International Journal of Science and Research (IJSR)
ISSN: 2319-7064
Impact Factor 2024: 7.101

Transforming Aerospace Mission Planning: A Comparative Study of MBSE Strategies and Tools

Astha Pandit[1], Priyangshu Sarkar[2], Utsav Saroha[3], Shaurya Singh[4], Satvik Kodali[5], Sana Asrar[6]

[1]Budhanilkantha School, Budhanilkantha, Nepal
Email: *asthapandit012[at]gmail.com*

[2]Springdale High School (H.S.), A3 City Centre Complex, Kalyani, West Bengal 741235, India
Corresponding Author Email: *therealpriyangshu[at]gmail.com*

[3]Imperial Heritage School, Dwarka Expy, Gurugram, Haryana, India
Email: *utsavsaroha78[at]gmail.com*

[4]St. Karen's High School, Gola Road, Danapur, Patna, Bihar, India
Email: *singhshaurya262008[at]gmail.com*

[5]Sharanya Narayani International School, #232/1, Thoranahalli PO, Off Malur - Byranahalli Road near Hoskote, Whitefield, Bengaluru, Karnataka 563130, India
Email: *satvik.kodali[at]gmail.com*

[6]Islamic University of Science and Technology, Awantipora, India
Email: *sanaaasrar2810[at]gmail.com*

Abstract: ***Model-Based Systems Engineering (MBSE), a model-centric framework for managing complex aerospace missions, emphasises improved traceability, early fault detection, and multidisciplinary integration. This review examines key MBSE tools such as MagicDraw, IBM Rhapsody, and Capella, along with the core modelling strategies–requirements, structural, and behavioural. Applications in avionics, spacecraft design, and autonomous systems are explored, highlighting MBSE's advantage over traditional engineering methods. Challenges, including steep learning curves and interoperability issues, are also addressed. The paper identifies gaps in current MBSE research and suggests future directions in agile integration and digital twin development.***

Keywords: Aerospace Mission Planning, Model-Based Systems Engineering (MBSE), SysML, MBSE Tools, Engineering System Integration

1. Introduction

Technological advancements have led to digitisation and complications in the engineering field. As the complexity of systems and their context grow, traditional document-based systems engineering, once deemed robust, struggles to maintain pace with the increasing scale and complexity requirements of modern systems, rendering them obsolete [1]. Over the past few decades, model-based systems engineering (MBSE) has become an industry-standard specific to aerospace, adopted and advocated by institutes such as the National Aeronautics and Space Administration (NASA), Lockheed Martin, Airbus, Boeing, as well as academic centers like the Georgia Institute of Technology's Center for MBSE and the University of Michigan's MBSE Leadership Lab in the United States alone [2]. This is evident when, in January 2020, NASA noted this trend by reporting that MBSE, "has been increasingly embraced by both industry and government as a means to keep track of system complexity."[3]

This shift is embodied by the fact that multifaceted projects, such as those in the aerospace industry, where the integration of subsystems like propulsion, payloads, and environmental controls is critical, often fall behind in maintaining reliability and performance when integrated across multiple disciplines. The solution to this is Model-based Systems Engineering (MBSE). As a structured method, MBSE enables engineers to systematically manage system requirements and design processes from the conceptual phase through development and later operational stages, ensuring consistency and reducing documentation errors [4]. The difference from the traditional documentation-based approach, concerning MBSE, lies in the methods' focus: while MBSE uses models, document-based systems are purely document-centric [3]. The increasing use of digital modelling environments in recent years has driven the expansive adoption of MBSE, which is anticipated to play a central role in the future of managing complex engineering projects. The International Council on Systems Engineering (INCOSE) predicts that MBSE will gradually surpass traditional document-centric methods and become the fundamental approach for systems engineering in the future [5].

By shifting from document-heavy practices to modelling practices, MBSE addresses challenges such as data inconsistencies, integration failures, and errors that frequently result from poor coordination among various system components. Furthermore, the designing of aerospace systems encompasses complex requirements such as autonomous operations, seamless collaboration among teams, and the reliable integration of diverse system elements.. Minute errors in the aerospace industry can lead to huge losses: whether that may be mission failure, huge financial losses, or even human lives. However, despite the growing recognition of MBSE's potential and applications, there is little research that critically analyzes MBSE's tools and strategies as applied to real-world private and public space agencies on a broader scale. Most research focuses on MBSE's applications in the aerospace industry, tool-specific capabilities as applied to a particular aspect of the aerospace industry. Therefore, it is of utmost importance to comprehensively evaluate the role of MBSE, its tools and modelling strategies, and challenges in modern aerospace missions.

In this systematic review, we conduct a comprehensive evaluation of the different modelling tools and strategies used in MBSE, analyze their underlying implications and limitations, and provide a comparative analysis with traditional system engineering in the aerospace industry. We begin by outlining the scope and objectives of MBSE in aerospace engineering, followed by an in-depth analysis of the different modelling languages and tools such as SysML, MagicDraw, Cameo, Capella, and IBM Rhapsody. Then, we examine and describe key modelling strategies - requirement, structural, and behavioural modelling, providing insights into their working principle, and illustrating how these techniques can enhance complex projects to greater strengths. Next, we discuss the implementation process, justification for adoption, and targets of implementation in critical space environments. This includes a detailed description of MBSE's functionality, early fault detection and its management, traceability, and reusability of the system models. We also explore the application of MBSE across various aerospace domains - planetary, interplanetary, and deep space missions, including propulsion, avionics, and autonomy. Furthermore, increasing adoption of advanced technologies like AI, digital twin, collaborative clouds, and future integration of SysML v2 into aerospace projects are also studied. Finally, we address the limitations and challenges associated with the adoption of MBSE and identify possible research gaps in the current literature. This review offers a comprehensive analysis of MBSE tools, strategies, and technological integration into the aerospace industry.

2. Background and Literature Review

2.1 Scope and Objectives of MBSE in Aerospace Missions

The scope of Model-Based Systems Engineering (MBSE) in aerospace missions covers the entire lifecycle of the system - from the conceptual design phase through development and into later lifecycle stages. In mission and performance-critical industries like aerospace, which are highly regulated, interdisciplinary, and complex, relying on traditional document-based engineering and acquisition lifecycle model approaches often leads to inconsistencies and poor integration across various domains and systems. This challenge necessitates a model-centric approach, with MBSE playing a crucial role in ensuring that software code, subsystems, and interfaces align with the system requirements and mission objectives. [6]

This part outlines the primary objectives of applying MBSE within the aerospace sector:

2.1.1 Improved requirement specification and management:

MBSE emphasizes establishing mission goals and necessary requirements at the earliest stages of the development process. It involves clearly stating these requirements before moving on to design creation and system verification, ensuring that the entire scope of the problem is addressed from the outset. This ensures consistency and traceability between stakeholder needs, system requirements, and design artifacts, reducing misinterpretations and fostering better and more efficient communication.

2.1.2 Enhanced system integration

Given that aerospace missions incorporate numerous subsystems—such as avionics, mission payloads, propulsion, thermal regulation, and electrical control—MBSE supports effective coordination between thermal engineers and system architects during habitat design. It facilitates the seamless integration of these diverse subsystems into a unified one. This integration allows for a more holistic analysis of the system, ensuring that all components and subsystems work together as intended. [7][6].

2.1.3 Detection of early design flaws and Risk Minimization:

Through early validation and verification, aerospace engineers can identify design flaws and integration issues, minimizing costly designs of physical prototyping and testing iterations.

2.1.4 Support for digital engineering:

Space missions enable Artificial Intelligence (AI) and digital twins to expand MBSE to autonomy and scalable designs. It allows for automated analysis to predict design outcomes, real-time monitoring, and advanced simulation, overall facilitating better lifecycle management. MBSE provides a foundation for the creation of digital twins, which are virtual models representing physical assets. Through the integration of real-time data and analytics, MBSE enables predictive maintenance, continued performance optimization, and troubleshooting, reducing downtime and ultimately the total cost of ownership.

A paper by Marshall Pratt and Matthew Dabkowski mentions MBSE to alleviate the strain put on the aerospace industry due to the delicate, regulatory nature and longevity of their products compared to more traditional methods. Indeed, private organisations such as Boeing and Lockheed Martin seem to advocate for MBSE, displaying its importance to the private aerospace sector [2].

2.2 Key MBSE Tools and Technologies

Within a Model-Based Engineering framework, the use of SysML models as the primary design artifacts emphasizes the importance of explicitly capturing assumptions regarding both system components and their operational environment within the models themselves. By embedding the Systems Modelling Language (SysML) within a formal logic and formal methods, can be used to maintain consistency as the design evolves. SysML's formal semantics enable engineers to employ reasoning in the course of a typical model-based development process. SysML has a demonstrated capability for top-to-bottom design refinement for large-scale aerospace systems. [8] This study employed a comparative qualitative analysis of leading MBSE tools used in aerospace mission planning. The evaluation focused on four major platforms: MagicDraw with SysML, IBM Engineering Systems Design Rhapsody, Capella, and selected auxiliary tools such as Simulink and ModelCenter.

2.2.1 MagicDraw

MagicDraw, when used in conjunction with Cameo Systems Modeller, was evaluated for its robust model traceability capabilities and its wide-ranging plugin ecosystem [9]. As one of the leading MBSE platforms in aerospace, it supports comprehensive SysML modelling and enables end-to-end traceability across requirements, structural components, and behavioural flows. Its usability and integration with external tools such as DOORS and Simulink enhance collaborative design efforts and support conformance to standards like ISO/IEC 19514 (SysML) [10]. MagicDraw's adaptability to mission-specific workflows, including its vigorous plugin system and collaborative modelling features, makes it a favoured choice for both public and private aerospace programs [11]. NASA employed MagicDraw in the design and development of small aircraft systems to construct and validate system architecture models.

2.2.2 IBM Rhapsody

IBM Engineering Systems Design Rhapsody's strengths lie in simulation and automatic code generation, making it highly suitable for aerospace systems with real-time and embedded software needs [12]. The tool supports SysML and UML and is frequently applied in safety-critical systems within organisations such as NASA and ESA. It shows the digital representation of the system, like structure, requirements and design. Its use of the Harmony SE modelling process streamlines requirement analysis, functional modelling, and design synthesis. Rhapsody also supports continuous validation through executable models and integrates well with existing toolchains, contributing to enhanced system integration and reduced development risk, including facilitating a functional decomposition. [13].

2.2.3 Capella and Other Tools

Capella, developed by the Eclipse Foundation using the Arcadia method, was reviewed for its effectiveness in early-phase architectural modelling. Using the ARCADIA method, Capella can be integrated with SysML to provide a guided approach to systems architecture with strong visualization and semantic clarity, particularly useful during the concept and preliminary design stages [14]. Its simplified interface and accessibility make it attractive for educational use and teams new to MBSE. In addition, supporting tools like Simulink and ModelCenter were analysed for their capabilities in system simulation, parameter analysis, and cross-domain co-simulation.

These tools complement core MBSE platforms by enabling iterative testing, rapid prototyping, and integration of multidisciplinary subsystems. They allow engineers to create detailed models of complex systems, including their structures, behaviour, and interactions, with powerful simulation capabilities to analyze these models.

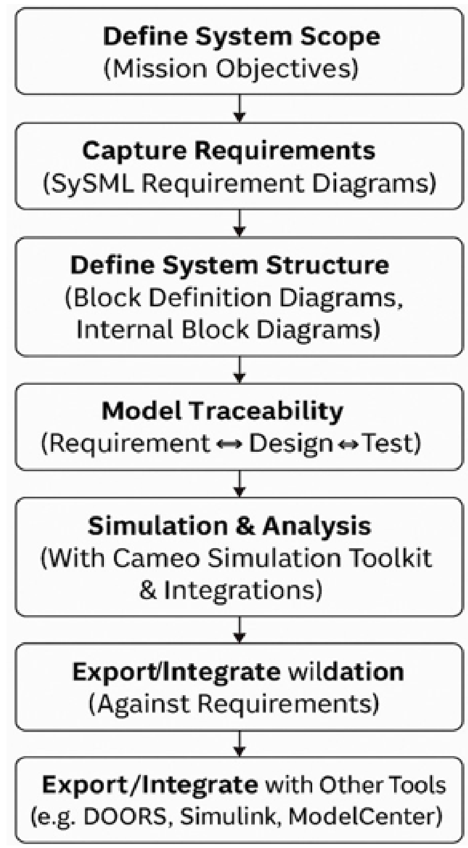

Figure 1: MagicDraw processing flowchart using SysML

2.3 Modelling Strategies in MBSE

By leveraging system models and simulations, MBSE provides a comprehensive approach to managing the requirements, design, and behavior of space and mission systems. These models are most popularly depicted using SysML diagrams, which provide a standardised way to model systems throughout their life cycle. SysML includes nine diagrams: requirement diagram, activity diagram, sequence diagram, state machine diagram, use case diagram, block definition diagram, internal block diagram, parametric diagram, and package diagram.

The three major modelling strategies derived from the SysML framework, which are widely implemented for engineering spacecraft, satellites, and space probes, are discussed below.

2.3.1 Requirement Modelling:

Requirement modelling creates a visual representation of the system's requirements using diagrams in MBSE. It utilizes requirement diagrams organized in a containment hierarchy to clearly depict the relationships and dependencies among requirements. These diagrams represent text-based requirements and show their links to other requirements, design components, and test cases, thereby facilitating requirement traceability [15]. It helps stakeholders understand the system's functionality and ensures alignment between the aerospace engineering and technical teams. Requirements modeling offers a clear visualization of system behavior, user interactions, and data flows, thereby minimizing ambiguity and miscommunication. Since requirements often evolve over time, employing effective requirements modeling tools for spacecraft, satellites, or space missions enables accurate tracking of these changes and their proper incorporation into the final system. MBSE uses requirement modelling to improve inconsistencies within the system, reduce error propagation, and enhance system behaviour and performance.

2.3.2 Structural Modelling:

Using Block Definition Diagrams (BDD) and Internal Block Diagrams (IBD), structure modelling defines the relationship between the system's logical and physical architecture. Block Definition Diagrams (BDD) illustrate structural elements called blocks, highlighting their composition and classification. Conversely, Internal Block Diagrams (IBD) depict the connections and interfaces among parts and blocks. Parametric diagrams capture constraints on property values, while package diagrams organize the model by grouping related elements into packages. SysML structure diagrams use blocks as basic structure elements and communicate via messages. Given the intricate and interdisciplinary nature of spacecraft, the estimation of all the model parameters from experimental measurements may quickly become intractable. Structural modeling techniques are therefore employed to calculate model parameters using knowledge of the structure's geometric and mechanical properties [16]. Structural modelling reveals internal architecture details, helps revise structural requirements, and forms a backbone for block-to-block interaction modelling, deeming it a vital tool for the aerospace industry.

2.3.3 Behavioural Modelling:

Behaviour modelling is a description of how the proposed system will interact during mission operation and execution. SysML provides various diagrams—including activity diagrams, state machines, sequence diagrams, and use case diagrams—to formally represent system behaviors. Specifically, an activity diagram models system behavior by illustrating the sequence of actions based on the flow of inputs, outputs, and control signals. A state diagram illustrates how a spacecraft or a satellite component interacts with other components, capturing distinct states with appropriate transitions through external events. While sequence diagrams model interactions between blocks, showing ordered message exchanges and timing relationships, use case diagrams represent functionality in terms of how a system or other entity is used by external entities to accomplish a goal [17]. Together, these diagrams enable executable specifications, support early verification, and reveal inconsistencies in the system before implementation.

Modelling Strategies	Key Diagrams	Application	Reference
Requirement Modelling	Requirement Diagram, Use Case Diagram	**SpaceX**: Uses requirement models to define specifications for rapid-launch Falcon 9 systems	[18]
Structural Modelling	Block Definition Diagram (BDD), Internal Block Diagram (IBD), Parametric diagram, Package diagram	**Blue Origin**: Uses structural modelling for reusable New Shepard architecture	19
Behavioural Modelling	Activity Diagram, Sequence Diagram, State Machine Diagram, Use case diagram	**ESA**: Implements safe-to-science operational transitions, as in ESA's Integral mission	[20]
Verification and Validation	Traceability Matrices, Dependency Diagrams	**NASA**: Artemis mission applied full traceability to manage complex interdependencies across mission systems	[21]

Figure: Table comparing Model-Based Systems Engineering (MBSE) modelling strategies and their application in various private and public aerospace organizations.

2.4 System Function, Fault Analysis, and Maintainability

2.4.1 Functional Analysis

MBSE enables the automation of functional interfaces for all scenarios, and it enhances traditional systems engineering by using formalized modes to support the entire system from requirements to verification. Analysing its ability to function, we get a better understanding of this advanced system and how to implement it efficiently.

MBSE can trace between different sequence diagrams and generate top-level functions; furthermore, the proposed framework can generate executable actors' behaviour to prepare functional simulations and create feedback with continuous validation. The functional architecture states that the goal of MBSE is to refine each high-priority function with low-level technical functions identified from domain knowledge while considering non-functional requirements. SysML *Activity* is used as a behavioral container, *Action* nodes to express behavioral elements, and *Control Flow* to specify their execution logic. In addition, functional interfaces of the SoI are specified using *Send Signal Action* (to send a signal), *Accept Event Action* (to wait for a given event), and "call behavior actions" represent function calls. [22]

2.4.2 Fault Detection and Risk Management

The fault detection in MBSE starts at the system level function, that is, it identifies the failure modes of the model by examining the tasks and associating relevant parameters. Primarily, it can detect the breakpoints of various parameters. Secondly, it identifies the sub-system-level failures which may be associated with the system-level parameters; furthermore, it uses the SysML's parameters diagram to point out the impact chain. Without the need for a repeat analysis (as the relation between components and subsystems is built within the model), finally, at the component level, it identifies the core faults and judges their causes. The faults may cause changes in the parameters associated with the component level, but based on the parameter change, it calculates the data at the subsystem level and evaluates the parameters within the predetermined thresholds, thereby assessing the impact of faults on the system. To ensure the completeness and hierarchical nature of the fault model, this approach considers capturing the parameters corresponding to each function. This classifies functional faults at different levels based on the function's completeness. As a result, functional faults are influenced by faults at all levels [23].

Organizations like NASA follow the MBSE method and also use the *Risk Management Plan* (RMP). The Risk Management Process (RMP) outlines the procedures for researching, evaluating, and approving risks before they are incorporated into the fault-tracking system. Generally, this process includes notions like 'L' for likelihood and 'C' for the consequence of occurrence. RMP also includes a criterion of when a risk can be closed, which generally occurs when 'L' and 'C' are below a certain threshold of concern. The fault detection is a prior action before the implementation of RMP, and finally, it all comes down to the *Risk Matrix Scoring,* which is the difference between the magnitude of the outcome result and the expected result [25].

2.4.3 Maintainability and Lifecycle Support

In the context of MBSE, maintainability is integrated early in the design process, which allows engineers to evaluate and optimize repair times, access strategies, and failure mode responses before system implementation. *Mean Time To Repair* (MTTR) is a modelling key parameter that gives designers access to simulate real-life world scenarios and identify design features that hinder serviceability. MTTR is a joining function of Risk management and Maintainability [25].

Through the use of SysML and Logical Architecture Frameworks, MBSC provides a comprehensive model that evolves from concept to retirement. It plays a vital role in enhancing lifecycle support for CubeSat missions by enabling a structured and integrated approach to system design and maintenance. It supports all operational phases, launch, early operations, and normal functioning, ensuring consistency and traceability across all disciplines. MBSE highly reduces dependency on fragmented documents by offering a unified, authoritative data source for engineering decisions. Though it needs an approach for learning, adoption, and encouragement of practices for early development and mission completion, MBSE significantly improves CubeSat systems' reliability, operational longevity, and end-of-line planning, making it a vital tool for supporting the complete lifecycle of modern satellites [26].

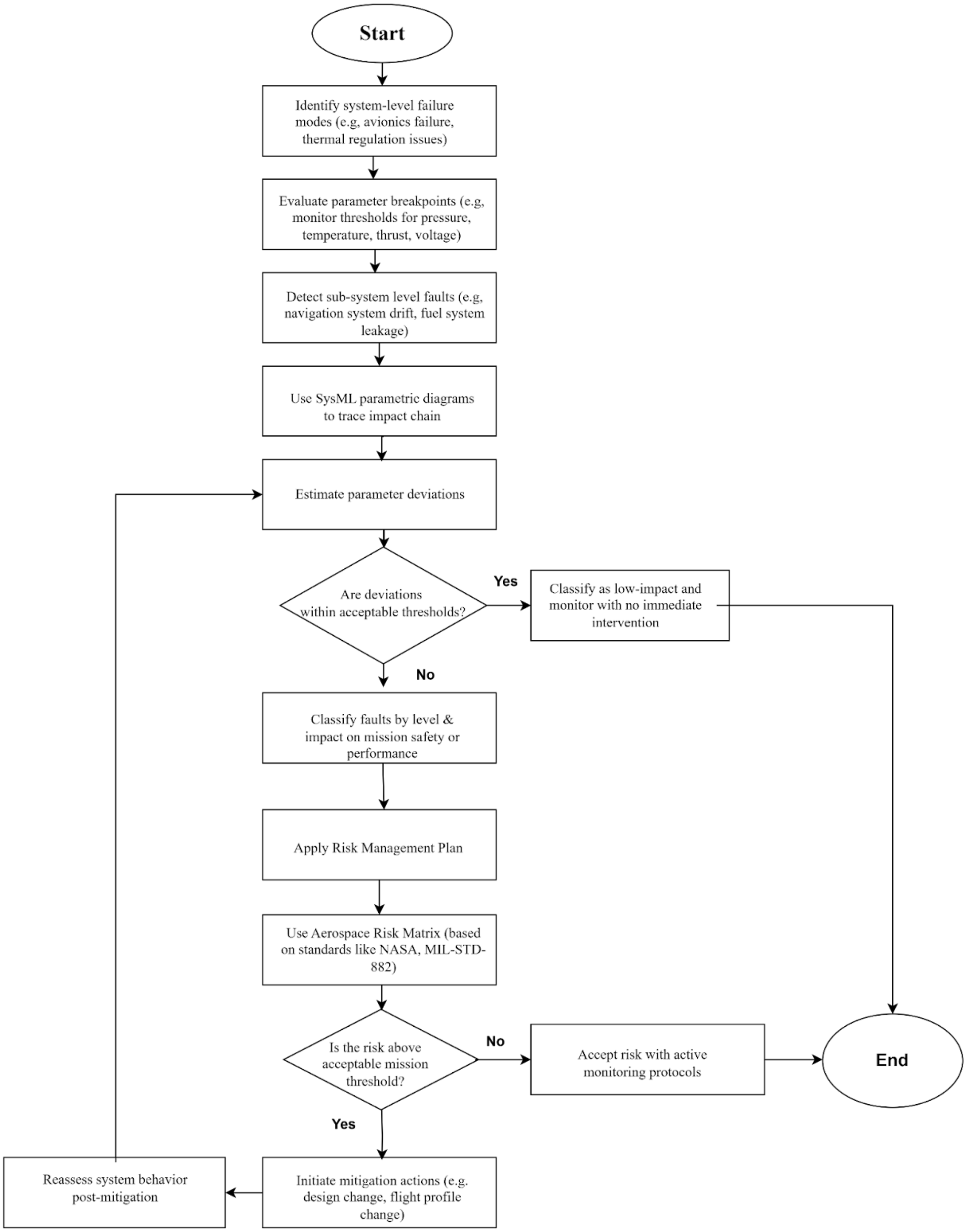

Fig: Model-Based Systems Engineering (MBSE) Fault Detection and Risk Management Flowchart for Aerospace Engineering Systems

2.5 Traceability and Reusability in MBSE Models

In the context of MBSE, traceability touches on the potential to link and steer between different model artifacts (e.g., requirements, design elements, test cases) to ensure stability, manage alerts effectively, and hold up confirmation and authentication [6]. Traceability empowers engineers to comprehend the consequences of planned conclusions and preserve coordination between system-level necessities and their execution in different model domains [27]. Reusability, on the other hand, targets leveraging existing model components or subsystems across distinct projects or product lines to diminish growth time and costs while guaranteeing uniform quality [28]. Reusable model collections and patterns allow teams to combine previously confirmed elements, enhancing efficiency and decreasing errors. In this study, we will integrate both traceability and reusability guidelines into the representation process to ensure methodical linking of requirements to design and testing operations while reusing found modelling modules wherever conceivable [29].

3. Discussion

Applications of MBSE in Aerospace Engineering Projects

MBSE has developed as a revolutionary approach for the growth of complicated systems in the aerospace and avionics industries. In aerospace, MBSE is used to manage the development of complex aircraft systems, where critical integration of subsystems is required. Although MBSE concepts emerged in the 1980s and 1990s, implementation of MBSE at NASA began through the MBSE Pathfinder project in 2015, and the MBSE Infusion and Modernization Initiative (MIAMI) launched in 2016. In 2016, NASA's MBSE Pathfinder team applied MBSE to model complex mission scenarios, exploring how astronauts could live, work, and sustain themselves on the surface of Mars. They focused on In-Situ Resource Utilization (ISRU) - an approach where astronauts use materials already on Mars to build shelter. MBSE was instrumental in enabling engineers to visualize the interactions between ISRU systems and key mission elements, including life support systems and habitats. By creating detailed system models, the team could simulate different mission designs, run trade studies, and test 'what-if' scenarios before any physical systems were built. This improved early decision-making, along with making the mission planning more collaborative and flexible across NASA's centres [39].

Another prominent instance of MBSE in aerospace engineering in interplanetary missions is the successful landing and embarkment of one of the most complicated rovers to date: the Mars rover, by the Jet Propulsion Laboratory (JPL). Based on the experience of the Curiosity rover - where documentation of design's data made it difficult for standardization and to reuse all information effectively, model was carried out for the Mars rover, focusing on the key elements required to develop a system design using a SysML model, followed by simulations to verify the accuracy of the design data, project deliverables automatically generated, web-based model visualization, and to realize cross-team collaborative design and stakeholders regular inspection. In contrast to the traditional design mode, where designers must stop the design work and concentrate on preparing all kinds of materials necessary for the key review node of the project, - namely project deliverables, Mars2020 automatically generates as many documents as possible for project delivery from the model during the MBSE-based rover design process to save designers the preparation time required for project review. Together, MBSE is generated based on integrated system models across domains to ensure consistency of document information [31]. Also, MBSE is extensively applied to solve the problem of low informatization and digitization degree in the development of manned spacecraft. For instance, the China Academy of Space Technology emphasizes the necessity to put forward a specific model-based procedure specification for manned spacecraft development based on MBSE fusion. The full lifecycle model of manned spacecraft, combined with the development requirements of MBSE throughout the full lifecycle, the Manned Space Department takes the model as the carrier of development data, and integrates all aspects of product development through the requirement model, function model, product model, engineering model, manufacturing model, and implementation model. MBSE provides a comprehensive framework for a multidisciplinary and 3D collaborative simulation while managing the complexity of the systems [32].

The widespread adoption and application of MBSE is linked to its versatility. Institutions using SysML-based tools reported a 40% reduction in condition-associated errors, a clear symptom of MBSE's role in ensuring system ethics [6]. Organizations adopting MBSE have recorded quantifiable gains in efficiency. For instance, model-based practices at NASA led to a 30% drop in design time and a 20% drop in general development costs. MBSE encourages cooperation across historically isolated divisions by providing an integrated framework for participants. In case studies from the automotive industry, companies such as Volvo achieved a 50% improvement in cross-functional communication due to mutual model databases. This benefit converts across industries where cross-domain teams must align swiftly and effectively. Tools enable engineers to handle large-scale, ranked structures with simplicity. Models with over 10,000 factors have been effectively maintained in protection and manufacturing deployment [27]. Therefore, the versatility and effectiveness of MBSE make it an indispensable tool in the aerospace industry, where complex, interdisciplinary systems involve thousands of interconnected elements and relationships.

Current trends and future of MBSE in aerospace:

The aerospace industry represents the largest domains for applications of complex engineering projects and systems. As the aerospace industry continues to grow increasingly complex, MBSE also evolves, integrating emerging technologies to enhance scalability and efficiency. Some of the most prominent trends in the aerospace industry include the increasing and widespread integration of cloud platforms and cybersecurity, growing use of AI and digital twins, and future integration of SysML v2.

One emerging trend in MBSE is the adoption of collaborative and Teamwork cloud platforms. Cloud-based MBSE enables real-time access to models and data from different geographical locations. Collaborative modeling tools, such as Cameo Collaborator for Teamwork Cloud, present simplified model views for stakeholders, while Team for Capella enables multiple users to collaborate directly on the same system model. "Cloud-based platforms, such as the IBM Engineering Lifecycle Management suite, provide a centralized repository for system models, enabling teams located in different geographic locations to collaborate on the same project [40]". As MBSE in aerospace grows more interdependent and complex, platforms based on concurrent design principles improve efficiency while offering advanced analytics and reporting capabilities, and improve efficiency during the design process, fostering smarter and faster building. However, reliance on cloud solutions raises concerns about data security and intellectual property protection, especially given the sensitive nature of aerospace projects. Also, the need for stable, high-bandwidth connectivity may limit cloud adoption in certain organizations or regions.

Moreover, the integration of Artificial Intelligence (AI) and machine learning into MBSE enables more advanced data analysis, although their current maturity level in safety-critical applications remains limited. AI enables automated model creation from high-level specifications to validation and optimization. Automating manual tasks such as requirement analysis, model generation, and verification allows engineers to focus on more complex and creative aspects of system design. Enhanced decision-making by analyzing large amounts of data prevents costly faults and downtimes, while simultaneously resulting in a more robust and reliable system.

Similarly, increasing adoption of digital twins in MBSE to bridge the gap between the physical and digital worlds, represents system states in real time, and enables a new era of connected and intelligent systems. Digital Twins retain a high level of precision, faithfully reflecting the system's specifications, leading to more reliable and realistic representations of physical assets in their digital form. Since most interplanetary and deep space missions expose spacecraft to extreme environmental conditions, digital twins enable the prediction and analysis of system behavior under such scenarios. It helps to counter potential risks, thereby enhancing the reliability and cost of mission planning. The use of MBSE principles ensures that the Digital Twins are built on a structured foundation, promoting scalability and maintainability [36].

Also, SysML v2 is the next-generation modelling tool aimed at transforming MBSE adoption and addressing limitations of SysML v1, particularly the precision, expressiveness, usability, interoperability, and extensibility of the language. It is intended to promote broader adoption and enhance the effectiveness of MBSE by supporting better traceability through robust visualizations based on flexible view and viewpoint specifications. SysML v2 introduces enhanced language features that allow for the precise representation of various system aspects. Future Directions for MBSE with SysML v2, including its structure, behavior, requirements, analysis cases, and verification cases. Additionally, it improves SysML v1 by facilitating 'usage-focused modelling', which enables one to define a system design configuration with parts that may or may not have explicit definitions. These capabilities enable highly integrated and complex systems to have their characteristics, which hence evaluate the entire aerospace engineering systems' mechanisms [37].

4. Limitations and Research Gaps

It is important to acknowledge that no engineering methodology is without trade-offs; MBSE offers substantial benefits but also introduces new challenges that must be managed carefully. The most significant one is the steep learning curve associated with advanced MBSE tools such as MagicDraw and IBM Rhapsody. These tools are complex and require specialized skills, and a considerable amount of investment in software tools and training. They require a deeper understanding of modeling languages like SysML, which can be unfamiliar and challenging to engineers who have been trained in traditional, document-centric systems engineering. Many traditional engineers may resist this transition, preferring to stick to conventional paper-based methods. This process is both time-consuming and resource-intensive [35].

Interoperability presents another significant challenge. MBSE tools often do not work seamlessly with legacy systems or different toolsets. While platforms like Capella provide user-friendly, guided modeling—particularly useful in the early conceptual phases—they do not fully support SysML, which can hinder integration into broader, flexible engineering environments. Implementing MBSE is costly and requires extensive investments in time and resources. Licensing fees, the need for high-performance computing infrastructure, and customization efforts can be too costly for smaller aerospace contractors or new startups.

Moreover, although MBSE facilitates automation and early verification, it still depends heavily on human inputs in critical steps like behavior modeling and fault analysis, which reduces overall efficiency. In fast-paced commercial spaceflight industries, where speed and Agile methods are the driving factors, MBSE's structured and documentation-heavy approach may feel too rigid, often deterring adoption. Therefore, while MBSE has demonstrated significant value in structured public missions like NASA's Perseverance, its long-term impact depends on continued evolution—particularly in tool interoperability, industry-wide standardization, and the integration of emerging practices such as AI-driven modeling and digital twin synchronization. [38]

Thereby, several research gaps in this field should be addressed. First, the highly structured framework of MBSE makes it a rigid choice to combine with the agile methodologies. The integration of Agile in MBSE has been found to further enhance our understanding and effectiveness in system engineering. Second, as systems' complexity increases, the complexity of tools and systems used to model them also increases, making it a painstaking task to manage and analyze these models. Consequently, further research is necessary to develop more robust and standardized system frameworks, especially given that aerospace is among the most complex industries with highly interconnected subsystems [37]. Further research is essential to create interoperable MBSE tools that balance complexity and usability, and to develop best practices for integrating Agile with MBSE without compromising safety and compliance.

5. Comparative and Critical Assessment

For this study, we used a comparative qualitative approach to investigate how MBSE tools and strategies are reshaping aerospace mission planning across public agencies and private companies. Our study clearly reveals the role of MBSE in managing the complex aerospace industry, from building spacecraft and satellites to testing, validating, and executing these operations. The model-centric approach of the MBSE, in which the main artifact is a coherent model being developed, somewhat contrasts with traditional-based systems' engineering. For calculating the anticipation and calculated design errors, MBSE can be a powerful tool through the application of structured and disciplined methods, such as 1) Model Stimulation; 2) Requirements Traceability; 3) Automated Tests; and 4) Integration with other elements of the system in the digital system development environment. The future of MBSE is crucial for ensuring traceability and agility in adapting to evolving requirements, designs, and technologies. MBSE enables a more rapid response to changes in system requirements and design. The use of MBSE is to verify that our system satisfies requirements and surfaces issues early; through this, it will significantly reduce the adverse impact on program cost, schedule, and the risk of discovering these issues later.[37]

MBSE has a vast potential for lifting the aerospace industry to greater autonomy and sustainably. By keeping up and integrating the latest technological advancements in AI, digital twins, cloud collaboration, more advanced designing, analysis, and decision making, optimization is possible. These factors are highly required in industries such as aerospace, where real-time monitoring and environmental factors are highly crucial for safe functional systems. Given the fast-paced innovation in science and technology, MBSE faces considerable difficulties in its scalability. Although many studies suggest MBSE reduces costs and risks, empirical evidence remains limited, with most successes reported in large-scale projects. Small and mid-sized projects may not experience the same level of benefit due to resource constraints.

In spite of these numerous benefits, MBSE faces several challenges. The steep learning curve to adopt MBSE tools like MagicDraw, Capella requires huge resources, a deeper understanding, and paramount time investments. Resistance to the adoption of this unfamiliar and challenging change shifts the environment for traditional document-centric methods. Their rigid structure, interoperability, high-cost infrastructure, and need for regular human validation limit their full potential and scalability. These drawbacks call attention to more investment in training and education for engineers in the early stages, along with the development of more flexible, scalable, standardized frameworks and cost-effective tools.

The paper by Friedenthal highlights, “MBSE is a promising solution to complex products and developing programs.”By leveraging MBSE, engineers can create elongated digital models that can integrate various subsystems and their components, which allows the system to have an efficient design simulation and validation. The ever-increasing complexity of systems and demand for shorter development cycles and lower costs require a more efficient and less error-prone communication and developmental approach. Instead of natural languages, MBSE adopts different models as the basic elements to store and transfer data. Hence, MBSE has demonstrated strong potential in addressing these challenges.

Aerospace engineering aspects	Before MBSE	After MBSE
Design approach	Follows a step-by-step, document-heavy process	Uses a parallel, model-focused method that integrates across domains
Requirement Handling	Requirements are written in plain text, which may be misinterpreted	Requirements are represented using structured models for clarity and precision
System Architecture	Built on expert intuition and experience	Developed through simulations, trade studies, and analytical models
Validation and Verification	Testing is planned and executed after the full design is completed	Validation and verification activities run in parallel with ongoing development
Team Collaboration	Communication is mainly through static documents	Communication enhanced by shared, interactive models
Managing Changes	Tracking changes manually is difficult and can introduce errors	Model versions allow clear tracking and easier implementation of changes
Cross-Team Integration	Different teams often use disconnected tools and workflows	All disciplines work in a unified model environment
Issue Detection	Mistakes are often caught late in development, increasing rework	Potential issues are discovered early via modelling and simulations
Managing Complexity	Challenging to represent and manage large, complex systems	Hierarchical models simplify and organize complexity across the system
Lifecycle Integration	Weak connection between design, development, and testing phases	Strong linkage through models from concept to validation

Figure: Table comparing complex engineering systems before and after modelling with Model-Based Systems Engineering (MBSE).

6. Summary and conclusion

Our review proves Model-Based Systems Engineering to be a crucial methodology for managing complexity in modern aerospace systems. We found MBSE to be a crucial tool for mission planning, demonstrating its value for both private companies such as SpaceX and Blue Origin, and public organizations like NASA and ESA. This research critically evaluates how MBSE serves as a present-day engineering framework in dissimilarity to conventional document-centric systems in domains like the Space Commission and satellite systems. It included a comparative study of industry-leading tools, such as MagicDraw, IBM Rhapsody, Capella, and Simulink, with a comparative analysis between traditional

Paper ID: SR25817011922 DOI: https://dx.doi.org/10.21275/SR25817011922

systems engineering and MBSE, including a combination of Agile, MagicDraw, IBM, and Capella. It assessed the contribution of MBSE to fault detection, risk management, and maintainability in system design and described fundamental modelling techniques based on SysML [40].

Additionally, it demonstrates how the integration of deep learning networks, digital twins, using collaborative cloud platforms, and future SysML v2 can enhance the autonomy, structure, behavior, interdependence, teamwork, and project outcomes. Nonetheless, the drawbacks due to the steep learning curve, interoperability, and scalability need attention from stakeholders for wider integration of MBSE in the aerospace industry.

In the near future, research can be expanded to evaluate tool effectiveness and scalability of MBSE with the rapid technological advancements in active aerospace projects. MBSE's relevance and flexibility in Agile-integrated development processes, and interdisciplinary applications in robotics and smart manufacturing can be explored. Future research should focus on developing and validating hybrid methodologies that seamlessly integrate MBSE principles with agile development frameworks. New ways to adapt SysML and other MBSE tools to support iterative development cycles, rapid prototyping, and continuous feedback loops common in agile environments can be explored. Another critical area for future investigation is the establishment of more robust and standardized MBSE frameworks specifically tailored for complex aerospace missions.

To summarise, MBSE is an economical method, helping aerospace institutes to save time and money by increasing the success rate of missions and overall efficiency during the design and execution processes. Overall, specific to aerospace engineering, MBSE is near-omnipresent within this field.

Author contributions

A.P. writing - review & editing, writing original draft, visualization. P.S. writing - review & editing, writing original draft, visualization, data curation. S.A. writing - review & editing, formal analysis, supervision. S.K. writing - review & editing, validation, and conceptualization. S.S. writing - review & editing, writing original draft, Resources. U.S. writing - review & editing, writing original draft, data curation.

Competing financial interests

The authors declare no competing financial interests.

References

[1] *A Complexity Primer for Systems Engineers*, INCOSE, 2015.
[2] Pratt, M. and Dabkowski, M., 'Analyzing the Integration of MBSE Approaches within the Aerospace Industry according to UTAUT', *Industrial and Systems Engineering Review*, 10(2), 127–134, 2022.
[3] Shevchenko, N., 'An Introduction to Model-Based Systems Engineering (MBSE)', *SEI Blog*, 2025.
[4] *SE Handbook V4*, INCOSE, 2015.
[5] *Systems Engineering Vision 2020*, International Council on Systems Engineering (INCOSE), 2007.
[6] Estefan, J., *Survey of Model-Based Systems Engineering (MBSE) Methodologies*, 2008.
[7] Mann, K., *NASA Access to Space for All – Systems Engineering Webinar Series*, NASA, 2025.
[8] Graves, H. and Bijan, Y., 'Using Formal Methods with SysML in Aerospace Design and Engineering', *Annals of Mathematics and Artificial Intelligence*, 63(1), 53–102, 2011.
[9] IBM, *Engineering Rhapsody*, IBM Corp., 2025.
[10] Hill, T. R., Carnevale, A. E., Morris-Eckart, A., Sundaram, V. and Farhaj, L. Z., 'NASA's Use of MBSE and SysML Modeling to Architect the Future of Human Exploration', *INCOSE International Symposium*, 34(1), 527–543, 2024.
[11] Consultant, D., Chonoles, M., Gasster, S., Hart, L., Levi, A., Massa, C., Intercax, R. and Shiotani, B., *Developing and Distributing a CubeSat MBSE Reference Model – Status*, 2016.
[12] IBM, *Model-Based Systems Engineering*, IBM Corp., 2025.
[13] MathWorks, *Model-Based Systems Engineering for Space-Based Applications*, MathWorks, 2025.
[14] *Capella MBSE Tool – Arcadia*, Thales, 2025.
[15] Friedenthal, S., Moore, A. and Steiner, R., *A Practical Guide to SysML: The Systems Modeling Language*, Morgan Kaufmann, Waltham, MA, 2012.
[16] Kerschen, G., Worden, K., Vakakis, A. F. and Golinval, J.-C., 'Past, Present and Future of Nonlinear System Identification in Structural Dynamics', *Mechanical Systems and Signal Processing*, 20(3), 505–592, 2006.
[17] Kaslow, D., Soremekun, G., Kim, H. and Spangelo, S., *Integrated MBSE Applied to the Simulation of a CubeSat Mission*, ESA, 2015.
[18] Space Exploration Technologies Corp., *Falcon Launch Vehicle User's Guide*, 2021.
[19] Ecker, T., Karl, S., Dumont, E., Stappert, S. and Krause, D., 'A Numerical Study on the Thermal Loads during a Supersonic Rocket Retro-Propulsion Maneuver', *53rd AIAA/SAE/ASEE Joint Propulsion Conference*, 2017.
[20] *Space Engineering Machine Learning Handbook*, ECSS Secretariat ESA-ESTEC, 2024.
[21] NASA, *6.2 Requirements Management*, NASA, 2025.
[22] Tang, J., Zhu, S., Faudou, R. and Gauthier, J.-M., 'An MBSE Framework to Support Agile Functional Definition of an Avionics System', *Complex Systems Design & Management*, 168–178, 2018.
[23] Zhao, Z., Li, R. and Meng, T., 'An MBSE-Based Fault Modeling Approach for Satellite', *Journal of Physics: Conference Series*, 2977(1), 012077, 2025.
[24] Darpel, S., Beckman, S., Ferlin, T., Havenhill, M., Parrot, E. and Harcula, K., 'Method for Tracking and Communicating Aggregate Risk through MBSE Tools', *Journal of Space Safety Engineering*, 7(1), 11–17, 2020.
[25] Diatte, K., O'Halloran, B. M. and Van, D. L., 'The Integration of Reliability, Availability, and

Paper ID: SR25817011922 DOI: https://dx.doi.org/10.21275/SR25817011922

Maintainability into MBSE', *Systems*, 10(4), 101–101, 2022.
[26] Kaslow, D. C., Hart, L. M., Ayres, B., Massa, C., Chonoles, M., Yntema, R., Gasster, S. D. and Shiotani, B., 'Developing a CubeSat MBSE Reference Model – Interim Status', *IEEE Aerospace Conference*, 2015.
[27] Friedenthal, S., Moore, A. and Steiner, R., *A Practical Guide to SysML: The Systems Modeling Language*, Elsevier, Amsterdam, 2015.
[28] Holt, J., Perry, S. and Brownsword, M., *SysML Distilled*, IET Professional Applications of Computing Series 20, 2015.
[29] Hart, L. and Martin, L., *Introduction to MBSE and SysML*, INCOSE, 2015.
[30] Voltaire and Pomeau, R., *Voltaire Par Lui-Même*, Seuil, 1955.
[31] Fosse, E., 'From SysML to Mars: Mars 2020's MBSE Infusion', *NAFEMS Model-Based Engineering*, 2019.
[32] Wenyue, W., Junjie, H., Yinxuan, M., Jinjie, L. and Zhiang, L., 'Application and Development of MBSE in Aerospace', *Journal of Physics: Conference Series*, 2235(1), 012021, 2022.
[33] United States Department of Defense, *Information Memo for Deputy Secretary of Defense*, 2018.
[34] Kilicay-Ergin, N. and Dagli, C., 'Incentive-Based Negotiation Model for System of Systems Acquisition', *Systems Engineering*, 18(3), 310–321, 2015.
[35] *MBSE for Complex Engineering Projects*, 2020.
[36] Barbie, P., Pollom, A., Fischer, R.-P. and Becker, M., 'Automated Generation of Standardised Digital Twins Based on MBSE Models', *Proceedings of the 13th International Conference on Model-Based Software and Systems Engineering*, 75–84, 2025.
[37] Friedenthal, S., 'Future Directions for MBSE with SysML V2', *Proceedings of the 11th International Conference on Model-Based Software and Systems Engineering*, 5–9, 2023.
[38] Lopez, V. and Akundi, A., 'A Conceptual MBSE Approach to Develop Digital Twins', *IEEE International Systems Conference (SysCon)*, 2022.
[39] Holt, J., Perry, S. and Brownsword, M., *SysML for Systems Engineering*, IET Professional Applications of Computing Series, 2015.
[40] Gracias, M. and Gallegos, E., 'Transitioning Perspectives: Agile and Waterfall Perceptions in MBSE Integration within Aerospace and Defense Industries', *ITEA Journal of Test and Evaluation*, 45(4), 2024.

Author Profile

Astha Pandit is a gap year student who completed her A Levels from Budhanilkantha School. She is passionate about space science, aerospace engineering, and computer science. She loves to meet new people and share her ideas. Her hobbies include swimming, tennis, dancing, reading books, and exploring new things and places.

Priyangshu Sarkar is a high school senior from Springdale High School (H.S.). He is passionate about aerospace, especially about avionics, astronautical engineering, and space technologies. He enjoys playing guitar, making model jet RC planes for a hobby, and driving his car.

Sana Asrar is a second-year undergraduate Physics honours student at the Islamic University of Science and Technology, Awantipora, India. She is driven by her desire to follow her passion in Astrophysics, High-Energy Astrophysics, and Archaeology. Her hobbies include horse-riding, car drifting, and boxing.

Satvik Kodali is a Grade 12 Student of Sharanya Narayani International School (SNIS) who partakes in the IB Diploma Program. He is interested in aerospace and mechatronics. His hobbies include swimming, martial arts, and (recently) guitar.

Utsav Saroha is a 12th-grade student from Imperial Heritage School (IHS) in the CBSE board. He is a fan of space and the idea of finding extraterrestrial life (aliens). He is passionate about aerospace and astrophysics. He loves to explore the world, travel, stargaze, socialize, play chess, football, and enjoys spending time with animals.

Shaurya Singh, an 11th-grade student at St. Karen's High School, CBSE boon is passionate about space, its silence, vastness, and mysteries. Dreaming in rockets and equations, where Aerospace engineering builds the path and astrophysics lights the way. He sees the universe as a challenge and a puzzle. His hobbies include swimming and singing.