Proceedings of the 11th North American International Conference on Industrial Engineering and Operations Management, Milwaukee, Wisconsin, USA, June 9-11, 2026

Proceedings of the International Conference on Industrial Engineering and Operations Management

Publisher: IEOM Society International, USA DOI: 10.46254/NA11.20260220
Published: June 9, 2026

Reducing Small Satellite Mission Failures through Model-Based Systems Engineering: A SysML-Driven Case Study

Iqtiar Md Siddique
Assistant Professor
Department of Manufacturing and Systems Engineering
Weber State University, Ogden, Utah, USA
iqtiarsiddique@weber.edu

Abstract

Nearly 40% of small satellite missions fail, causing mission delays, financial loss, and reduced confidence in low-cost space systems (NASA 2024). Many of these failures are linked to integration gaps, fragmented documentation, weak traceability, and delayed validation across subsystem interfaces. Traditional document-based systems engineering methods are often difficult to maintain when requirements, architecture, behavior, interfaces, and verification evidence change during rapid satellite development. This paper presents a SysML-based Model-Based Systems Engineering approach for improving traceability, integration, and early validation in a representative small satellite system. The case study models the system architecture, internal interfaces, operational behavior, state transitions, requirements, and parametric constraints using Cameo Systems Modeler. The revised evaluation connects each SysML view to specific engineering outcomes, including requirement coverage, interface visibility, behavioral consistency, and validation support. The study shows how Block Definition Diagrams, Internal Block Diagrams, activity diagrams, state machines, sequence diagrams, parametric diagrams, and requirements tables can be used together to reduce ambiguity and support earlier detection of integration issues. The main contribution is a practical MBSE workflow for small satellite development that links system structure, behavior, requirements, and validation evidence within one model-based environment. The findings support the use of MBSE as a scalable method for improving lifecycle traceability and reducing integration risk in small satellite engineering.

Keywords
Model-Based Systems Engineering, SysML, Small Satellite Systems, Requirements Traceability, Systems Integration.

1. Introduction
Small satellite systems are increasingly used for Earth observation, communications, technology demonstration, scientific missions, and low-cost access to space. Their growth has been supported by modular hardware, shorter development cycles, and lower launch costs. However, these same advantages also create engineering challenges because small satellite projects often operate under strict limits in mass, volume, power, schedule, and budget. When subsystem interactions are not managed carefully, integration gaps can appear late in development and affect mission reliability.

Traditional document-based systems engineering methods remain common in many small satellite programs. Requirements, interface descriptions, operational logic, verification records, and design decisions are often stored in separate documents or discipline-specific tools. This approach can be workable for simple systems, but it becomes difficult to manage when power, communication, control, payload, thermal, and ground support functions must operate together. Fragmented documentation can reduce visibility across the system, weaken traceability, and delay the detection of interface conflicts.

Model-Based Systems Engineering provides a structured alternative by using connected system models instead of isolated documents. SysML supports the representation of system structure, behavior, requirements, interfaces, and constraints within a common modeling environment. In small satellite development, this approach can help engineering teams understand subsystem dependencies, maintain consistency between requirements and design elements, and support early validation before physical integration. These capabilities are important because many satellite failures are not caused by one component alone, but by incomplete understanding of how components interact across the mission lifecycle.

This paper presents a SysML-driven MBSE case study for a representative small satellite system. The study uses Cameo Systems Modeler to develop structural, behavioral, requirements, and parametric views of the system. The purpose is to show how these views can be connected to support lifecycle traceability, subsystem coordination, and early validation. In response to the need for stronger evidence, the revised paper also organizes the results around measurable model-based indicators, including requirement coverage, interface visibility, behavioral modeling coverage, and validation support. This provides a clearer basis for evaluating how MBSE contributes to reducing integration risk in small satellite system development.

1.1 Objectives

The objectives of this research are as follows:

1) To develop a SysML-based MBSE model for a representative small satellite system using structural, behavioral, requirements, and parametric views.
2) To demonstrate how MBSE improves traceability between requirements, system elements, interfaces, behaviors, and validation evidence.
3) To evaluate subsystem integration support through model-based representations of architecture, internal connections, operational flow, and state behavior.
4) To provide measurable support for the results by linking SysML diagrams to engineering outcomes such as requirement coverage, interface visibility, behavioral consistency, and validation support.
5) To present a practical MBSE workflow that can support small satellite development under cost, schedule, and integration constraints.

2. Literature Review

Short development cycles and increasing technical demands have exposed limitations in traditional systems engineering practices for small satellite missions. Many small satellite projects still rely on static documents, including requirement lists, interface control documents, verification records, and design descriptions. These artifacts are often maintained separately, which can create version inconsistencies, weak cross-referencing, and delayed identification of integration issues. Such limitations are important in CubeSat and small satellite programs because tight budgets, compressed schedules, and constrained technical margins reduce the opportunity for late redesign. NASA (2024) reported that satellite failures are often associated with integration and system-level issues rather than hardware alone. Honour (2010) also showed that disciplined systems engineering effort can improve project outcomes and reduce lifecycle risk.

Model-Based Systems Engineering has been widely discussed as an alternative to document-centered systems engineering. SysML provides a standard graphical language for representing system structure, behavior, requirements, interfaces, and constraints within a common model-based environment (Delligatti 2013). Estefan (2007) described MBSE methodologies as a way to support lifecycle consistency and structured system development. Fosse (2012) further emphasized that MBSE can improve communication, reduce artifact fragmentation, and support earlier system-level analysis. These studies support the use of MBSE when engineering teams must coordinate multiple design views and maintain traceability across development stages.

Prior aerospace and CubeSat studies demonstrate the practical value of MBSE in small satellite development. Spangelo et al. (2012) applied MBSE to a standard CubeSat and showed how SysML can support interface specification, requirements organization, and verification planning. Kaslow et al. (2015) developed a CubeSat MBSE reference model to support subsystem behavior representation, reuse, and mission modeling. Tübitak Uzay (2019) demonstrated full lifecycle SysML implementation for satellite missions, including interface logic and system constraints. Siddique (2024a) discussed the growing role of small satellites in space applications and emphasized the need for improved lifecycle coordination. These studies indicate that MBSE can improve the clarity of subsystem interactions, but they also show that model value depends on its connection to engineering evidence.

MBSE adoption also depends on process discipline, tool usability, and organizational readiness. Vaneman and Carlson (2019) stressed the importance of stakeholder engagement, governance, and implementation planning throughout the lifecycle. Morkevicius et al. (2017) identified scalability, interoperability, and toolchain integration as continuing challenges in complex system modeling. Henderson and Salado (2021) noted that modeling effort, tool learning curve, and usability concerns can limit early adoption benefits. INCOSE (2023) also indicated that MBSE benefits may require initial modeling investment before measurable improvements are observed. These findings suggest that MBSE should not be evaluated only by diagram production, but by its ability to improve traceability, integration visibility, validation support, and decision quality.

Recent work has continued to examine MBSE in relation to reliability, security, and complex system coordination. De Saqui-Sannes et al. (2022) classified MBSE approaches by language, method, and tool, showing that modeling choices must align with project context. Mažeika (2021) confirmed the value of structured system modeling for secure system design. Smith et al. (2024) used model-based architecture concepts to examine unreliable systems and fault propagation, showing the relevance of model structure for reliability analysis. Siddique (2024b) examined MBSE implementation in small satellite systems and identified traceability and system-level validation as important adoption drivers. Revell et al. (2023) also demonstrated that MBSE can support coordination in small-scale system development when modeling artifacts are connected to engineering decisions.

The literature shows that MBSE and SysML are useful for improving system representation, traceability, interface visibility, and early validation in aerospace and small satellite projects. However, many studies remain descriptive and do not provide clear quantitative indicators for comparing MBSE outcomes with traditional document-based practice. This gap is directly relevant to the present study. Therefore, this paper extends the discussion by linking SysML diagrams to measurable model-based indicators, including requirement coverage, interface visibility, behavioral coverage, and validation support. This structure provides a stronger basis for evaluating how MBSE can reduce integration risk in a representative small satellite system.

3. Methods

This study uses a SysML-based Model-Based Systems Engineering approach to model and evaluate a representative small satellite system. The method focuses on connecting system structure, interfaces, behavior, requirements, and validation evidence within a single model-based environment. Cameo Systems Modeler is used as the modeling tool to develop the SysML views and maintain consistency across the system model.

The method is based on three core MBSE elements: modeling language, modeling method, and modeling tool. SysML provides the modeling language for representing system components, interfaces, requirements, behaviors, and constraints. The modeling method guides the transition from stakeholder needs and system functions to architecture, behavior, requirements, and validation views. Cameo Systems Modeler provides the tool environment for model construction, traceability, and model-based analysis. These elements work together to support structured system development and lifecycle validation (Figure 1).

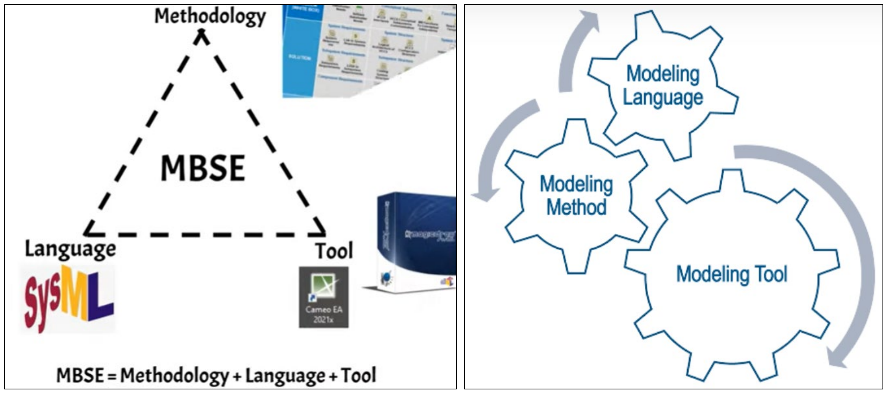

Figure 1. Core MBSE Pillars: Interdependence of Methodology, Language, and Tool.

The modeling process includes four main steps. First, the small satellite system boundary and major subsystems are defined. Second, structural models are developed to represent subsystem decomposition and internal interfaces. Third, behavioral models are developed to represent operational functions, system states, and component interactions. Fourth, requirements and parametric models are developed to support traceability and early validation. This process allows the system model to be evaluated beyond diagram presentation by linking each SysML view to specific engineering outcomes.

The model includes structural, behavioral, requirements, and parametric views. Structural modeling is used to define subsystem hierarchy and interface connections. Behavioral modeling is used to capture operational workflow, use cases, state transitions, and time-based interactions. Requirements modeling is used to connect stakeholder and system requirements to design elements through <<satisfy>> and <<verify>> relationships. Parametric modeling is used to evaluate system constraints, such as operating temperature limits, through model-based validation checks.

To address the need for stronger evaluation evidence, the method links the SysML model to measurable indicators. These indicators include requirement coverage, interface visibility, behavioral modeling coverage, and validation support. This provides a clearer basis for assessing how MBSE improves traceability, subsystem coordination, and early detection of integration issues in small satellite development.

4. Data Collection

The data for this study are collected from the representative small satellite SysML model developed in Cameo Systems Modeler. The model represents a small satellite system with major subsystems, including the main unit, communication subsystem, energy management subsystem, satellite control subsystem, imaging system, robotic mechanism, mission payload, data storage unit, and ground interaction elements.

The primary data sources are the SysML diagrams and model artifacts created during the case study. These include the Block Definition Diagram, Internal Block Diagram, Use Case Diagram, Activity Diagram, State Machine Diagram, Sequence Diagram, Parametric Diagram, Requirements Table, and Package Diagram. Each artifact provides a different type of engineering evidence. The Block Definition Diagram supports subsystem decomposition. The Internal Block Diagram supports interface visibility. The behavioral diagrams support operational and state-based analysis. The Requirements Table supports requirement traceability. The Parametric Diagram supports validation of system constraints.

Four model-based indicators are used to organize the collected data. Requirement coverage measures the extent to which requirements are captured and linked to system elements. Interface visibility measures how clearly subsystem interactions, power flows, data flows, and control paths are represented. Behavioral modeling coverage measures whether major operational functions, system states, and interaction sequences are represented. Validation support measures whether system constraints and requirement checks can be evaluated through model-based evidence.

The collected data are used to compare the MBSE approach with a traditional document-based approach. In a traditional approach, requirements, interfaces, behaviors, and validation evidence are usually stored in separate documents. In the MBSE approach, these elements are represented in connected model views. The numerical evaluation of these indicators is presented in Section 5.1, while the graphical results from the SysML model are presented in Section 5.2.

5. Results and Discussion

5.1 Numerical Results

The MBSE case study was evaluated using four model-based indicators: requirement coverage, interface visibility, behavioral modeling coverage, and validation support. A 1 to 10 scoring scale was used, where 1 indicates very limited support and 10 indicates strong model-based support with clear traceability or validation evidence. The traditional document-based approach was used as the baseline, while the MBSE-based approach was evaluated using the SysML model artifacts developed in Cameo Systems Modeler (Table 1).

Table 1. Comparison of traditional and MBSE-based small satellite development indicators.

Indicator	Traditional document-based score	MBSE-based score	Improvement (%)	MBSE evidence source
Requirement coverage	5	9	80.0	Requirements table with <<satisfy>> and <<verify>> relationships.
Interface visibility	4	9	125.0	Internal Block Diagram showing power, data, and control interfaces.
Behavioral modeling coverage	4	8	100.0	Use case, activity, state machine, and sequence diagrams
Validation support	3	8	166.7	Parametric diagram with satisfied and not satisfied constraint checks.
Average score	4.0	8.5	112.5	Combined model-based evidence across SysML views.

The requirement coverage score improved from 5 to 9, representing an 80.0 percent improvement, supported by the requirements table with <<satisfy>> and <<verify>> relationships. Interface visibility improved from 4 to 9, representing a 125.0 percent improvement, supported by the Internal Block Diagram showing power, data, and control interfaces. Behavioral modeling coverage improved from 4 to 8, representing a 100.0 percent improvement, supported by use case, activity, state machine, and sequence diagrams. Validation support improved from 3 to 8, representing a 166.7 percent improvement, supported by the parametric diagram with satisfied and not satisfied constraint checks. Overall, the average score increased from 4.0 to 8.5, representing a 112.5 percent improvement across the selected model-based indicators.

Table 1 shows that the MBSE-based approach improves all four evaluation indicators compared with the traditional document-based baseline. Requirement coverage increased from 5 to 9 because requirements were organized and linked to model elements through <<satisfy>> and <<verify>> relationships. Interface visibility increased from 4 to 9 because subsystem interactions, power paths, data paths, and control connections were represented in the Internal Block Diagram. Behavioral modeling coverage increased from 4 to 8 because the system functions, operational flow, states, and interaction sequences were captured through multiple SysML behavior views. Validation support increased from 3 to 8 because parametric modeling enabled early checking of operating constraints before physical integration.

The average score increased from 4.0 for the traditional approach to 8.5 for the MBSE-based approach, representing an overall improvement of 112.5 percent. This numerical comparison supports the claim that MBSE improves traceability, integration visibility, behavioral consistency, and early validation in the representative small satellite case study.

5.2 Graphical Results

The graphical results are based on the SysML model views developed for the representative small satellite system. These figures provide visual evidence of system decomposition, interface coordination, operational behavior, state logic, and early validation. Each figure supports a specific engineering purpose within the MBSE workflow.

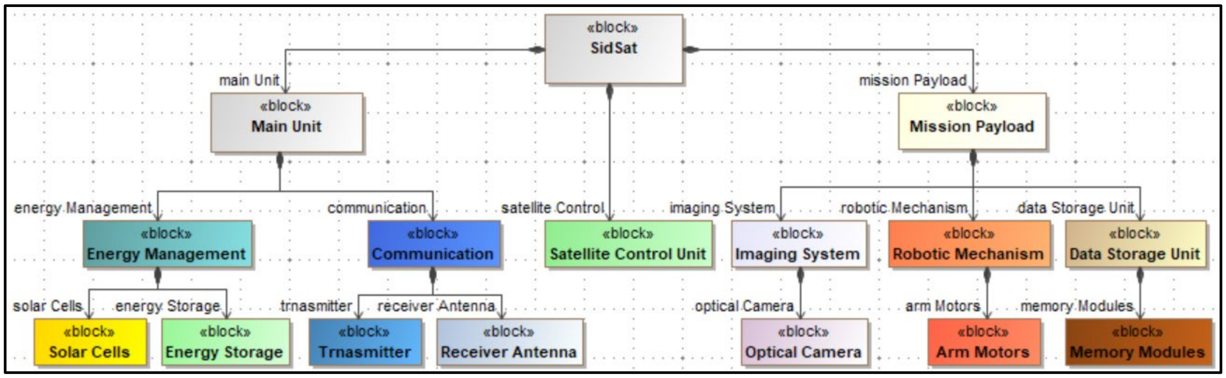

Figure 2. Small Satellite Block Definition Diagram (BDD).

Figure 2 presents the system decomposition of the small satellite. The diagram identifies the main satellite unit and its major subsystems, including communication, energy management, satellite control, imaging, robotic mechanism, mission payload, and data storage. This view supports modular architecture development by showing how system responsibilities are allocated across subsystems.

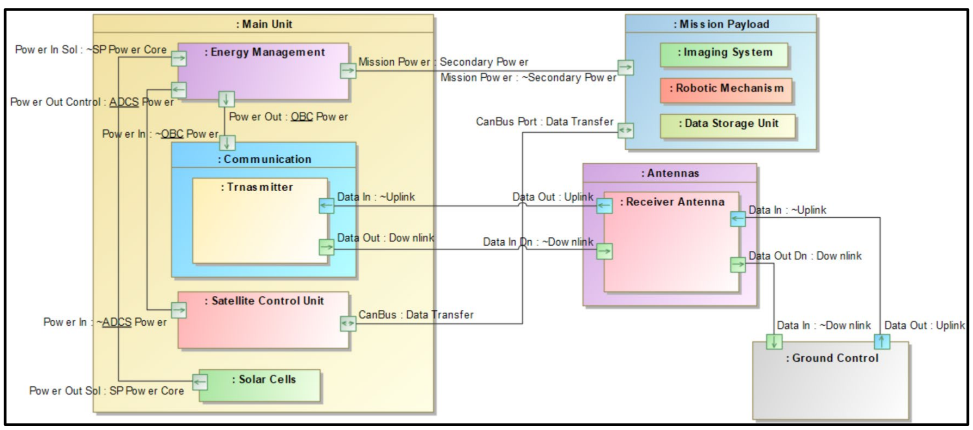

Figure 3. Small satellite internal block diagram.

Figure 3 shows the internal interfaces among the small satellite subsystems. The diagram represents power, data, and control interactions using ports and connectors. This view supports integration analysis because it makes subsystem dependencies visible and reduces the need to track interfaces only through separate documents.

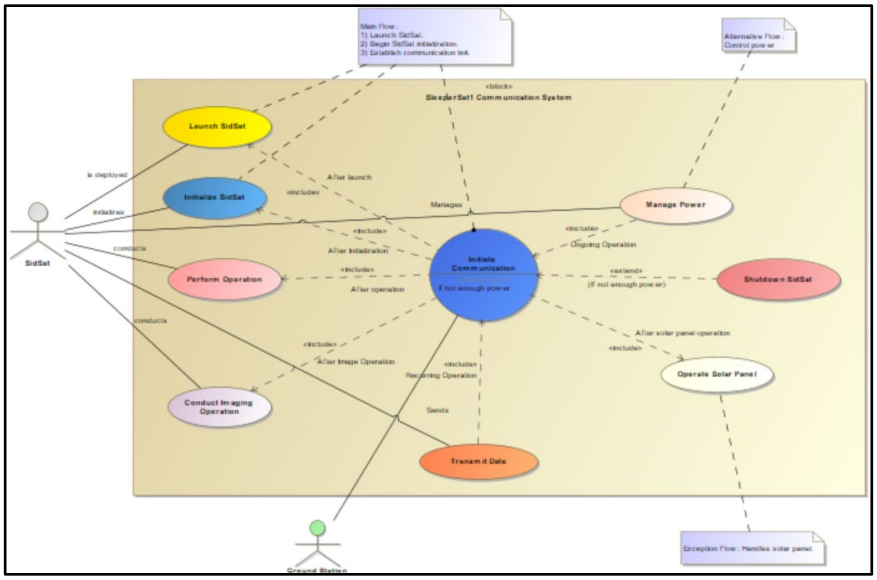

Figure 4. Small satellite use case diagram.

Figure 4 identifies the external actors and major system functions. The use case view defines how the satellite interacts with mission operators, ground station functions, and operational commands. This supports early understanding of system boundaries and links stakeholders need to functional behavior.

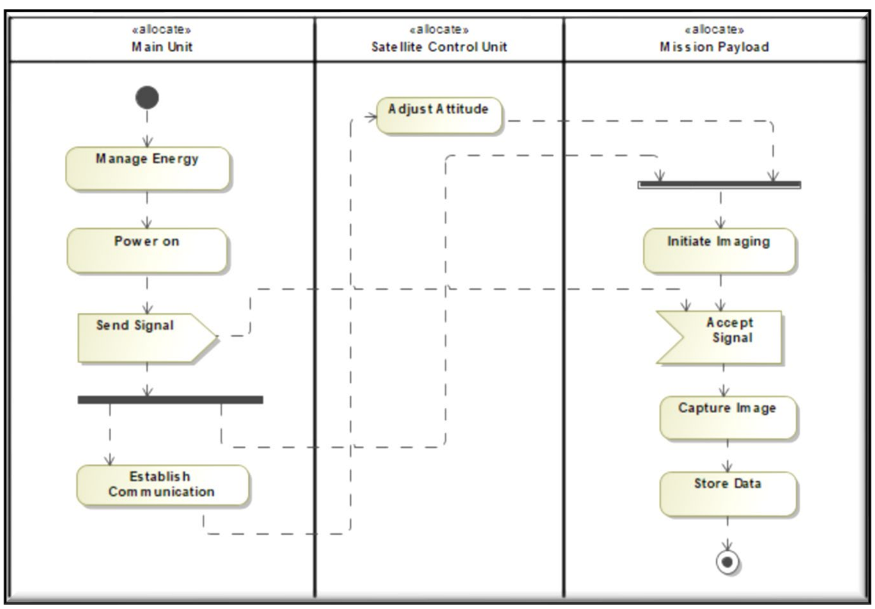

Figure 5. Small satellite activity diagram.

Figure 5 presents the operational workflow of the small satellite system. The activity view shows how mission tasks are sequenced and coordinated across system functions. This supports behavioral analysis by identifying the order of operations and the logical flow between mission activities.

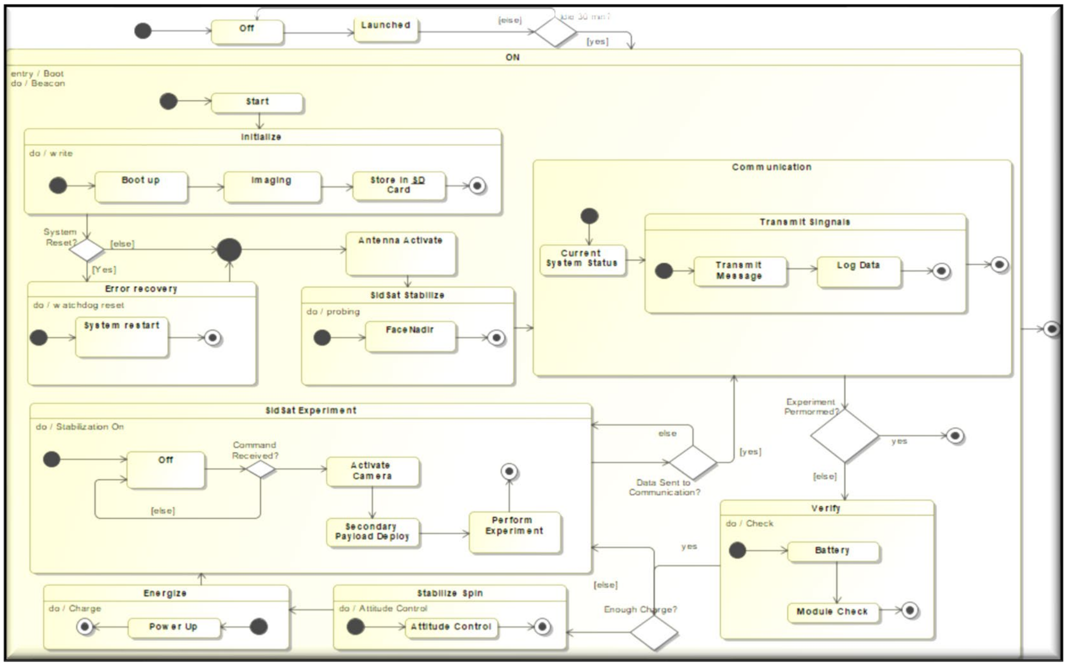

Figure 6. Small satellite state machine diagram.

Figure 6 shows the operational states and transitions of the small satellite. The state machine view supports analysis of system response under different operating conditions, including mode changes, command responses, and fault-related transitions. This helps verify that the system's behavior is consistent with mission logic.

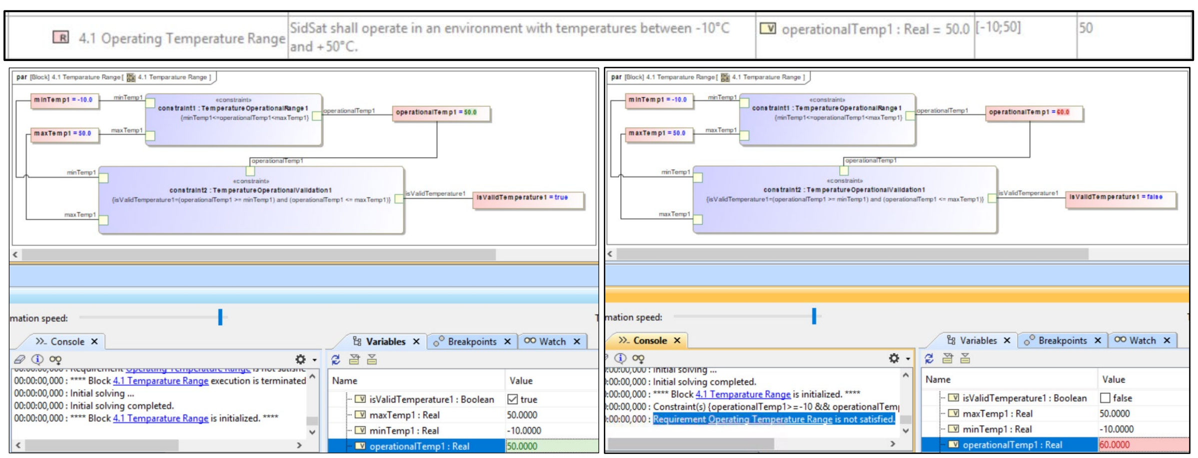

Figure 7. Operating temperature range validation using SysML parametric modeling.

Figure 7 presents the parametric validation view for operating temperature constraints. The diagram shows satisfied and not satisfied conditions, which demonstrates how model-based validation can identify design concerns before physical integration. This graphical result supports early verification by connecting system properties to constraint checks.

Together, the graphical results show that the SysML model provides more than visual documentation. The diagrams create connected engineering evidence across structure, interfaces, behavior, requirements, and validation. This supports the numerical results in Section 5.1 by showing how MBSE improves requirement coverage, interface visibility, behavioral modeling coverage, and validation support.

5.3 Proposed Improvements

The proposed MBSE workflow improves small satellite development by replacing isolated documents with connected SysML model views. The first improvement is stronger lifecycle traceability. Requirements are linked to system elements through <<satisfy>> and <<verify>> relationships, which helps maintain alignment between stakeholder needs, subsystem design, and validation evidence. This reduces the risk that requirement intent will be lost during design updates.

The second improvement is improved interface control. The Internal Block Diagram represents power, data, and control interactions among satellite subsystems. This improves visibility of subsystem dependencies and supports earlier identification of integration concerns. For small satellite projects with limited development time and constrained resources, early interface visibility is important because late integration issues can increase redesign effort and schedule risk.

The third improvement is stronger behavioral consistency. Use case, activity, state machine, and sequence views represent system functions, operational flow, states, and interaction timing. These views allow mission behavior to be reviewed before physical integration. This reduces ambiguity in how subsystems are expected to operate during mission execution.

The fourth improvement is early validation support. Parametric modeling allows selected constraints, such as operating temperature limits, to be checked within the model. This supports earlier detection of satisfied and not satisfied design conditions. For practical implementation, small satellite teams should begin with a focused MBSE scope that includes requirements, architecture, internal interfaces, critical behaviors, and key validation constraints. Additional model details can then be added as the design matures.

5.4 Validation

The validation in this study is based on model-based evidence from the representative small satellite case study. Because the system is not a flight mission, the validation does not claim flight performance verification. Instead, the validation evaluates whether the SysML model provides engineering evidence for traceability, integration visibility, behavioral consistency, and early constraint checking.

The requirements table supports traceability validation by showing that requirements can be organized and linked to model elements through <<satisfy>> and <<verify>> relationships. The Internal Block Diagram supports integration validation by representing power, data, and control interfaces among subsystems. The behavioral diagrams support operational validation by showing system functions, workflow, state transitions, and interaction sequences. The parametric diagram supports constraint validation by checking operating temperature conditions and identifying satisfied and not satisfied cases.

The numerical results in Section 5.1 provide additional validation support. The average indicator score increased from 4.0 for the traditional document-based approach to 8.5 for the MBSE-based approach. This represents a 112.5 percent improvement across requirement coverage, interface visibility, behavioral modeling coverage, and validation support. These results show that the MBSE model provides measurable improvement in the areas most directly linked to small satellite integration risk.

This validation is preliminary but appropriate for a representative case study. It confirms that the proposed MBSE workflow provides structured evidence beyond diagram presentation. Future validation should apply the workflow to an active satellite project, compare model-based indicators against project records, and include measured outcomes such as defect discovery time, interface issue count, requirement verification status, and rework reduction.

6. Conclusion

This study presented a SysML-based Model-Based Systems Engineering workflow for improving traceability, visibility, behavioral consistency, and early validation in a representative small satellite system. The research addressed the limitations of traditional document-based engineering practice, where requirements, interfaces, behaviors, and validation evidence are often maintained separately. These limitations can delay the identification of integration issues and reduce confidence in system-level coordination during small satellite development.

The research objectives were met by developing a connected SysML model using structural, behavioral, requirements, and parametric views. The Block Definition Diagram and Internal Block Diagram supported subsystem decomposition and interface visibility. The use case, activity, state machine, and sequence views supported behavioral analysis and operational consistency. The requirements table supported traceability through <<satisfy>> and <<verify>> relationships, while the parametric model supported early validation of operating constraints.

The main contribution of this paper is a practical MBSE workflow that links small satellite system architecture, interfaces, behavior, requirements, and validation evidence within a single model-based environment. The numerical comparison showed that the average indicator score improved from 4.0 for the traditional document-based approach to 8.5 for the MBSE-based approach, representing a 112.5 percent improvement across the selected model-based indicators. This provides stronger support for the claim that MBSE can reduce integration risk and improve lifecycle coordination in small satellite engineering.

The findings are based on a representative system model, so they should be interpreted as early-stage model-based validation rather than flight-level validation. Future work should apply the workflow to an active satellite project, collect project-level metrics, and compare MBSE outcomes with actual development records, including interface issue count, requirement verification status, defect discovery time, and rework reduction.

Acknowledgments
The author gratefully acknowledges the College of Engineering, Applied Science & Technology at Weber State University providing financial support for conference participation, and thanks Dr. Brian W. Rague and Nicole Batty-Falkenberg for their guidance.

References
De Saqui-Sannes, P., Vingerhoeds, R. A., Garion, C. and Thirioux, X., A taxonomy of MBSE approaches by languages, tools, and methods, *IEEE Access*, vol. 10, pp. 120936-120950, 2022, https://ieeexplore.ieee.org/document/9950507

Delligatti, L., SysML Distilled: A Brief Guide to the Systems Modeling Language, *Addison-Wesley*, 2013, https://books.google.com/books?hl=en&id=3bMJAgAAQBAJ

Estefan, J. A., Survey of model-based systems engineering methodologies, *INCOSE MBSE Focus Group*, vol. 25, no. 8, pp. 1-12, 2007, https://www.omgsysml.org/MBSE_Methodology_Survey_RevB.pdf

Fosse, E., Model-based systems engineering 101, *NASA*, 2012, https://ntrs.nasa.gov/citations/20150007154

Henderson, K. and Salado, A., Value and benefits of model-based systems engineering: Evidence from the literature, *Systems Engineering*, vol. 24, no. 1, pp. 51-66, 2021, https://doi.org/10.1002/sys.21566

Honour, E., Systems engineering return on investment, *INCOSE International Symposium*, vol. 20, no. 1, pp. 1422-1439, 2010, https://doi.org/10.1002/j.2334-5837.2010.tb01150.x

INCOSE, INCOSE Systems Engineering Handbook, *John Wiley & Sons*, 2023, https://www.incose.org/products-and-publications/se-handbook

Kaslow, D., Anderson, L., Asundi, S., Ayres, B., Iwata, C., Shiotani, B. and Thompson, R., Developing a CubeSat model-based systems engineering reference model: Interim status, *Proceedings of the 2015 IEEE Aerospace Conference*, pp. 1-16, 2015, https://doi.org/10.1109/AERO.2015.7118965

Mažeika, D., Model-based systems engineering method for creating secure systems, Doctoral dissertation, *Kauno technologijos universitetas*, 2021, https://epubl.ktu.edu/object/elaba:100405770/

Morkevicius, A., Aleksandraviciene, A., Mazeika, D., Bisikirskiene, L. and Strolia, Z., MBSE grid: A simplified SysML-based approach for modeling complex systems, *INCOSE International Symposium*, vol. 27, no. 1, pp. 136-150, 2017, https://doi.org/10.1002/j.2334-5837.2017.00350.x

NASA, Why satellites fail? It's not only the hardware, *Space Agency*, 2024, Available: https://spaceagency.prowly.com/404102-why-satellites-fail-its-not-only-the-hardware

Revell, D., Gross, D. C., Zhou, G. and Hernandez, A., Applying a MBSE methodology in small scale technology development, *Proceedings of the 2023 IEEE International Systems Conference*, pp. 1-6, 2023, https://ieeexplore.ieee.org/document/10131099

Siddique, I., Small satellites: Revolutionizing space exploration and Earth observation, *European Journal of Advances in Engineering and Technology*, vol. 11, no. 3, pp. 118-124, 2024, https://doi.org/10.5281/zenodo.12750228

Siddique, I., MBSE implementation in small satellite systems: Rationale for adoption over traditional document-based systems engineering, *International Journal of Global Science and Technology*, vol. 1, no. 1, 2025, https://doi.org/10.46610/IJGST.2025.v01i01.001

Smith, E., Siddique, I., Kreinovich, V. and Posthuma, R., Hierarchical model of unreliable systems in comparison to an anti-fragile replacement system, *Proceedings of the IISE Annual Conference*, pp. 1-6, Institute of Industrial and Systems Engineers, 2024.

Spangelo, S. C., Kaslow, D., Delp, C., Cole, B., Anderson, L., Fosse, E. and Cutler, J., Applying model-based systems engineering to a standard CubeSat, *Proceedings of the 2012 IEEE Aerospace Conference*, pp. 1-20, 2012, https://doi.org/10.1109/AERO.2012.6187339

Tübitak Uzay, Application of model-based systems engineering with SysML in a small satellite project, *Proceedings of the 9th Ankara International Aerospace Conference*, AIAC-2019-187, 2019, https://www.researchgate.net/publication/336130617

Vaneman, W. K. and Carlson, R., Model-based systems engineering implementation considerations, *Proceedings of the 2019 IEEE International Systems Conference*, pp. 1-6, 2019, https://doi.org/10.1109/SYSCON.2019.8836888

Biography

Iqtiar Md Siddique is an Assistant Professor in the Department of Manufacturing and Systems Engineering at Weber State University. He received a Ph.D. in Mechanical Engineering and an M.S. in Systems Engineering from The University of Texas at El Paso, an M.Eng. in Computer Engineering from RMIT University, and a B.S. in Computer Science from North South University. His research focuses on systems engineering lifecycle processes, requirements engineering, Model-Based Systems Engineering, digital engineering, aerospace systems, complex adaptive systems, and AI-enabled engineering methods. His doctoral research examined MBSE for aerospace applications, with emphasis on requirements prioritization, lifecycle traceability, digital integration, and small satellite development. Dr. Siddique has served as a Lecturer, INCOSE Academic Equivalency Consultant, Doctoral Research Associate, and Graduate Teaching Assistant at The University of Texas at El Paso. His academic and technical work includes systems modeling, requirements analysis, SysML, Cameo Systems Modeler, MATLAB, STK, Teamcenter, and model-based verification activities. He is a Certified Systems Engineering Professional, SysML Model User, Certified Professional in Requirements Engineering, and Lean Six Sigma Green Belt.