
 Companion Website: www.wiley.com/go/forlingieri/mbple1e

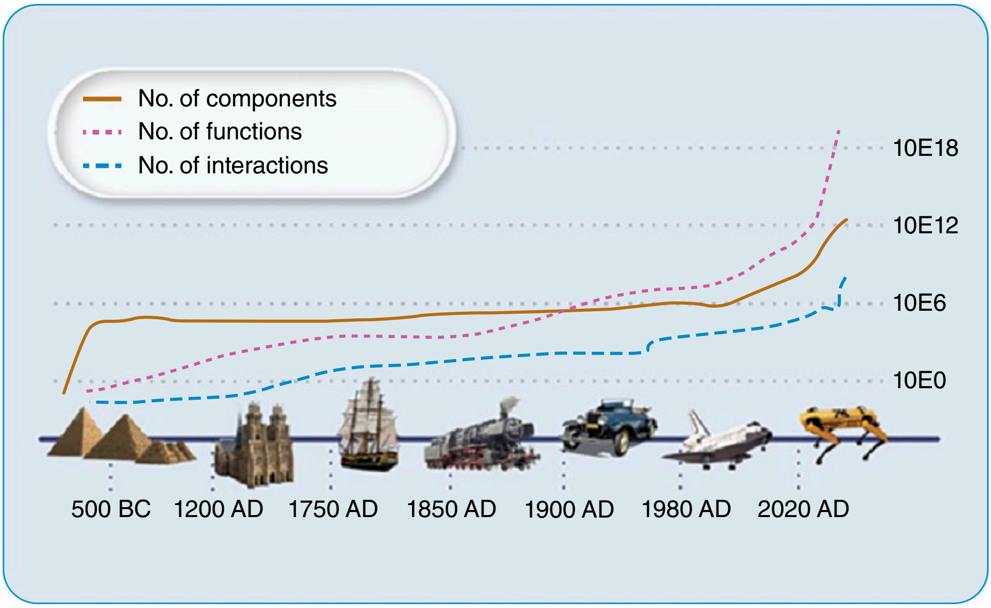


Figure 2.1 Representation of the increasing complexity in history. *Source:* Adapted from INCOSE 2022.

Bang are made possible by the coordinated effort of numerous components, including hardware, software, and human collaboration. However, he also emphasized that many aerospace programs are facing rising costs and challenges in management, partly due to a limited grasp of how system performance, complexity, and the effort needed for design, construction, and validation are interrelated. This can be also explained by what he theorizes as the “First Law of System Science for Conservation of Complexity.” In simpler terms, it suggests that the complexity of a system is directly related to expected improvement in its performance, adjusted by the amount of work and resources put into its development and construction (Conservation of Complexity 2023).

In systems engineering, the typical process that aims at managing complexity involves breaking down a “problem” into smaller, more manageable parts, designing solutions for these individual parts, and then reassembling them to form the complete system. This approach is effective for “complicated” systems with fixed interactions among parts, even if they contain many interrelated components and may exhibit unpredictable behavior (INCOSE 2016).

However, this method encounters difficulties when new technologies are integrated into traditional systems and when the scope of these systems expands to ambitious developments. In complex systems, the emergent properties crucial to the system’s functionality cannot be fully understood by examining the individual components separately. These properties only become apparent when considering the system as a whole. According to INCOSE’s Vision 2035 (INCOSE 2022), the complexity in engineering is continuously escalating, primarily due to two key factors: technology-driven complexity and scope-driven complexity. Other factors contribute to an increase of complexity in systems engineering, such as the complexity of the business context or of the organization in which the systems are developed. However, the focus here is on the complexity intrinsic to the system under development. Let’s examine both drivers in detail.

2.2.1 Technology-driven Complexity

Technology-driven complexity arises from the rapid pace of technological advancements and their integration into systems and products. This type of complexity is often characterized by the incorporation of cutting-edge technologies, which, while enhancing capabilities, also add challenges to the system's design, operation, and maintenance, as exemplified in Figure 2.2.

For instance, the implementation of software-defined vehicles in the automotive industry describes the complexity of technology-driven vehicles well. A software-defined vehicle is any vehicle that manages its operations, adds functionality, and enables new features primarily or entirely through software. Tesla cars are the most famous example. Software-defined vehicles are the next evolution of the automotive industry. Their architecture usually divides the vehicle's functions into different server zones, such as infotainment, safety systems, and vehicle control. Each zone integrates advanced technologies like sensor fusion, connectivity modules, and real-time data processing. The challenge lies in harmonizing these technologies to work together, ensuring vehicle performance and safety in diverse driving conditions. This complexity is compounded by the need to constantly update and maintain each server zone over the air to meet evolving technological standards and consumer demands (Burkacky et al. 2023).

2.2.2 Scope-driven Complexity

Scope-driven complexity emerges from the expansion in the scale and interconnectedness of systems. It reflects the transition from developing standalone products to creating systems part of more extensive, often interconnected networks, as illustrated in Figure 2.3.

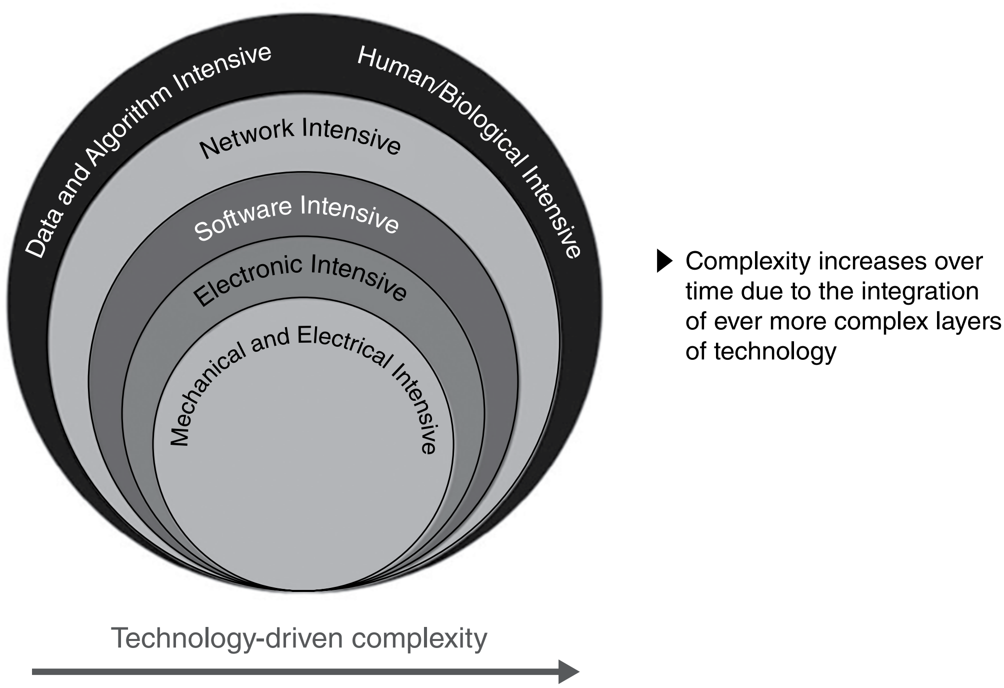


Figure 2.2 Representation of the technology-driven complexity driver. *Source:* Adapted from INCOSE 2022.

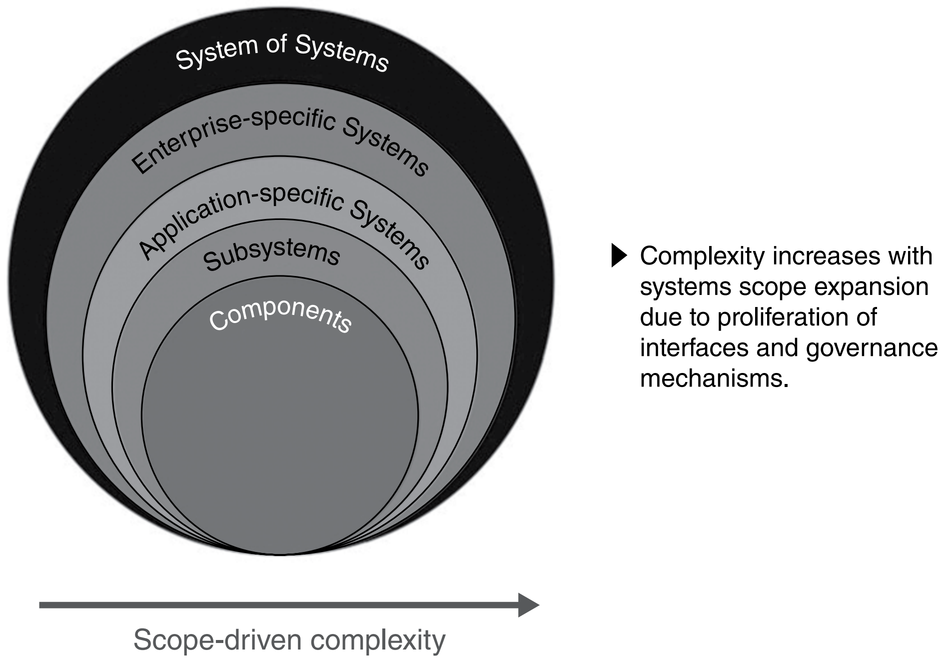


Figure 2.3 Representation of the scope-driven complexity driver. *Source:* Adapted from INCOSE 2022.

Extending the example of software-defined vehicles, scope-driven complexity becomes apparent when these vehicles connect to more extensive networks. Features like over-the-air software updates and vehicle-to-everything communication add layers of complexity. The challenge is ensuring the vehicle's different server zones work well with these external connections, maintaining reliable and secure performance in an interconnected setting. This shows how expanding the scope of vehicle systems naturally makes them more complex.

In summary, understanding complexity in systems engineering involves recognizing the multifaceted challenges posed by technological advancements and expanding project scopes. As we continue to push the boundaries of what is possible, mastery of complexity becomes a critical skill in the engineer's toolkit.

2.3 Variability

In product development, customization refers to the process of tailoring products or systems to meet the specific requirements of a customer or market segment. This often involves modifying or configuring the design, features, functionalities, or even the aesthetics of a product to align with distinct preferences, needs, or operational environments. From a portfolio perspective, customized products or systems exhibit variability, as their characteristics may differ among the members of the product portfolio. Unlike mass production, which focuses on uniformity and scale, customization emphasizes individuality and flexibility, often resulting in diverse product variants. Customization is the primary driver of variability within a product portfolio.

With increasing demand for customization, companies face the double challenge of providing market – or customer-specific variety while mastering the consequences of high variability in engineering and production. Especially in an increasingly saturated market, new products must find ways to differentiate themselves from the competition, like distinctively meeting specific customer needs to enhance customer's satisfaction (Simpson et al. 2005).

However, as observed by Meyer and Lehnerd (1997), focusing on individual customer preferences often leads to overlooking commonality and standardization across product lines. This can result in an overwhelming diversification of products and parts. While offering a comprehensive product variety has merits, it can also incur significant costs and complexity within a company.

Have you ever noticed, while traveling on different flights, even within the exact airline or across various airlines, which the airplane model might be the same, yet many aspects inside differ significantly? For instance, consider the variability in-cabin configuration and layout, the number and style of toilets, infotainment systems, or even the types of movies available during the flight. These differences result from airlines' deliberate customization and high differentiation to enhance their brand identity and customer service offerings. However, this level of variability often becomes a "nightmare" for airplane manufacturers and their suppliers. They frequently face the challenging task of reworking complex systems to accommodate these customizations, sometimes questioning the necessity of altering even minor details, like the exact millimetric position of power and water outlets within the galley of a commercial aircraft!

If not adequately managed, variability can significantly impact the design and development of systems in several ways:

Increased complexity: high variability introduces complexity into the design process, requiring engineers to consider a broader range of variables and potential configurations.

Design flexibility: More flexible design approaches are needed to accommodate changes and variations without extensive redesigns or cost overruns.

Production and supply chain adjustments: Variables impact production processes and supply chains, which need to be adaptable to produce a variety of customized products efficiently.

Increased costs: while customization can lead to higher customer satisfaction and market differentiation, variability often leads to increased production costs and complexity in inventory management.

In summary, customization in product development reflects the dynamic and varied needs of customers and market trends. However, this leads to increased variability, which, if not properly managed, can hinder project success and overall organizational performance.

2.4 The Trade-off Between Complexity and Variability

The increasing complexity in engineering and the escalating demand for customization leading to higher variability are not isolated trends; instead, they are profoundly interconnected and often feed into each other, creating additional challenges in modern systems engineering.

Have you ever considered that a European, South America, or Southeast Asia metro system might be based on the identical product family? This idea was pursued by Bombardier Transportation in 2014 with their "Entry Segment Metro" concept (Forlingieri 2014). They aimed to create a metro design that could be adapted across different regions. However, the project encountered significant challenges and ultimately did not materialize.

One of the primary reasons for this was that metro systems and light rail vehicles often symbolize a city's identity, necessitating unique and strongly differentiated designs. Additionally, the need to comply with diverse local norms and regulations significantly increased the complexity and the potential number of variants required. However, other rolling stock products, such as high-speed trains, may require heavier customization, going beyond style and aesthetics into the specific train architecture (Chalé Góngora et al. 2014).

This example showcases the trade-off between the desire for high customization and the goal of limiting complexity, which arises from the variability of the product variants. The effort to customize a metro system or a light rail vehicle across various geographical and cultural contexts shows how challenging it can be to manage both these aspects effectively.

This applies in particular to a category known as complex products and systems (CoPS). They consist of prime products that demonstrate a high degree of customization and complexity (Hobday 1998). These are high-cost, engineering-intensive items, encompassing a broad range of tailored components, specialized knowledge, and involvement from various organizations. The complexity of CoPS arises not just from their technical difficulties but also from the extensive customization required to meet the unique demands of each project (Forlingieri 2014). Naval, Aerospace, and Civil Engineering are just some of the most representative industries where this category of systems is produced.

The CoPS category is just one of the possible areas where the combination of variability and complexity closely interact.

To conclude, addressing diverse business context factors, specific customer needs, and increasing complexity, particularly within a highly regulated environment, requires new engineering practices and methods. These approaches help to balance variability and complexity in today's systems. You will find some answers later in this book!

References

Burkacky, O., Steiner, F., Deichmann, J. et al. (2023). Getting ready for next-generation E/E architecture with zonal compute. https://www.mckinsey.com/industries/semiconductors/our-insights/getting-ready-for-next-generation-ee-architecture-with-zonal-compute (accessed June 2024).

Chalé Góngora, H.G., Ferrogalini, M. and Moreau, C. (2014). How to boost PLE by using MBSE – a case study of a rolling stock product line. In: *Proceedings of the 5th International Conference on Complex Systems Design and Management (CSDM)* Paris, France: Springer.

Conservation of Complexity: The First Law of System Science (2023). MIT AeroAstro. https://aeroastro.mit.edu/news-impact/conservation-of-complexity-the-first-law-of-system-science/ (accessed January 2024).

Forlingieri, M. (2014). *Platform Planning and Design for Rail Rolling Stock: An Exploratory Study of Platform Development within Bombardier Transportation*. Enschede: University of Twente.

Hobday, M. (1998). Product complexity, innovation and industrial organisation. *Research Policy* 26 (6): 689–710. doi: 10.1016/S0048-7333(97)00044–9.

International Council on Systems Engineering (INCOSE) (2016). A complexity primer for systems engineers.

International Council on Systems Engineering (INCOSE) (2022). Systems engineering vision 2035. https://sevisionweb.incose.org (accessed July 2024).

Meyer, M.H. and Lehnerd, A.P. (1997). *The Power of Product Platforms: Building Value and Cost Leadership*. New York, NY: Free Press.

Simpson, T.W., Siddique, Z. and Jiao, J. (eds.) (2005). *Product Platform and Product Family Design: Methods and Applications*. New York: Springer. 1–15.

3

Reuse Strategies

3.1 Introduction

In today's highly competitive markets, most organizations usually try to leverage previously developed assets for their reuse into a new product. This understanding of reuse, shared by most common folk, is one of the more widespread in the engineering of systems: using again what one has done before to produce a solution that is similar or close to a previous one. The *purpose* of reuse, however, is often much less formalized or even understood: produce the same thing or something similar, only *faster* and *better*. In other words, a reuse strategy should be set to produce clearly stated outcomes, such as improving product characteristics, like quality and performance, or program objectives, like cost-effectiveness, time to delivery, or risk mitigation.

Indeed, in most industrial organizations, practically no system is created from scratch. Engineers are most likely to reuse knowledge from a previous project or product in the form of documents, procedures, diagrams, or models. The problem is that this knowledge, albeit optimal for local usage, is very often partial and sometimes even inadequate when used in a different context. In siloed organizations, this usually results in incompatibilities of engineering artifacts, inconsistent technical data, and duplication of information in different repositories, making the implementation of the principles exposed in Chapters 5 and 6 extremely challenging. In a nutshell, these practices make reuse very problematic at a cross-company level and hamper the creation of well-structured, corporate-wide product lines. Whilst Section 3.2 introduces some of the industrial reuse approaches, Section 3.3 provides a brief analysis of the benefits and risks of those reuse patterns.

3.2 Different Reuse Approaches

As an engineering practice, reuse is well-documented in software and manufacturing, where examples of successful reuse approaches are well known, such as configurable software modules, standard software architectures, flexible manufacturing systems, or automobile platform sharing (Holweg 2008). Reuse is also associated with concepts like modular architectures, system families, and standardization of parts.

The origins of product line engineering (PLE), as the main reuse practice of this book, are often traced back to McIlroy's work on "Mass Produced Software Components" (McIlroy 1968) (see Chapter 10 for a more detailed history of PLE).

Model-Based Product Line Engineering (MBPLE): The Feature-Based Path to Product Lines Success, First Edition.
Marco Forlingieri et al.

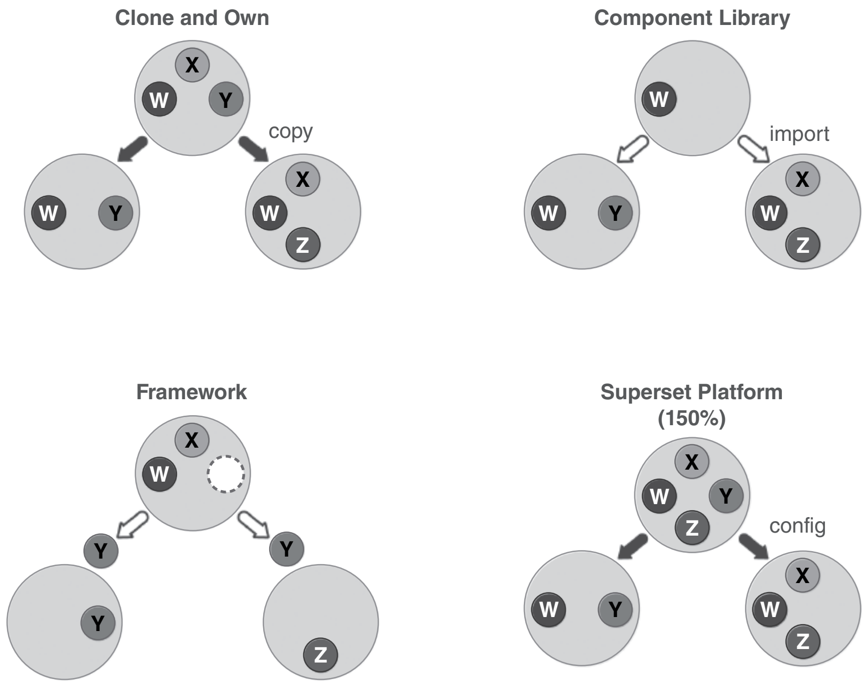


Figure 3.1 Different reuse patterns. *Source:* Adapted from Czarnecki, K.

However, the formalization of PLE and other reuse practices for their application to the engineering of large-scale systems is still scarce and relatively new (International Council on Systems Engineering [INCOSE] 2019) despite historical, *de facto*, or casual reuse in industry, like the reutilization of existing specification documents, technical drawings, or test procedures, and the existence of some reports on successful industrial applications (Chalé Góngora et al. 2014; Flores et al. 2013; Gregg et al. 2016). Figure 3.1 represents a spectrum of different reuse patterns explained in the sections below.

3.2.1 Clone and Own

Usually associated with the search for short-term benefits, clone-and-own (or branch-and-merge) approaches are the easiest to implement since engineering teams are usually independent and able to manage engineering assets as they like. There is practically no sharing among the different branches, and separate efforts to maintain each individual branch are required. This approach might be a suitable scheme for businesses in which single product instances are delivered over very large periods of time. It might also be appropriate for companies that want to reuse existing products to enter a new market or propose a groundbreaking application in an opportunistic way. The success of this approach relies on mastering the technical debt and on top-class configuration management processes.

3.2.2 Component Library and Framework

Component library and framework approaches produce more substantial benefits than the clone-and-own approach. The component library approach is based on the definition of *standard components*. Besides the need to master the interactions and interdependencies amongst the components, the biggest challenge of the component library approach is to ensure that the emerging properties of the composed product are those expected by the customers. In the framework approach, efforts would instead be dedicated to the definition of *standard interfaces* in such a way that any component that complies with the standard can potentially be used to "compose" a particular product. The Autosar (Automotive Open System Architecture) framework (Autosar 2024) is a well-known example of a standardized software framework supporting a composable approach for electric and electronic system architectures in the automotive industry.

In practice, these two approaches are carried out simultaneously to foster compsability and ease of integration. They are adapted to products evolving in markets that require the delivery of a rather stable set of functionalities over time. These functionalities can be allocated to dedicated components that would evolve at a slow pace, while isolating those components that support fast-evolving functions or technologies. The main difficulties of the component library and framework approaches are that the evolution of standard components must be thoroughly planned and controlled and that managing the definition of standard interfaces can be a complicated task in large-scale systems.

3.2.3 Superset Platform

The superset platform approach is closely related to advanced PLE approaches proposed in this book, as explained below. In this approach, a "platform" of asset supersets is designed and made available for its use in customer projects. This is done through the configuration of the supersets to match the needs of a given customer. The term 150% is used to illustrate the fact that the supersets contain more than what it would take to define a single customer solution, which would sum up to 100%. The benefits of this approach include substantial reuse and automation of the production of engineering assets, which requires capabilities to control variability efficiently, master the dependencies amongst the supersets, and control the configuration mechanism of specific products to keep this from becoming a complex task.

3.3 Quick Benefits Versus Risk Analysis of Reuse Patterns

To decide which reuse pattern should be implemented in a given organization, a business case must be established, taking into consideration the business context and strategy of the organization. The business case should consider the cost-effective investment of engineering resources and time evidenced by the elimination of low-value, trivial, repetitive, demotivating, and energy-consuming tasks. This liberatory effect should translate into a more profitable investment of energy and time in high-value activities and innovation. Examples of possible activities and outcomes include, but are not limited to, improving products, reducing organizational complexity, reducing time to market and lead time, reducing engineering costs and efforts, exploring possibilities, and advancing business objectives. Chapter 21 presents the elements that can be used to build a compelling business case for MBPLE.

When the products within a portfolio are engineered as individual, unique solutions, clown-and-own techniques usually result in duplication and divergent engineering efforts for every "branch" that is created. Since controlling the consistency of the different branches usually relies on tacit knowledge and interpersonal communication, it is inevitably an error-prone activity. This difficulty (usually known as "branching hell") increases in organizations in which each discipline adopts ad-hoc techniques for managing the variation among the different products of the portfolio, introducing further inconsistencies and misunderstandings. This way of working systematically translates into process-induced complexity, which adds on top of the intrinsic complexity of the products and of the business contexts in which organizations evolve.

What sets mature PLE practices apart from the approach mentioned above is their holistic nature and the systems engineering groundings that support them. PLE can be defined as the engineering of a product line (i.e. a family of similar products with variations in features and functions) as a single entity, using a shared set of engineering assets and an efficient means of production, taking advantage of the commonality shared across the family, while efficiently and systematically managing the variation among the products (more details are given in Chapter 5). This implies that a programmatic approach must be put in place to manage the portfolio of customer projects (i.e. those that benefit from the shared assets supersets of the product line) as a single entity too, instead of managing a multitude of individual projects separately. Such an approach enables an organization to perform informed trade-offs and make balanced cost-benefit decisions regarding investments, with a view on short-term expectations and long-term goals, particularly business profitability.

The consolidation of shared asset supersets within disciplines eliminates duplication, inconsistencies, and divergence across the products of the family, on the one hand. On the other hand, consistent transdisciplinary management of variation across the different supersets and across the product line life cycle, made possible by its formalization in a variability model, helps avoid miscommunication and misalignment amongst disciplines (Chalé Góngora and Greugny 2017). Although this delivery approach might be appealing and appear as sheer common sense, it represents a true breakthrough and, for many organizations, a major shift that requires changes in governance and organizational structures, as well as strong commitment from all functions, disciplines, and leadership. These aspects will be further developed in Part IV of this book: Adoption of MBPLE.

To conclude, PLE relies on processes that help manage the life cycle of the portfolio of products of an organization (as well as the underlying reusable assets used to perform the architecting, design, and evolution of the portfolio) in order to maximize the benefits of reuse. Reusing assets, however, should be the result of a well-documented decision process, meaning that implementing PLE should require upfront investment and forethought. And yet it is safe to postulate that, in most organizations, one could find examples where reusing an asset actually proved to be less profitable over the system life cycle than developing a new one: while the initial engineering cost of the asset might have been low, the overall costs induced by debugging, repair, validation, warranty expenses, or penalties slowly but surely end up consuming the originally planned profits (Wymore and Bahill 2000; Chalé Góngora et al. 2014). To put it simply, copy-pasting assets from one project to another without an overarching strategy cannot be considered PLE (or engineering, for that matter), and it cannot yield by itself all the potential benefits of PLE.

References

Autosar (2024). Automotive open system architecture. www.autosar.org (accessed 4 July 2024).

Chalé Góngora, H.G., Ferrogalini, M. and Moreau, C. (2014). How to boost PLE by using MBSE – a case study of a rolling stock product line. In*: Proceedings of the 5th International Conference on Complex Systems Design and Management (CSDM*), Paris, France: Springer.

Chalé Góngora, H.G. and Greugny, F. (2017). Where the big bucks (will) come from – implementing product line engineering for railway rolling stock. In: *27th Annual INCOSE International Symposium*, *Adelaide, Australia*. San Diego: INCOSE.