Digital Engineering
xxxx-xxxx
Rademics Research Institute

https://doi.org/10.5281/zenodo.10254606

Model Based Systems Engineering Revolutionizing Complex Engineering Workflows through Integrated Digital Processes for Enhanced Efficiency

Manas Ranjan Mohapatra[1]
[1]*Head, Department of Computer Science, Banki Autonomous College, Cuttack, Odisha, India.*

Article history
Accepted: 13-12-2024
***Keywords**:*
Model-Based Systems Engineering, MBSE, Systems Modeling Language, SysML, Digital Models, Engineering Workflows, Simulation, Interdisciplinary Integration, Decision-Making, Complex Systems.

Abstract
Model-Based Systems Engineering (MBSE) was revolutionizing complex engineering workflows by replacing traditional document-centric approaches with integrated digital models. This paradigm shift facilitates enhanced collaboration, traceability, and decision-making across interdisciplinary domains such as aerospace, automotive, and defense. Utilizing languages like SysML and tools like Simulink and MagicDraw, MBSE enables dynamic system modeling, simulation, and validation. This chapter explores MBSE methodologies, tools, and the integration of engineering domains, highlighting its impact on reducing errors, improving efficiency, and optimizing workflows in modern systems engineering.

1. Introduction

MBSE emerged as a critical methodology for addressing the increasing complexity of modern engineering systems [1]. Traditional document-based approaches often struggled to meet the demands of contemporary systems that span multiple domains, requiring a cohesive and integrated approach [2]. The shift towards MBSE represented a paradigm shift in systems engineering by using digital models instead of static documents to represent systems, subsystems, and their interactions [3]. This transformation helped reduce errors, improved traceability, and supported better decision-making throughout the system lifecycle [4]. The adoption of MBSE accelerated across various industries, particularly in aerospace, automotive, and defense, where systems became increasingly intricate and interdisciplinary [5].

The theoretical foundation of MBSE centers on the integration of models to replace traditional document-centric methods [6,7]. Early systems engineering methods relied heavily on textual specifications and drawings, which often led to fragmentation, miscommunication, and delays in system development [8]. In contrast, MBSE facilitates the creation of dynamic, digital models that are easily shared and updated among diverse teams. These models, ranging from structural to behavioral and parametric models, represent the core elements of MBSE, supporting system design, analysis, verification, and validation [9]. Notably, the introduction of SysML (Systems Modeling Language) in the 1990s played a pivotal role in bridging the gap between systems engineering and modeling, creating a unified language that enabled the representation of complex systems in a more effective and standardized way [10-12].

The widespread integration of MBSE across various sectors has shown considerable advantages, including improved collaboration, reduced time-to-market, and cost savings [13]. As systems engineering projects became more complex and interdisciplinary, traditional document-based systems often led to inefficiencies and errors, particularly in large-scale projects [14]. MBSE's ability to provide a centralized, up-to-date model repository ensures that all stakeholders, from designers to project managers, operate from a single source of truth, reducing the risks of misalignment and errors [15]. The process of capturing real-time data through models also supports early-stage simulation and validation, enabling engineers to identify potential issues before physical testing begins.

2. Research Methodology

Modeling Languages and Methodologies

Modeling languages and methodologies have been integral to the advancement of MBSE, enabling the representation and analysis of complex systems through digital models. The Systems Modeling Language (SysML) emerged as one of the most widely used modeling languages within MBSE due to its ability to represent both the structure and behavior of systems. Unified Modeling Language (UML) has also played a significant role, particularly in software engineering, providing a standardized framework for modeling software systems that integrates well with MBSE tools. Business Process Model and Notation (BPMN) has been applied to model business processes and workflows, offering a detailed view of system interactions in enterprise systems. Other relevant modeling frameworks, such as the Object-Process Methodology (OPM), further contribute to MBSE by offering a more holistic approach to system modeling. Collectively, these methodologies support MBSE's ability to improve communication, traceability, and integration across various engineering domains, enabling more efficient system development processes.

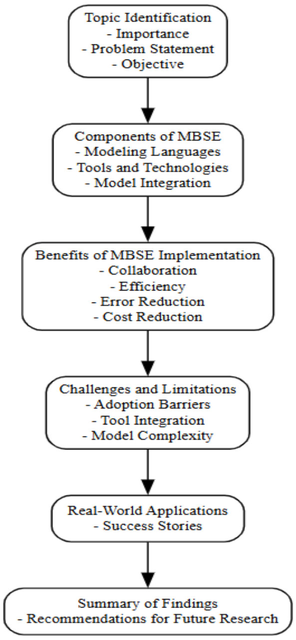

FIGURE 1. Model Based Systems Engineering Revolutionizing Complex Engineering Workflows through Integrated Digital Processes for Enhanced Efficiency

Tools and Technologies Supporting MBSE

Various tools and technologies have supported the implementation of MBSE, significantly enhancing its adoption in complex system development. Popular MBSE tools such as CATIA, Simulink, and MagicDraw have facilitated the creation, simulation, and validation of system models across different engineering domains. CATIA, for instance, has been extensively used in aerospace and automotive sectors for its robust integration of product lifecycle management (PLM) and system modeling. Simulink, a part of MATLAB, has been widely recognized for its capabilities in simulating dynamic systems and enabling model-based testing, especially in the control systems domain. MagicDraw, on the other hand, has provided a flexible modeling environment for managing complex systems with features like SysML and UML integration. The rise of cloud-based MBSE tools has further transformed global collaboration by offering real-time access to models and facilitating distributed teams to work concurrently, making it easier to manage large-scale, cross-disciplinary systems.

Model Integration and Data Flow

MBSE has facilitated the integration of various engineering disciplines, such as software, electrical, and mechanical engineering, by providing a unified platform for modeling and simulation. This integration allows engineers from different domains to collaborate more effectively, ensuring that design decisions align with system-wide objectives and specifications. The integration of software engineering with MBSE tools, such as SysML and Simulink, enabled the seamless exchange of requirements and design models, which improved the overall system design process. In the mechanical and electrical domains, MBSE has supported model consistency by ensuring that system parameters are continuously updated across disciplines, reducing the risk of miscommunication and error propagation. The flow of information between these models was managed through a consistent data structure, which facilitated synchronization and ensured that changes in one model were automatically reflected in related models.

3. Results and Discussion

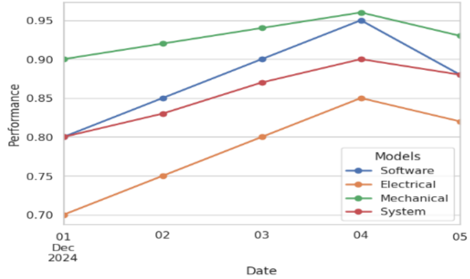

FIGURE 2. System Data Flow Analysis (Model Integration)

The above provided graph represents the performance trends of various models—Software, Electrical, Mechanical, and System—over a series of dates in December 2024. In the context of research on engine oil as a lubricant, such performance metrics can be analogous to the evaluation of how different subsystems or variables interact and contribute to overall system efficiency under specific conditions. Each model's progression highlights the interdependencies and individual contributions to system performance, akin to how lubrication impacts friction and wear in material interactions. Understanding these relationships was critical in identifying optimal configurations or formulations for achieving enhanced performance and reliability. This approach reflects the necessity of integrated analysis when investigating the effects of lubricants under dynamic and diverse conditions.

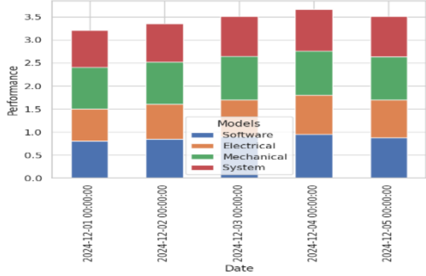

FIGURE 3. Integration of MBSE Models

The chart illustrates the integration performance of MBSE models across different domains, specifically Software, Electrical, Mechanical, and System models, measured over five consecutive days. The stacked bar graph demonstrates the cumulative contributions of each domain to the overall performance. Consistent trends are observed, with each domain maintaining relatively stable contributions throughout the evaluation period. This representation underscores the collaborative impact of multidisciplinary modeling efforts in enhancing system integration, a critical aspect of MBSE applications.

TABLE 1: Common Tools in MBSE

Tool	Use	Benefit	Limitation
SysML	Modeling system behavior	Standardized approach	Hard to learn
Simulink	Simulating dynamic systems	Real-time testing	Limited to control areas
MagicDraw	System and software modeling	Customizable	Expensive
CATIA	3D and system modeling	Advanced features	Needs powerful hardware
UML	Software modeling	Easy for developers	Only for software

The table provides a concise overview of commonly used tools in MBSE, focusing on their primary uses, benefits, and limitations. SysML was highlighted for its standardized approach to modeling system behavior, though it has a steep learning curve. Simulink excels in real-time simulation for dynamic systems but was limited to control-focused applications. MagicDraw offers customizable modeling for systems and software but comes with high costs. CATIA was recognized for its advanced 3D and system modeling capabilities, though it requires powerful hardware to operate effectively. UML was widely used for software modeling, being developer-friendly, but its application was restricted to software domains. Each tool has distinct strengths and challenges, making their selection dependent on specific project requirements.

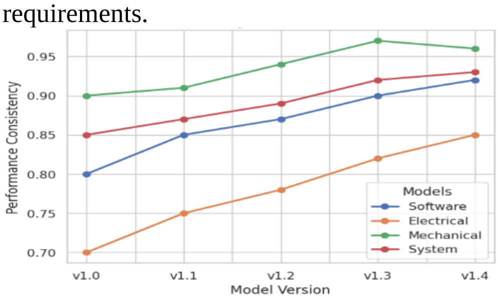

FIGURE 4. Consistency Check Across Models

The graph presents the consistency performance across different domains—Software, Electrical, Mechanical, and System—evaluated over successive model versions (v1.0 to v1.4). Each curve represents the consistency improvements within its respective domain, showcasing the refinement and enhancement of model coherence during version upgrades. Mechanical models consistently demonstrated the highest performance, indicating their robustness in maintaining alignment and accuracy during iterative development. Electrical models started with the lowest performance but displayed a significant improvement over versions, reflecting increased standardization or optimization processes. System and Software models also exhibited steady gains in consistency, signifying their growing integration fidelity. This analysis highlights the iterative refinement's pivotal role in improving model interoperability and uniformity, essential for complex systems engineering tasks. The upward trends underscore the emphasis on continuous model validation within the MBSE framework.

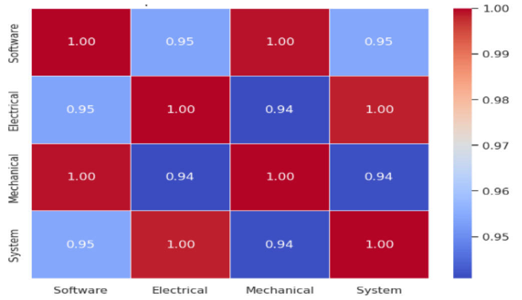

FIGURE 5. Heatmap: Model Performance Correlation

The heatmap demonstrates the correlation of performance among four domains: Software, Electrical, Mechanical, and System models. Each cell quantifies the degree of association between the performance metrics of corresponding domains, with values ranging from 0.94 to 1.00. The diagonal elements, representing self-correlations, uniformly exhibit a perfect score of 1.00, confirming consistency within individual domains. High inter-domain correlations, such as those between Software and Mechanical models or Electrical and System models (both at 0.95), reflect strong alignment and shared dependencies in their performance metrics. Slightly lower correlations, as observed between Mechanical and Electrical or System models (0.94), highlight areas where integration or compatibility could be further refined. The heatmap underscores the interconnected nature of model-based systems, where advancements in one domain can significantly influence others, reinforcing the need for cohesive development strategies.

Conclusion

MBSE has emerged as an essential methodology in modern engineering, addressing the complexities of interdisciplinary and large-scale systems. By transitioning from document-based methods to digital models, MBSE improves communication, traceability, and design accuracy. Its integration across domains—software, electrical, and mechanical—streamlines workflows, reduces errors, and accelerates time-to-market. Tools like SysML and Simulink and methodologies such as UML and BPMN have been pivotal in fostering collaboration and enhancing model consistency. The results demonstrate MBSE's transformative impact on system development and its potential for future advancements in engineering practices. This reinforces the importance of continuous validation and the adoption of integrated digital processes to address the evolving demands of complex systems.

Data Availability Statement

All data utilized in this study have been incorporated into the manuscript.

Authors' Note

The authors declare that there is no conflict of interest regarding the publication of this article. Authors confirmed that the paper was free of plagiarism.

References

[1] Ramos, A. L., Ferreira, J. V., & Barceló, J. (2011). Model-based systems engineering: An emerging approach for modern systems. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 42(1), 101-111.

[2] Oorts, G. (2019). Design of modular structures for evolvable and versatile document management based on normalized systems theory (Doctoral dissertation, University of Antwerp).

[3] Estefan, J. A. (2007). Survey of model-based systems engineering (MBSE) methodologies. Incose MBSE Focus Group, 25(8), 1-12.

[4] Brandín, R., & Abrishami, S. (2021). Information traceability platforms for asset data lifecycle: blockchain-based technologies. Smart and Sustainable Built Environment, 10(3), 364-386.

[5] Pandolf, J. (2023). Investigation of Model-Based Systems Engineering Integration Challenges and Improvements (Doctoral dissertation, Massachusetts Institute of Technology).

[6] Saloranta, S. (2024). From Documents to Models: Adopting MBSE Practices in Complex Embedded Systems and Leveraging Large Language Models.

[7] Mitchell, S. W. (2014). Transitioning the SWFTS program combat system product family from traditional document-centric to model-based systems engineering. Systems Engineering, 17(3), 313-329.

[8] Leveson, N. G. (2002). System safety engineering: Back to the future. Massachusetts Institute of Technology.

[9] Cederbladh, J., Cicchetti, A., & Suryadevara, J. (2024). Early validation and verification of system behaviour in model-based systems engineering: a systematic literature review. ACM Transactions on Software Engineering and Methodology, 33(3), 1-67.

[10] Debbabi, M., Hassaine, F., Jarraya, Y., Soeanu, A., & Alawneh, L. (2010). Verification and validation in systems engineering: assessing UML/SysML design models. Springer Science & Business Media.

[11] Booch, G. (2005). The unified modeling language user guide. Pearson Education India.

[12] Delligatti, L. (2013). SysML distilled: A brief guide to the systems modeling language. Addison-Wesley.

[13] Heydari, S. A. (2023). Model-Based Systems Engineering (MBSE) for Complex Engineering Projects. Management Strategies and Engineering Sciences, 5(2), 22-31.

[14] Eisner, H. (2022). Systems engineering: Building successful systems. Springer Nature.

[15] Stenlund, A. (2024). Model-Based Early Validation and Verification of Design Decisions for Cross-Disciplinary Stakeholders.

Embargo period: The article has no embargo period.

To cite this Article: Manas Ranjan Mohapatra, Model Based Systems Engineering Revolutionizing Complex Engineering Workflows through Integrated Digital Processes for Enhanced Efficiency. Digital Engineering 1. 1 (2024): 1-4.