

A scalable framework for demand-oriented model-based systems engineering application: the MBSE Cube

Umut Volkan Kizgin   and Thomas Vietor 

Technische Universität Braunschweig, Germany

 u.kizgin@tu-braunschweig.de

ABSTRACT: The complexity of modern products poses significant challenges for the industry. Existing model-based systems engineering (MBSE) methodologies often lack the scalability and mechanisms for assessing maturity required to meet diverse organizational needs. Implementing MBSE all at once is impractical due to the complexity of changes required and resistance to change among employees. The MBSE Cube was developed as a scalable, demand-oriented framework to support organizations transitioning from no systems engineering processes or document-based approaches to model-based practices. This artifact-based approach guides the systematic creation of development artifacts, forming the foundation for MBSE implementation. By integrating abstraction levels, system views, and maturity levels, the Cube helps organizations assess their state and develop tailored MBSE adoption strategies.

KEYWORDS: model-based systems engineering (MBSE), systems engineering (SE), design methodology, organisation of product development, product modelling / models

1. Introduction

The increasing complexity of modern products presents significant technical and organizational challenges across industries such as automotive, mechanical and plant engineering, and the railway sector. The growing role of software in product development (PD) further intensifies these challenges (Wilms et al., 2019). Traditional development approaches are often insufficient to manage this complexity. Systems Engineering (SE), a methodology long established in aerospace and defense, has shown great potential in addressing these issues (Friedenthal et al, 2015). Model-Based Systems Engineering (MBSE), an evolution of SE, leverages digital models to represent system requirements, functions, and structures, offering a “single source of truth” for all stakeholders (Estefan, 2008). This model-based approach enables efficient management of complex systems across their lifecycle, reduces risks, and shortens development times (INCOSE, 2007; Weilkiens, 2007). A literature-based comprehensive analysis of MBSE benefits can be found in Henderson and Salado (2021).

While MBSE offers significant advantages, its implementation faces numerous challenges. Key issues include establishing a comprehensive model as a single source of truth, governance and management of the MBSE environment, and organizational culture change (Vaneman & Carlson, 2019). Successful implementation requires an enterprise-wide approach, addressing new skill requirements, model management, and integration with existing tools and processes (Papke et al., 2020). The challenges of transitioning to MBSE align with those observed in enterprise transformation projects, such as ERP (Enterprise Resource Planning) implementations, where business process customization and alignment are critical. To address these challenges, organizations should develop a well-planned implementation strategy, guided by an enterprise architecture that encompasses strategic vision, capabilities, and operational concepts (Papke et al., 2020; Chami & Bruel, 2018). However, these challenges can differ significantly depending on the type and size of the organization. Small and medium-sized enterprises

(SMEs) often struggle with limited resources and expertise, which hinders their ability to implement structured approaches effectively. (Baschin et al., 2023) A further complication we have identified is the absence of established SE processes, which results in a lack of comprehensive development artifacts, such as documented requirements or system architectures. These artifacts are essential for a smooth and effective transition to MBSE, as they form the foundation for model-based practices and ensure consistency and traceability throughout the development lifecycle. Without these elements, the implementation of MBSE becomes significantly more challenging. On the other hand, large organizations often have well-established, document-based SE processes deeply embedded in their workflows. Transitioning to modern model-based development is complex and often met with resistance to change, requiring careful planning. Both SMEs and large organizations face unique challenges in adopting MBSE, requiring primarily demand-oriented strategies to overcome this resistance by demonstrating its practical benefits, while addressing their specific needs and constraints.

2. State of the art - MBSE

Systems engineering is an interdisciplinary and systematic approach for managing complex systems throughout their lifecycle. It is widely used in industries such as defense and aerospace, addressing both technical and non-technical systems, including organizations and processes (Gausemeier et al, 2013; INCOSE, 2007). Its processes are standardized in ISO 15288, which defines a structured framework for the lifecycle of a system that includes technical processes such as requirement definition, system design, integration, verification, and validation, as well as project, agreement, and enterprise processes (ISO, 2023). Model-Based Systems Engineering is an emerging paradigm that uses formalized system models to support and streamline Systems Engineering tasks throughout a system's lifecycle. It differs from traditional document-based approaches by offering a more integrated and standardized method for capturing system design information as models. It employs modeling languages like SysML to express system information concisely and coherently. Numerous commercial tools are available to support MBSE applications, such as Enterprise Architect and Cameo Systems Modeler. These tools facilitate collaborative modeling, version control, and integration with other engineering tools, making them indispensable for modern SE (Ramos et al., 2012; Eigner et al., 2017a; Friedenthal et al., 2015). In addition to the modeling language and tool, selecting an appropriate modeling approach and ensuring its consistency with other modeling components is critical for the successful application of MBSE (Kaiser, 2013). Particularly in the development of complex technical systems, the chosen approach shapes the structure of the system model, which serves as the foundation for a paradigm shift from document-based to model-based development. Therefore, a modeling approach must be selected that organizes the system's development artifacts according to its complexity and creates a modeling framework for the system (Alt, 2017; Delligatti, 2013).

The literature offers numerous MBSE approaches that provide methodologies for a comprehensive and interdisciplinary system description. However, the terminology is not consistent, and MBSE approaches are often referred to by terms such as “method,” “methodology,” or “process.” A detailed overview of MBSE approaches was presented by Estefan (2008), where “methodology” is defined as a collection of processes, methods, and tools. In this context, MBSE approaches such as OOSEM, SYSMOD, IBM RUP SE, Harmony SE, OPM, Vitech, and JPL SA are frequently analyzed in the literature. Additionally, there are other approaches referred to as methodologies in Estefan's definition, such as Arcadia, METUS, and CONSENS as some of them provide their own processes and modeling tools.

It is important to note that many approaches—particularly those labeled as methodologies in the literature—do not offer a modeling framework but merely describe process steps, which are often too abstract and inadequate for solving real-world system modeling problems (Morkevičius et al., 2017). To address this, approaches such as RFLP, MecPro, CESAM, and SPES MF have been developed by various institutions to be language- and tool-independent, providing a framework for organizing modeling efforts. (Eigner et al, 2017a; Krob, 2022; Pohl et al., 2012; Baughey, 2011; Kleiner & Krame. 2013) The names of these approaches also vary, and they are referred to by terms such as “architecture method” or “model framework.” Additionally, many architecture frameworks for system description are analyzed in the literature, such as the Automotive Architecture Framework (AAF) and Architecture Design Framework (ADF) (Broy et al., 2009; Chale et al., 2013). There are also modeling frameworks like the MBSE-Grid,

which are based on older enterprise architecture frameworks such as the Zachman, DoDAF, MoDAF, TOGAF and FEAF.(Morkevičius et al., 2017; Reichwein & Paredis, 2011).

Existing MBSE methods often focus on new system development with an emphasis on modeling, while neglecting the systematic creation of development artifacts essential for effective product development (PD). This gap makes the transition to MBSE particularly challenging for SMEs that lack prior SE activities or operate with limited resources, as well as for large companies with established but complex processes. These challenges underline the necessity for approaches tailored to the specific requirements and constraints of organizations of all sizes. Scalable and demand-oriented methods are essential to facilitate the shift from document-based processes to integrated and efficient practices. These methods must align with varying SE maturity levels and integrate into existing workflows while minimizing employee resistance to change. Therefore, companies need demand-oriented strategies that enable a gradual and structured introduction of MBSE practices.

3. Contribution of the paper - MBSE in practice

This paper presents the results of the RePASE project, funded by the German Federal Ministry of Education and Research (BMBF), which aimed to establish SE and support the transition to MBSE in response to increasing system complexity. It sought to transform PD by integrating suitable innovative processes, methods, and tools. Within this context, the identified challenges and gaps in MBSE implementation led to the formulation of two key research questions:

- How can structured methods support a phased MBSE transition while ensuring stakeholder acceptance?
- What strategies can facilitate MBSE adoption in organizations with different maturity levels?

To address these questions, SE activities were simplified into five core areas for better acceptance and comprehension and aligned with ISO 15288 to partially cover technical processes and ensure compatibility with the V-Model. The framework is based on requirements management, architectural design, implementation, integration & test, and project management, enabling a structured transition from document-based to model-based processes.

4. The MBSE Cube as an implementation framework

The MBSE Cube is introduced as a solution to address the challenges in PD and the limitations of existing MBSE methods. Traditional approaches are often designed for ‘greenfield’ developments and focus narrowly on specific processes or modeling steps. In contrast, the MBSE Cube provides a holistic and flexible framework, while integrating PD methods with maturity-level analysis to enable organizations to transition to MBSE regardless of their starting point. For organizations with established SE processes, the MBSE Cube bridges the gap by incorporating key SE domains and system views while assessing the maturity of modeling processes. This comprehensive approach offers targeted methodological support and specific recommendations for each step of MBSE implementation. It defines PD methods to systematically guide the creation of artifacts, ensuring that even companies with no prior SE activities or incomplete documentation can effectively develop the necessary system elements and transition to MBSE.

4.1. Methodological basis

The MBSE Cube is a comprehensive framework that synthesizes various established methodologies, frameworks, and standards in SE and MBSE to provide a structured and adaptable approach for managing the complexities of modern PD. Its foundation lies in combining concepts from existing approaches and adapting them to address the challenges of implementing MBSE across different organizational contexts. The essential aspects that form the foundation of the MBSE Cube include:

- Grid Structure: Inspired by the MBSE Grid by Morkevičius et al. (2017) and enriched with architecture frameworks such as AAF, ADF, and CESAM, offering a modular and systematic approach to system development. (Broy et al., 2009; Chale et al., 2013; Krob, 2022)
- RFLP Principle: Employing the Requirements, Functional, Logical, and Physical layers for structured and comprehensive system representation, highlighting the distinction between the

solution-neutral logical perspective and the solution-specific implementation level. (Baughey, 2011; Kleiner & Kramer, 2013)

- System Granularity: Based on concepts from the SPES Modeling Framework to address varying levels of detail and abstraction. (Pohl et al., 2012)
- Three-Dimensional Space Structure: Inspired by product concretization model according to Pfenning (2017), enabling a multidimensional view of system development across the axes of concretization, decomposition, and variation. (Eigner et al., 2017b)
- Process Models: Incorporating methodologies like VDI 2206 to ensure structured workflows.
- Maturity Models: Leveraging approaches such as SE4OWL (Wilke et al., 2022) and others to evaluate and enhance the development maturity of systems.
- Standards: Consideration of ISO 15288 to cover the technical SE processes. (ISO, 2023)

4.2. Structure of the MBSE Cube

A generic MBSE implementation methodology must support scalable, demand-oriented adoption, align with organizational goals, integrate with legacy systems, and establish a basis for a “Single Source of Truth” model. It should provide clear SE activities, grounded in models like the V-Model and ISO 15288, ensure traceability, and foster employee acceptance through a defined transition path. To assess progress and identify improvements, a robust maturity assessment is essential. Addressing these needs, the MBSE Cube offers a structured, scalable framework for evaluating maturity levels and enabling demand-oriented MBSE application. While the MBSE Cube is currently referred to as a “framework,” the integration of all specific PD methods for each step will contribute to its potential evolution into a methodology. As Estefan (2008) notes, the distinction between a framework and a methodology often lies in the breadth of application and the integration of specific practices. However, the MBSE Cube should not be confused with an architectural framework for system design; rather, it is a framework specifically tailored for implementing SE methods and MBSE practices. It introduces the steps for an artifact-oriented development approach and provides a structure for creating suitable artifacts to support the effective application of MBSE. It could also be described as a “MBSE implementation framework.” The architecture for the system to be developed must be created and analyzed in the steps of functional and logical architecture.

4.2.1. Dimensions

The MBSE Cube is divided into smaller cubes, each representing a specific perspective on a system, based on frameworks such as the Magic Grid (The context of each cell is explained in detail in Section 4.4.2.) Its structure is organized around three main dimensions: granularity levels, system views, and maturity grades, providing a comprehensive framework for applying MBSE. Figure 1 illustrates the structure of the MBSE Cube, its dimensions, and the scope of the cube within the V-model.

The **abstraction levels** within the MBSE Cube offer a hierarchical view of the system, structured into three distinct layers. The **System of Systems (SoS)** level focuses on analyzing interactions and dependencies between system of interest (SoI) and neighboring systems. It can also be viewed as a conceptual layer where operational development artifacts are created. The **System and Subsystems** level focuses on SoI and its internal structure, providing a solution-neutral view of its subsystems and their interactions. At this level, the system is analyzed in terms of its functional and logical architecture, capturing both its behavior and structural organization. At the most granular level, the **Component Level** defines and analyzes individual components of a system, which represent a solution for the defined logical systems at the intermediate level. Therefore, the actual system granularity must be determined at the intermediate level. This means that the SoI must be decomposed into its logical, solution-neutral subsystems until the final level of granularity for the transition to the domain-specific level is reached. Consequently, the architecture and the definition of the system’s final level of granularity for decision-making must be clearly specified. The “Component Level,” particularly the Behavior and Structure pillars, corresponds to the “P” in the RFLP approach (Figure 2). The term “solution” has been excluded as a designation for this level because it also encompasses the requirements space. This hierarchy allows the Cube to scale from high-level strategic views to detailed component-level analysis, enabling organizations to address varying levels of complexity and create the corresponding artifacts.

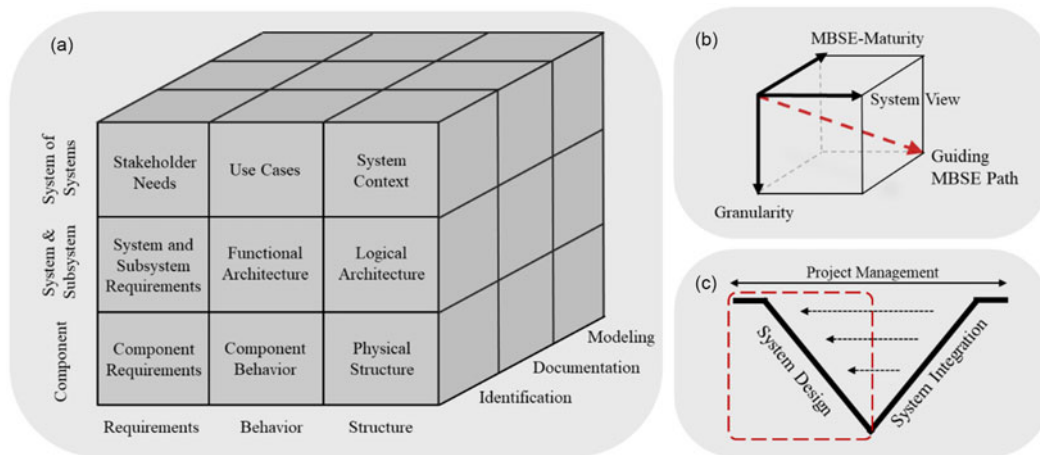


Figure 1. a) Structure of MBSE Cube b) Dimensions c) Scope in the V-model

The second dimension of the MBSE Cube covers **system views**, which represent the core aspects of a system that need to be modeled and understood. These views form the primary perspectives in MBSE. The **requirements view** focuses on capturing and managing stakeholder needs and system requirements, ensuring they are properly documented and linked to other system elements. The **behavior view** describes the functional behaviors of the system, illustrating how it operates and interacts within its environment, spanning from the operational level with use cases to implementation representations such as state machines. The **structure view** defines the physical and structural elements of the system, including their relationships and configurations. Unlike traditional MBSE frameworks, the parametric pillar has been excluded, as it can be considered part of the structure pillar. Since this is an implementation framework, these three pillars are sufficient.

A unique feature of the MBSE Cube is its third dimension, which addresses **maturity levels**. This dimension provides a structured pathway for organizations to systematically enhance their MBSE capabilities. It is divided into three progressive stages: identified, documented, and modeled, each representing a distinct level of maturity in applying MBSE principles. At the **identified** stage, the focus is on recognizing and defining key system elements such as requirements, behaviors (e.g. functions), and structures (e.g. logical subsystems). This stage ensures that all relevant elements are acknowledged and clearly outlined. However, the information at this level remains unstructured and lacks the systematic organization required for more advanced MBSE applications. It serves as the foundation for initiating SE practices by ensuring that the organization has a comprehensive understanding of the necessary components. The next stage, **documentation**, builds on the identified elements by applying SE methods to formalize, classify, structure, and document the information. This stage emphasizes aligning the data with established SE practices, ensuring consistency, traceability, and completeness. By organizing the elements within the framework of SE methods, this phase establishes a solid foundation for developing dynamic and robust system models. Method-based preparation acts as the critical bridge between the initial recognition of elements and their transformation into actionable, model-based representations. The final stage, **modeled**, represents the most advanced level of MBSE maturity. At this stage, the prepared and documented elements are converted into comprehensive SysML models using MBSE tools and modeling methods. Additionally, Figure 2 illustrates the positioning of the two additional key SE fields, Project Management and Integration & Test, which are not explicitly included in the Cube. Project Management runs parallel to the MBSE maturity levels, ensuring holistic consistency and oversight in the documentation and modeling processes. Meanwhile, the Integration & Test domain aligns with the X- and Y-axes to encompass elements from each system view and level, illustrating system validation at each level (right side of the V-Model) and the need for behavior and structure views to facilitate verification and validation.

4.2.2. Cells

The MBSE Cube's dimensions—granularity levels, system views, and maturity grades—interact to form a three-dimensional grid of smaller cubes, based on the Magic Grid framework with several distinctions. In this section, we describe the purpose of these cells, the SE activities they include, and the methods used to create the necessary artifacts.

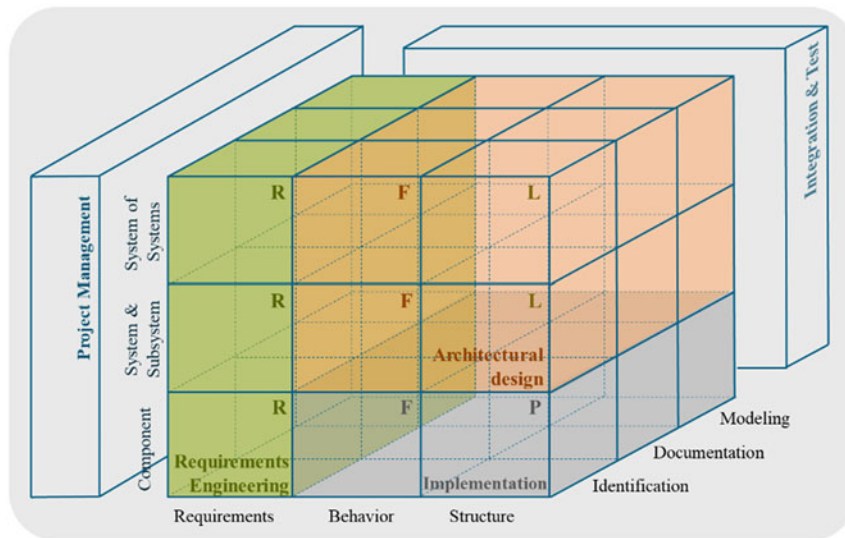


Figure 2. Visualization of the comparison between RFLP and SE processes with the MBSE Cube

Stakeholder Needs (SN): Represents information gathered from various stakeholders of the system, including user requirements, regulations, organizational policies, procedures, among others. The key activities involve identifying stakeholders, eliciting their needs, categorizing and prioritizing requirements, and establishing traceability to use cases and system requirements. For creating the artifacts, methods like *Stakeholder Analysis*, *Interviews*, *workshops*, *surveys*, and *personas* can be used.

Use Cases (UC): Defines scenarios that illustrate the interactions between the system and its users. It captures functional goals and workflows, providing a user-centric perspective on how the system fulfills its intended purpose. Activities include identifying system users, specifying use cases, and developing detailed scenarios and workflows. Methods such as *Use Case Analysis*, *Scenario Analysis* and *User Journey Mapping* support the creation of artifacts.

System Context (SC): Defines the boundaries, interfaces, and interactions of the system within its broader environment at the SoS level, capturing dependencies with external systems. Activities include defining system boundaries, identifying external interfaces, and specifying relationships with external entities. *System Context Analysis* and *Rich Picture* are useful methods for creating these artifacts.

System Requirements (SR): Focuses on defining and formalizing the functional and non-functional requirements of the system and its subsystems, derived from stakeholder needs, while considering other artifacts in an iterative process. This involves analyzing relationships between requirements and the system context, structuring and prioritizing requirements, and validating them with stakeholders. In addition to the techniques mentioned in SN, techniques such as *Literature and Patent Analysis* or *Requirement Breakdown Structures* can also be applied.

Functional Architecture (FA): Remains solution-neutral while describing the functions required to fulfill the system requirements. It defines the functional hierarchy and the interplay of functions, ensuring a clear structure for meeting system objectives. Key activities include identifying and decomposing system functions, structuring them hierarchically, and analyzing dependencies. *Functional Decomposition* and *Functional Allocation Matrix* are methods that can assist in artifact creation.

Logical Architecture (LA): Translates the functional architecture into logical components and their interactions. While the Functional Architecture focuses on *what* the system must do to meet requirements, the Logical Architecture describes *how* the system logically organizes these functions into components, remaining solution-neutral and independent of physical implementation. Activities include defining logical components, mapping them to system functions, and ensuring consistency with the functional architecture. Logical decomposition or modularizations methods such as *Design Structure Matrix* are helpful for this purpose.

Component Requirements (CR): Focuses on deriving detailed requirements and constraints for individual system components from the logical architecture. It defines the objectives each component must fulfill, ensuring consistency with overall system requirements. In addition to the methods applied at higher requirement levels, methods such as *Constraint Analysis* can be specifically utilized at this stage to address component-specific needs and dependencies.

Component Behavior (CB): Operates at the solution level, representing the behavior of real components or, depending on where the line between problem and solution is drawn, even subsystems that are composed of multiple components. While named “Component Behavior” for clarity, it provides flexibility in granularity to align with the development process. Activities involve defining component behavior, documenting state transitions, and validating behavior against system requirements.

Physical Structure (PS): Reflects the actual structure of the system, representing solution decisions by detailing the physical components and their relationships. It provides a tangible realization of the logical and functional architectures, solidifying design choices made at the solution level. Activities include defining physical components, specifying physical interfaces, and aligning the structure with logical and functional architectures. To define the real interfaces, methods such as *Physical Interface Specification* can be applied, while *Utility Analysis* or *Trade Study Analysis* can be used to evaluate and document optimal solution options.

4.3. Steps of the implementation

By analyzing an organization’s current practices and existing development artifacts, an implementation path is charted through these smaller cubes, outlining the optimal trajectory for MBSE adoption. This path ideally begins with foundational activities for greenfield projects, such as identifying stakeholders and their needs, before progressing to more advanced tasks like identifying system functions or defining the interfaces of logical subsystems. However, as previously highlighted, since MBSE adoption is often implemented in legacy systems, the transition must be tailored to the organization’s existing systems and goals to demonstrate feasibility. The pilot project with the MBSE Cube represents a phase within the overall RePASE initiative (Figure 3). Following the demand assessment in a specific project, the MBSE Cube was iteratively developed within the pilot projects. Aspects such as acceptance, resources, costs, and organizational factors were analyzed by an institute for work organization and psychology. Meanwhile, as an Engineering Design Institute, our focus was on developing a guiding tool to facilitate MBSE orientation in technical processes (product development & system modeling) rather than a comprehensive organizational implementation method.

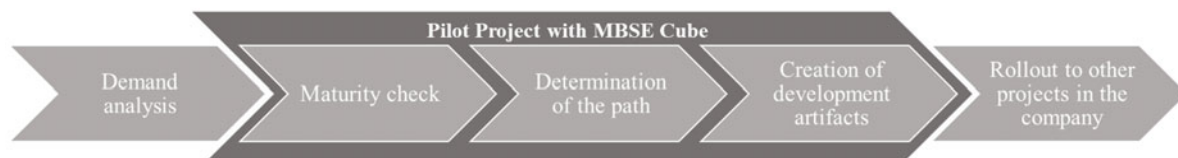


Figure 3. Implementation process of the MBSE cube in the pilot project

Based on the identified demands, the process begins with an initial assessment, evaluating the current state of MBSE maturity across all dimensions. This includes analyzing the abstraction levels at which systems are managed, the extent and quality of documentation for system views, and the maturity levels of existing processes and tools. This comprehensive assessment helps identify strengths and weaknesses, setting the stage for targeted implementation. Based on the identified gaps, priority areas are determined. For instance, to achieve the goal in terms of demand, if requirements documentation is already well-established but behavioral information about the system is lacking, the focus shifts to first identifying and then methodically and SE-compliantly documenting the artifacts of the behavior view as a foundation for behavior modeling. This ensures that efforts are directed toward the areas with the most significant potential for improvement, creating a solid foundation for subsequent steps. Each cell of the Cube includes specific SE activities and methods for creating the artifact needed for modeling, facilitating smooth transitions between adjacent perspectives.

The Cube’s structure also supports visualizing the implementation path, offering clarity and alignment across teams. For instance, on the X-axis (system views), the path might begin with identifying the functional architecture, then proceed to the logical architecture by first identifying logical components. From there, the path could progress along the Z-axis to document the logical architecture by structuring and organizing it. Finally, the process may culminate in modeling the logical architecture in SysML. This visualization helps stakeholders understand the organization’s current position, the next steps, and the long-term goals, fostering transparency, alignment, and clarity throughout the process. Figure 3 shows the steps of the approach to create development artifact to create a basis for the modeling.

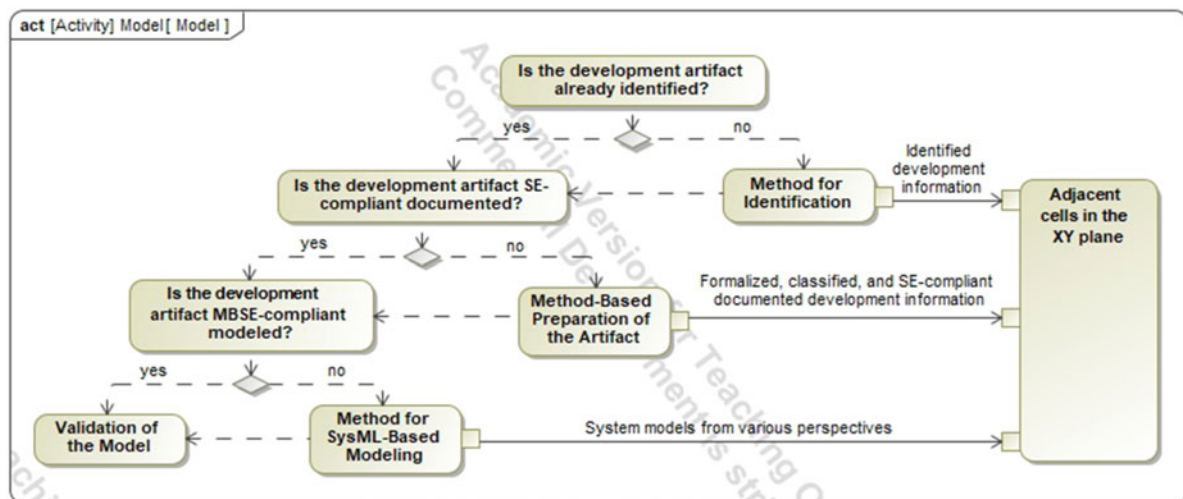


Figure 4. Process for creating development artifacts in the MBSE cube

5. Case study

The MBSE Cube was implemented in two organizations with differing SE maturity levels: a large enterprise in the railway sector and a SME producing woodworking machinery. Case selection was based on a demand analysis conducted during the pilot project initiation with stakeholders. These cases demonstrate the Cube's adaptability and its ability to support companies at various stages of SE implementation. The pilot phase lasted one year, led by a designated change agent (project manager), who coordinated the modeling transition in collaboration with university partners and stakeholders to ensure a structured application. A guideline was developed to document the methods used at each step, serving as a reference for systematically mapping tools to specific cells/views within the Cube, ensuring a structured and repeatable transition process.

5.1. Case 1: Large enterprise in the railway sector

In the railway company, the development project for an Automatic Train Operation (ATO) control unit was chosen, as it combined a well-established SE environment—including documented system artifacts such as stakeholder specifications, system requirements specifications, and test case specifications—with key MBSE transition challenges. The primary demand was to enable automated requirements validation through test case generation. Additionally, the company aimed to enhance requirements traceability through systematic MBSE implementation, with a focus on requirement modeling. To achieve this, an analysis of development artifacts (maturity assessment of development information) was conducted. Using the MBSE Cube, it was determined that there was no clear distinction between the solution-neutral and solution-specific levels. The development information in the documents was disorganized, making navigation and understanding difficult. There was insufficient traceability for the requirements, with weak connections between them. Elements in the requirements text, such as keywords, parameters, and signals, were not clearly defined, making the interpretation of the requirements more challenging. Additionally, the documentation lacked a functional perspective, necessitating behaviour modeling based on the requirements specification. To address these gaps, requirements in natural language at the component level were formalized and structured in requirement diagrams. Based on these, behavior SysML diagrams, including activity and state machine diagrams, were created to define and model the system's behavior. These models were crucial for accurately representing the system's behavior and providing a foundation for generating automated test cases directly from the behavioral diagrams. This approach effectively bridged the gap between traditional SE practices and model-based methods, enhancing the company's ability to validate and trace requirements. Due to space constraints, the paths within the Cube are represented using the abbreviations: CR-D → CR-M → CB-M → PS-M → Test Case Generation. (I: Identified, D: Documented, M: Modeled, the abbreviations for the cells can be found in Section 4.2.2.)

5.2. Case 2: SME in the woodworking machinery sector

The primary demand in the SME case was to enable variant modeling and conduct automated, model-based trade study analyses to identify the best solutions for the overall design of the cross conveyor, demonstrating the potential of MBSE in improving product development efficiency. A maturity analysis revealed a lack of basic SE practices and documentation. Critical system artifacts, such as the product's logical architecture and its components, had not been identified or documented. To address this, methods for identifying system functions were initially applied, leading to the establishment of a general functional structure. Building on this foundation, logical components and potential solutions were subsequently defined. Product development methods, such as the Morphological Box and Utility Analysis (with defined evaluation criteria and weighting), were used to systematically document and evaluate optimal solutions. These solutions were then implemented in Cameo System Modeler, enabling an automated trade study analysis, which was successfully conducted. The path is outlined as follows: SR-I → FA-I → LA-I → LA-D → LA-M → Trade Study Analysis.

These two use cases demonstrate the versatility of the MBSE Cube in addressing diverse organizational needs. The pilot projects, utilizing the MBSE Cube, were designed to help the company visualize opportunities within these specific projects and encourage them to scale the approach to other projects. The results highlight the Cube's ability to adapt to varying SE maturity levels, providing organizations with a framework for implementing MBSE at their own pace while achieving measurable improvements in their development processes.

6. Conclusion & outlook

As part of the RePASE project, the MBSE Cube was developed to help companies transition from document-based to model-based approaches gradually and effectively. By considering maturity levels and technical needs, the Cube facilitates demand-oriented MBSE implementation through the creation of development artifacts. Over a one-year pilot phase, a designated change agent collaborated with academic partners, supported by a guideline documenting methods and tools. In the railway company, the primary demand was automated requirements validation. The Cube helped identify gaps in development artifacts, enabling structured modeling and verification. In the SME case, the focus was on introducing structured MBSE practices, facilitating functional trade studies and model-based subsystem selection, demonstrating its flexibility for different maturity levels. The case studies highlight three key aspects of MBSE adoption: Maturity Levels, which assess readiness and enable a gradual transition; Implementation Complexity, which involves managing the structured integration of MBSE within existing processes; and Internal Resistance, which demonstrates the feasibility and benefits of MBSE to facilitate acceptance and adoption. The findings confirm that MBSE adoption requires a structured, demand-based approach rather than a one-size-fits-all solution. The Cube enables a phased MBSE transition; however, several challenges remain. Aligning existing systems with abstraction levels, defining problem-to-solution transitions, and integrating domain-specific models present complexities. Addressing these challenges requires further refinement of the Cube, particularly by defining additional methods for each step and cell. These methods could be documented as structured guidelines or "method profiles," providing practical, step-by-step support for users.

Future work should focus on leveraging these advancements to refine the Cube's methodology, enabling smoother transitions and stronger connections between domain-specific and cross-domain models. This includes integrating comprehensive, specific product development methods, SysML v2, and advanced technologies such as AI to further enhance MBSE processes. These advancements could significantly strengthen the Cube's capabilities, particularly in artifact generation, process automation, and real-time maturity assessments, paving the way for more efficient and adaptive MBSE adoption.

Acknowledgement

"This research and development project is funded by the German Federal Ministry of Education and Research (BMBF) within the "Innovations for Tomorrow's Production, Services, and Work" Program (funding number 02J19B140) and implemented by the Project Management Agency Karlsruhe (PTKA). The author is responsible for the content of this publication."

References

Alt, O. (2017). *Modellbasierte Systementwicklung mit SysML*. Carl Hanser Verlag.

- Baughey, K. (2011). Functional and logical structures: A systems engineering approach. *SAE Technical Paper Series*. <https://doi.org/10.4271/2011-01-0517>
- Baschin, J., Schmidt, R., Schneider, D., Vietor, T., & Kizgin, U. V. (2023). Linking cross-domain information to support the development of complex systems. *Proceedings of the Design Society*, 3, 2455–2464. <https://doi.org/10.1017/pds.2023.246>
- Broy, M., Gleirscher, M., Kluge, P., Krenzer, W., Merenda, S., & Wild, D. (2009). Automotive Architecture Framework: Towards a Holistic and Standardised System Architecture Description.
- Chalé Góngora, H. G., Gaudré, T., & Tucci-Piergiovanni, S. (2013). Towards an architectural design framework for automotive systems development. In *Complex Systems Design & Management* (pp. 241–258). Springer.
- Chami, M., & Bruel, J.-M. (2018). A survey on MBSE adoption challenges. In *INCOSE EMEA Sector Systems Engineering Conference* (pp.1–16).
- Delligatti, L. (2013). *SysML distilled: A brief guide to the Systems Modeling Language*. Addison-Wesley.
- Eigner, M., & Koch, W., Muggeo, C. (2017a). Modellbasierter Entwicklungsprozess cybertronischer Systeme: Der PLM-unterstützte Referenzentwicklungsprozess für Produkte und Produktionssysteme. Springer.
- Eigner, M., Dickopf, T., & Apostolov, H. (2017b). The evolution of the V-Model: From VDI 2206 to a systems engineering-based approach for developing cybertronic systems. In J. Ríos, A. Bernard, A. Bouras, & S. Foufou (Eds.), *Product Lifecycle Management and the Industry of the Future* (pp. 382–393). Springer. https://doi.org/10.1007/978-3-319-72905-3_34
- Estefan, A. J. (2008). *Survey of model-based systems engineering methodologies*. INCOSE MBSE Initiative.
- Friedenthal, S., Moore, A., & Steiner, R. (2015). *A practical guide to SysML: The Systems Modeling Language* (3rd ed.). Morgan Kaufmann. <https://doi.org/10.1016/C2013-0-14457-1>
- Gausemeier, J., Czaja, A. T., Wiederkehr, O., Dumitrescu, R., Tschirner, C., & Steffen, D. (2013). Studie: Systems Engineering in der industriellen Praxis. *Tag des Systems Engineering 2013*.
- Henderson, K., & Salado, A. (2021). Value and benefits of model-based systems engineering (MBSE): Evidence from the literature. *Systems Engineering*, 24, 51–66. <https://doi.org/10.1002/sys.21566>
- INCOSE Technical Operations. (2007). *Systems Engineering Vision 2020* (INCOSE-TP-2004-004-02).
- International Organization for Standardization. (2023). *ISO 15288:2023: Systems and software engineering - System life cycle processes*.
- Kaiser, L. (2013). *Rahmenwerk zur Modellierung einer plausiblen Systemstruktur mechatronischer Systeme* (Dissertation, Universität Paderborn).
- Kleiner, S., & Kramer, C. (2013). Model-based design with systems engineering based on RFLP using V6. In M. Abramovici & R. Stark (Eds.), *Smart Product Engineering* (pp. 93–102). Springer. https://doi.org/10.1007/978-3-642-30817-8_10
- Krob, D. (2022). *Model-Based Systems Architecting – Using CESAM to Architect Complex Systems*. Wiley.
- Morkevičius, A., Aleksandravičienė, A., Mazeika, D., Bisikirskiene, L., & Strolia, Z. (2017). MBSE Grid: A simplified SysML-based approach for modeling complex systems. *INCOSE Int. Symposium*, 27, 136–150.
- Papke, B. L., Wang, G., Kratzke, R., & Schreiber, C. (2020). Implementing MBSE – An enterprise approach to an enterprise problem. *INCOSE International Symposium*, 30.
- Pfenning, M. (2017). *Durchgängiges Engineering durch die Integration von PLM und MBSE (Doctoral dissertation, Technische Universität Kaiserslautern)*.
- Pohl, K., Hönninger, H., Achatz, R., & Broy, M. (2012). *Model-based engineering of embedded systems*. Springer. <https://doi.org/10.1007/978-3-642-34614-9>
- Ramos, A.L., Ferreira, J.V., & Barceló, J. (2012). Model-Based Systems Engineering: An Emerging Approach for Modern Systems. *IEEE Transactions on Systems, Man, and Cybernetics, Part C*, 42, 101–111.
- Reichwein, A., & Paredis, C. (2011). Overview of architecture frameworks and modeling languages for model-based systems engineering. In *Proceedings of ASME*.
- Vaneman, W. K., & Carlson, R. R. (2019). Model-based systems engineering implementation considerations. *IEEE Systems Conference*. <https://doi.org/10.1109/SYSCON.2019.8836888>
- Weilkiens, T. (2007). *Systems engineering with SysML/UML: Modeling, analysis, design*. Morgan Kaufmann.
- Wilke, D., Pfeifer, S. A., Heitmann, R., Anacker, H., Dumitrescu, R., & Franke, V. (2022). Implementation of systems engineering: A maturity-based approach. *IEEE International Symposium on Systems Engineering (ISSE)*. <https://doi.org/10.1109/ISSE54508.2022.10005414>
- Wilms, R., Inkermann, D., Cemmasson, V. F., Reik, M., & Vietor, T. (2019). Distinction of domain-specific and cross-domain linkage types for engineering change management. In *Proceedings of the 22nd International Conference on Engineering Design (ICED19)*. <https://doi.org/10.1017/dsi.2019.118>