INCOSE

ASIA-OCEANIA SYSTEMS ENGINEERING CONFERENCE

Model-Based Design Methodology for Vehicle Body Domain Systems

A Shift-Left Approach Using MBSE

INCOSE AOSEC · 2025

AUTHORS

Xia Yi, Jian Qiong Zhang, Feng Li, Ling Yan Wu, Ho Kit Robert Ong, and Guanyong Wang

PUBLICATION DATE

July, 2026

SUBMISSION #

31

INCOSE PUBLICATIONS

Model-Based Design Methodology for Vehicle Body Domain System: A Shift-Left Approach Using MBSE

YI Xia
SAIC Motor Co., Ltd. Technical Center
yixia@saicmotor.com

ZHANG Jianqiong
SAIC Motor Co., Ltd. Technical Center
zhangjianqiong@saicmotor.com

LI Feng
SAIC Motor Co., Ltd. Technical Center
lifeng05@saicmotor.com

WU Ling Yan
Dassault Systèmes
daphne.wu@3ds.com

ONG Ho Kit Robert
Dassault Systèmes
hokitrobert.ong@3ds.com

WANG Guanyong
Dassault Systèmes
Guanyong.Wang@3ds.com

Abstract

The rapid evolution of Software-Defined Vehicles (SDVs) is driving transformative trends, including Over-the-Air (OTA) updates, enhanced connectivity, decoupled hardware-software platforms, vehicle autonomy, personalized user experiences, and enhanced safety. These trends are reshaping how engineers design a vehicle's electrical/electronic (E/E) architecture. Traditional document-based systems engineering struggles to meet the agility and complexity demands of modern E/E development under tight time-to-market pressures. This paper presents an agile, "shift-left" Model-Based Systems Engineering (MBSE) methodology integrated into existing engineering processes to support advanced E/E architectures for SDVs. In this approach, modeling and simulation like upstream requirements and system architecture with downstream development enable early analysis and validation of designs. Key benefits of the approach include: (1) improved clarity in requirements and interface definitions; (2) earlier design validation through the shift-left agile process; (3) support for scenario-driven design and functional decomposition, enabling data-validated E/E architecture optimizations; and (4) improved cross-disciplinary collaboration.

Keywords

MBSE, E/E architecture, SDV, SysML, body domain systems, model-based design, shift-left engineering.

Introduction

Software-Defined Vehicles (SDVs) place software at the core of vehicle functionality, enabling continuous upgrades, autonomy, adaptability, personalized user experiences. As an industry example, SAIC

Motor has introduced a comprehensive "1-3-5" intelligent connectivity strategy, that integrates a centralized E/E architecture, a service-oriented architecture (SOA) software platform, a vehicle data cloud factory, Over-the-Air (OTA) updates capability, and cybersecurity measures. The objective is to deliver an end-to-end scenario-based service platform that can evolve over the vehicle's lifetime (SAIC, 2021). Achieving this vision requires the transition from traditional distributed E/E architectures to centralized or zonal architectures. These advanced architectures incorporate high-performance computing units and standardized software frameworks to support complex software-driven functions (Vignesh et al., 2023). They also emphasize modular, scalable hardware design (Ayres et al., 2024) and greater decoupling of hardware and software, enabling dynamic software updates and subscription (Kirchner et al., 2025). To ensure safety and security in such a connected environment, robust in-vehicle communication protocols and compliance with automotive cybersecurity standards are critical (Toghuj & Turab, 2023).

Despite the potential benefits, these technological shifts pose significant engineering challenges. Traditional document-based systems engineering processes cannot efficiently handle the increased complexity and fast iteration cycles required for SDV development (Askaripoor, H., 2022; Jiang, S., et al., 2024). Validating designs earlier (i.e. shift-left) in the lifecycle is crucial, as late-stage issues in such complex systems are extremely costly to fix. To address these challenges, this paper proposes an approach that combines Agile systems engineering principles with Model-Based Systems Engineering (MBSE).

Literature Review

Challenges in Transitioning to Advanced E/E Architectures: The introduction of Software-Defined Vehicle (SDV) concepts and centralized E/E architectures in automotive systems engineering is facing several key challenges.

First of all, transitioning from traditional distributed architectures to centralized or zonal architectures requires a comprehensive restructuring of vehicle hardware and software. The shift to a centralized computing model entails reassigning functions from many discrete Electronic Control Units (ECUs) to a few powerful domain or zonal controllers (Mauser & Wagner, 2024). Employing Model-Based Systems Engineering (MBSE) with simulation capabilities can enhance the efficiency and accuracy of this architectural redesign by enabling early evaluation of design decisions under various scenarios.

Second, achieving hardware modularity and scalability poses a challenge. Designing a plug-and-play scalable hardware component demands a highly modular architecture, which significantly increases system complexity. Managing numerous interchangeable modules (sensors, actuators, controllers) and their interconnections can impact vehicle reliability and safety if not carefully engineered. This is especially critical in high-performance computing domains and real-time control systems, where added complexity may introduce new failure modes (Wen et al., al., 2024; Shetiya et al., 2024).

Third, managing software complexity and enabling service-oriented architecture (SOA) has become paramount. In an intelligent connected mobility setting, often described as a vehicle-road-cloud ecosystem; software services play an increasingly central role (Ding et al., 2022; CAICT, 2024). Mobility functions are being decomposed into independent service units of *apps* that can be developed, tested, and updated separately, which present challenges in maintaining compatibility and interoperability across numerous suppliers and platforms.

Fourth, ensuring functional safety and cybersecurity in SDVs is more challenging than ever. Modern vehicles exchange data with the cloud and infrastructure, which exposes them to cyber threats. Simultaneously, safety-critical functions (Steering, braking, etc.) rely on complex software that must be fail-safe. Guaranteeing both safety (ISO26262:2018; ISO21448:2022) and security (ISO/SAE 21434:2021) demands compliance and requires substantial efforts.

To tackle the above challenges, systems engineering (SE) provides an overarching disciplined approach (INCOSE, 2023). However, traditional systems engineering guidelines do not prescribe a specific life cycle model tailored to SDVs and new E/E architectures. Recent work suggests that combining agile principles with MBSE can facilitate the digital transformation needed for SDV development (Ågren et al., 2022).

Methodology

To address the challenges identified, this paper proposes a tailored Agile MBSE methodology based on the MagicGrid framework (Dassault Systèmes, 2021) for the E/E design process. A four-abstraction level architecture was adopted: Requirements, Functional, Logical, and Physical or RFLP (Figure 1). The methodology emphasizes continuous validation (Shift-left) and cross-domain traceability throughout these layers.

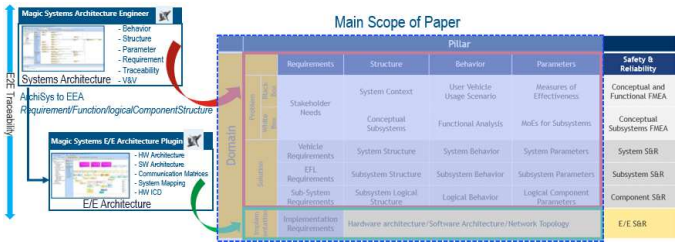

Figure 1. Proposed Methodology and Main Scope of the Paper

This paper implements an approach using a SysML-compliant and ISO 26262 certified modeling tool (CATIA Magic Cyber Systems Engineer from Dassault Systèmes) to construct the system model, with its built-in simulation capability (Magic Model Analyst and Magic Systems EE Architect), support analysis, and Model-in-the-Loop (MiL) and electrical architecture mapping capabilities. It is important to note that the approach is tool-agnostic; any modeling environment compliance to SysML and supports simulation could be used, as the focus is on the methodology rather than specific tool features. The case study in this paper is a simplified Vehicle Body Domain (VBD) system. The VBD includes features like windows, locks, and lighting systems, which are integrated and software controlled. This paper illustrates how the methodology links user-level features to E/E architecture design. The methodology comprises a six-step process, preceded by a preliminary setup step. Each step in the RFLP continuum entails a different abstraction layer. The following describes the six-step process methodology.

Step 0 Preliminary Setup: Before modeling the system, a consistent framework was established to define terminology and relationships specific to the automotive industry. The engineers at SAIC Motor developed a SysML profile called "BodyDomainProfile" that extends standard SysML elements for domain-specific concepts with stereotypes (Figure 2). This profile ensures that the model elements correspond to real-word concepts used by the engineers at SAIC, improving clarity and reuse. Step 0 also involves setting up templates and guidelines for model organization (naming conventions, package structure) so that all subsequent modeling activities were done in a consistent manner.

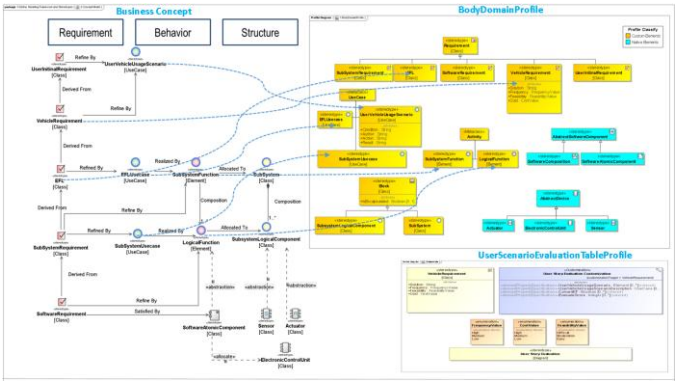

Figure 2. Body Domain Profile

Step 1 Requirements Level: The initial step was to analyze the user and vehicle usage scenarios (Figure 3). System use cases were captured in a textual form specifying Actor, Condition, Action, and Result for each scenario. From these, the engineers derived an initial Electrical Feature List (EFL), essentially a set of candidate electrical features the vehicle should support. Each scenario was evaluated using a multi-criteria matrix (considering factors like cost, feasibility, and frequency of use). Scenarios that scored above a threshold (for example, 6/10) were promoted to official features in the EFL (Figure 4). These features were then modeled as EFL requirements in the SysML model. For instance, one use case scenario might be *Automatically close windows*

upon detecting rain. This scenario would be evaluated and, if deemed important, translated into an electrical feature requirement for automatic rain-sensing windows closure. This ensures that the design started with a clear understanding of what the user and stakeholders expected the system to do. The outcome was a set of validated feature requirements, each traced back to one or more usage scenarios and ready to be imposed functionally.

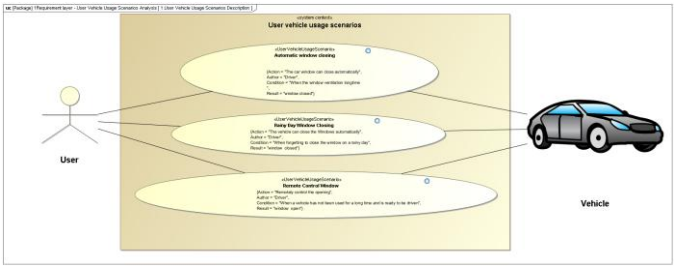

Figure 3. Use Cases of User Vehicle Usage Scenarios

Figure 4. User Vehicle Usage Scenario Evaluation Form

Step 2 Functional Level: The engineers refined the high-level feature requirements into detailed functional requirements and allocated them to subsystems. Using the EFL from Step 1, the subsystems (or domain) of the vehicle that would deliver each feature were identified. For the body domain case, typical subsystems include the doors, windows, lighting, climate, etc. The use case diagram modeled at the subsystem level shows how external actors (drivers, environment sensors, etc.) interact with each subsystem (Figure 5).

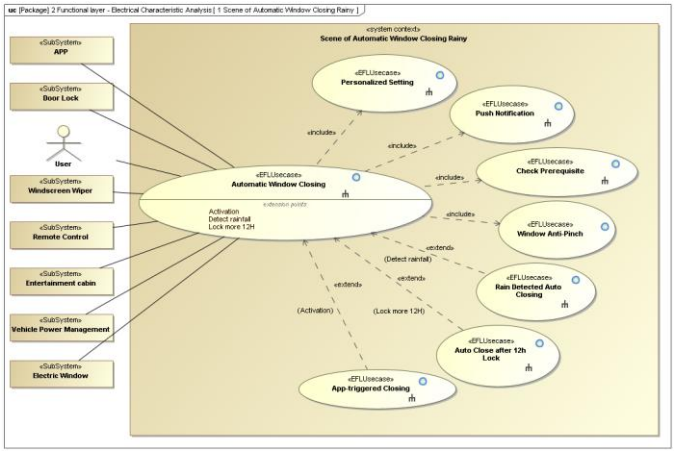

Figure 5. Use Case of Automatic Window Closing during Rain

Next, the engineers specified the functional behavior for each feature via scenarios and flows. A template defining Trigger, Preconditions, Basic Flow, Alternative Flows, and Postconditions was used to ensure completeness of each functional requirement. For example, the constituent functions of the automatic windows closing during rain feature are Rain Detection, Windows Control, User Notification, and Anti-pinch Protection. An activity diagram was modeled to illustrate how these functions interact (Figure 6). These functional elements were allocated to specific subsystems through a swimlane.

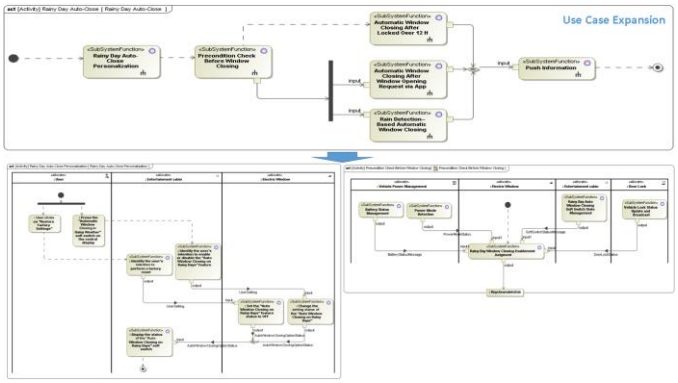

Figure 6. Use Case Analysis of Automatic Window Closing during Rain

At the end of this step, a set of functions were grouped into subsystems with well-defined interfaces and interaction logic (Figure 7). Every feature requirement was thus realized by a network of functions, and with automated traceability, each function was connected back to the originating feature

and usage scenario. The mapping relationships among the subsystems, subsystem functions, and subsystem requirements were managed in the subsystem functions and requirements table (Figure 8).

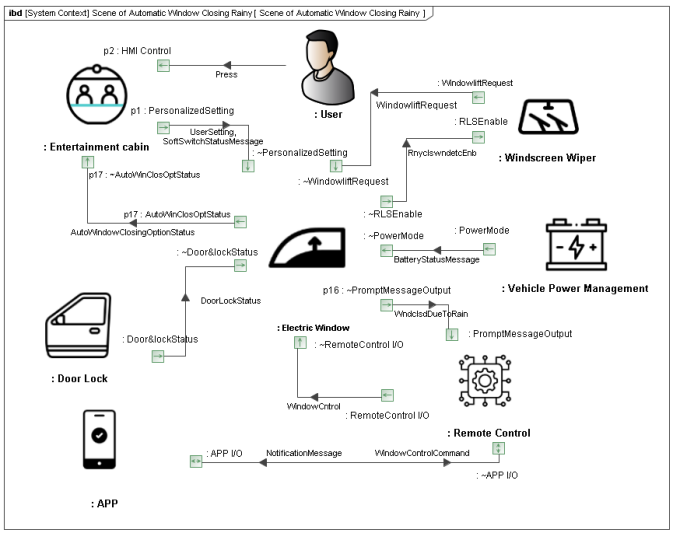

Figure 7. Subsystem Composition Structure Diagram

Figure 8. Subsystem Functions and Requirements Table

Step 3 Logical Level: This step transformed the functional view into a logical design that would later map to actual hardware and software components. The logical architecture is a network of logical components (blocks) with well-defined interfaces and item flows modeled in SysML. Each logical component can be thought of as an abstract subsystem or software component that would implement one or more functions from Step 2. The functional elements were allocated to the logical components and communication between these components was established (using internal block diagrams and ports/flows) to mirror the functional flows (Figure 9). State machines were used to model the key components behavior (Figure 10).

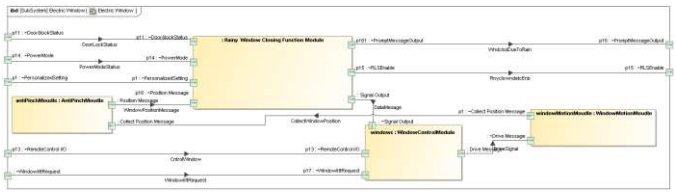

Figure 9. Internal Logical Structure of the Power Window System

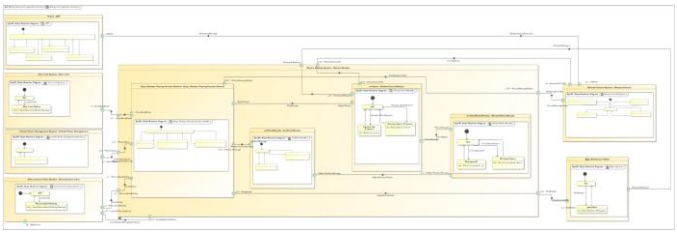

Figure 10. Component State Machine Diagram

The logical-to-software mapping is a crucial step to map each logical component to a software component that will exist in the implementation. For instance, Window Controller Logic might map to a specific AUTOSAR software component in the body control ECU software. This mapping was captured in the model, linking the logical view to the software requirements (Figure 11).

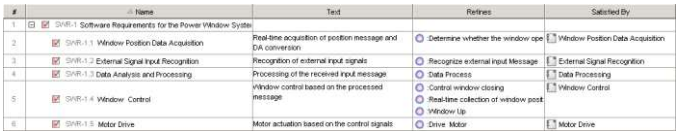

Figure 11. Software Requirements and SW Component Definition

At the end of Step 3, the logical architecture serves as a blueprint that guides software developers to identify what components and interfaces are needed to fulfill the requirements was created. Elements traceability was maintained, enabling the engineers to track and trace a software component in the model back to the logical function it implements, the feature it supports, and ultimately the originating scenario.

Step 4 Physical Level: This step refers to the process of defining a network of electronic control units (ECUs), sensors, actuators, and communication buses that would realize the logical components. This involves mapping the logical architecture to the hardware E/E architecture (Figure 12) (the

physical topology of the vehicle's electronics). Finally, a hardware ICD was generated automatically.

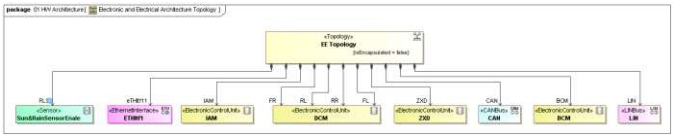

Figure 12. E/E Topology Structure

In the model, the logical components from Step 3 were allocated to hardware nodes. This includes specifying software components architecture (Figure 13), selecting and defining the E/E hardware topology through the Physical Component Library, their communication links (CAN, Ethernet, LIN, etc.), and the signals exchanged (Figure 14) such as RainDetected, CloseWindowCommand, WindowPosition, etc., with their senders, receivers, and bus properties. The model now contains a detailed hardware diagram showing, for instance, a central Body Controller connected via a CAN bus to Door Zone Controllers, with sensors and actuators attached to these controllers. The software components were then allocated to these hardware elements (Figure 15), completing requirements-to-software-to-hardware mapping.

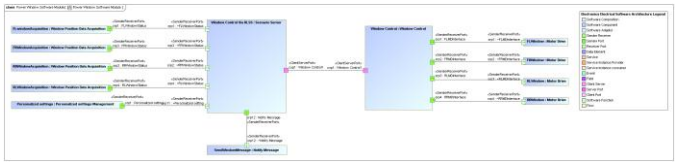

Figure 13. Software Component Architecture

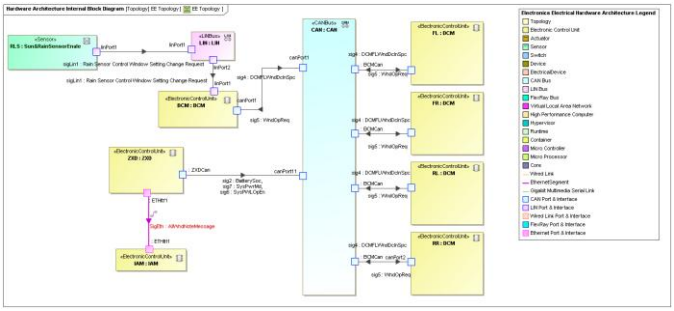

Figure 14. Network Topology of Physical Components

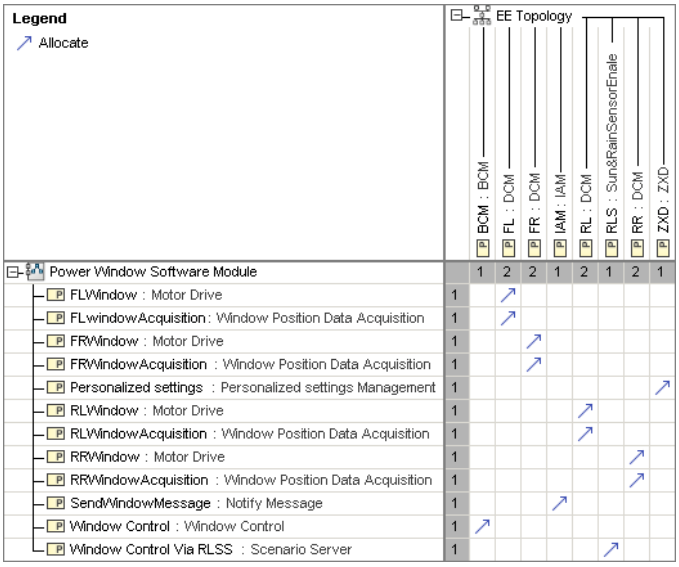

Figure 15. Software Components Allocated to E/E Hardware Components

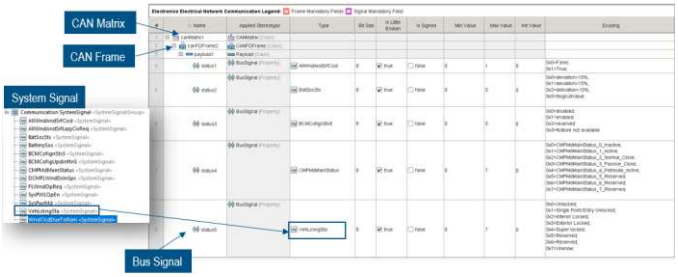

Figure 1. Communication Matrix Table

A communication matrix table is a structured table used to describe the communication behavior and message exchange between nodes (ECUs - Electronic Control Units) in a network based on protocols. The engineers focused on what message would be sent or received on the communication bus, defined frame, payload, bus signal, frame length, payload length, bus signal bit size, signal minimum, maximum, and encoding (Figure 16). The frame ID and bit Start were further defined by the downstream software application development team. The team would also further analyze computing capacity, network bandwidth, and safety isolation.

The software architecture, hardware topology, and communication matrix are amendable to consistent synthesis, and they could be validated by the hardware ICD tables (Figure 17) generated automatically from the authoritative E/E architecture model and passed downstream for the AUTOSAR-based

software development to ensure they are synchronized with design changes.

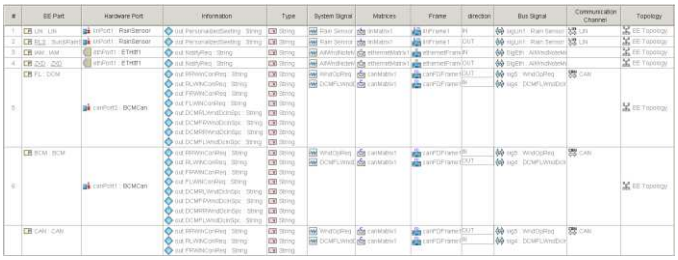

Figure 17. Hardware Interface Control Document (ICD)

Step 5 Simulation and Early Validation: The core principle of the shift-left approach is to perform dynamic validation of the design as early as possible in the software development lifecycle to improve quality. The SysML tool allowed the engineers to perform validation on an integrated architecture as the verification tool to configure and execute simulations, and run MiL simulations directly on the logical/physical model. Test scenarios corresponding to the use cases (from Step 1) were executed against the model to validate its behavior and identify potential issues. In the context of windows closing during rain simulation, a test scenario might simulate a rain event and verify that it would trigger the windows to close and that no pinch was detected (Figure 18).

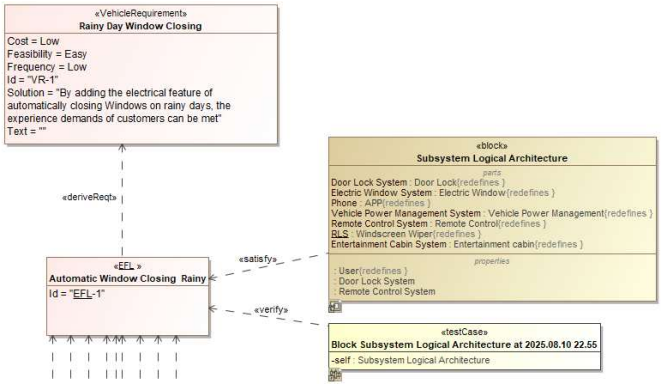

Figure 18. Automatic Window Closing during Rain Test Case

During simulation, the engineers could observe signal transitions and state changes in the state machines. The tool captured these behaviors in sequence diagrams or time series, which enabled the engineers to analyze the system's behavior under the simulated conditions (Figure 19). The engineers also leveraged automated test case generation and execution for regression testing every time the model is updated, rerun a suite of scenarios to ensure changes to the model do not impact its functionality.

This continuous verification practice embodied the shift-left principle, enabling the engineers to address potential design flaws or requirement mismatches in the model simulation, well before performing integration testing on actual vehicles. In summary, Step 5 provides confidence that the integrated requirements, logical, and physical design would fulfill the intended functions under various conditions. As noted in industry reports, moving such validation to earlier phases can shorten development cycles and improve design quality (Heyman, 2024).

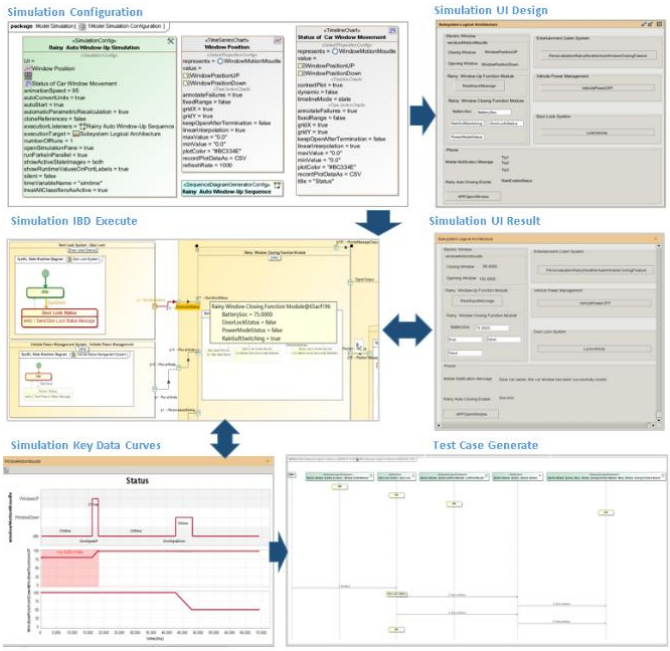

Figure 19. Simulation Configuration Design and Simulation Results

Step 6 Traceability and Impact Analysis: In the final step of the MBSE methodology, the engineers formalized traceability built into the model and used it to perform impact analysis and change management, which proves extremely beneficial for evaluating changes as it will highlight which parts of the design are affected if a certain sensor specification changes or a new requirement is

added. It allows the engineers to quickly identify all functions and components related to that sensor or requirement. This level of traceability and fast impact analysis (Figure 20) greatly improve change management efficiency; any proposed change can be assessed in the model for ripple effects before implementing it in a real system.

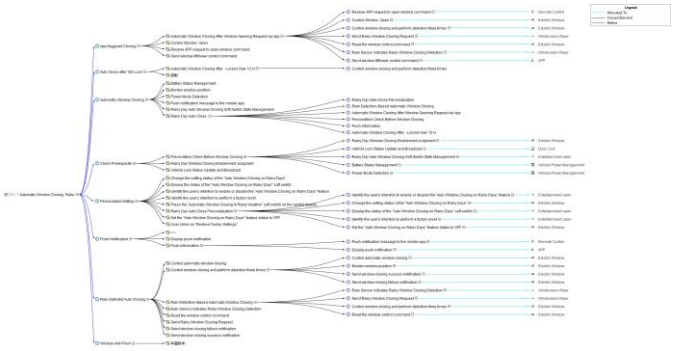

Figure 20. EFL to EFLUseCase to SubsystemFunction to Subsystem traceability

Results and Future Work

The proposed MBSE methodology enables the forward engineering (Zhu, X., 2025) of the VBD system. This approach provides a structured yet flexible process, from capturing stakeholder scenarios to verifying the system design using simulation early in the lifecycle with end-to-end traceability, all within a unified MBSE environment.

The inclusion of a VBD case study illustrated how each step was applied in a real-world context. The benefits include (1) improved clarity in requirements and interface definitions; (2) earlier design validation through shift-left agile process with MiL to verify that the design aligned with the intended functional goals (El Sallab et al., 2021; Mou et al., 2023); (3) support for scenario-driven design and functional decomposition, enabling data-validated E/E architecture optimizations; and (4) improved cross-disciplinary collaboration and reduced time spent on downstream integration. In summary, the structured modeling of requirements, behavior, and interfaces enabled one-click tracing of functional impact scopes, facilitating rapid change impact analysis and issue localization and enhanced communication across teams and supported compliance with industry standards such as ISO 26262 and Automotive SPICE (ASPICE). Future work will focus on extending this methodology to include modular design, performance requirements verification, product line engineering, and trade-off analysis within digital twin environments.

References

Ågren, S. M., Heldal, R., Knauss, E., & Pelliccione, oi4arxiv.org/abs/2203.13130

Askaripoor, H., Hashemi Farzaneh, M., & Knoll, A. (2022). E/E Architecture Synthesis: Challenges and Technologies. Electronics, 11(4), 518. https://doi.org/10.3390/electronics11040518

Ayres, N., Deka, L., & Paluszczyszyn, D. (2024). Container based electronic control unit virtualization: A paradigm shift towards a centralised automotive E/E architecture. Electronics, 13(21), Article 4283. https://doi.org/10.3390/electronics13214283

中国信息通信研究院 [China Academy of Information and Communications Technology] (CAICT). (2024).车路云一体化系统建设与应用指南.

Dassault Systèmes. (2021). MagicGrid Book of Knowledge: A practical guide to systems modeling using MagicGrid (2nd ed.). Dassault Systèmes

丁飞 (Ding), 张楠, 李升波, 边有钢, 童恩, 李克强. (2022). 智能网联车路云协同系统架构与关键技术研究综述. 自动化学报, 48(12), 2863–2885. https://doi.org/10.16383/j.aas.c211108

El Sallab, A., Yogamani, S., Hussain, A., Herzig, J., Shao, Y., Mita, S., & Trivedi, M. M. (2021). Model in the loop testing and validation of embedded autonomous driving algorithms. In 2021 IEEE Intelligent Vehicles Symposium (IV) (pp. 366–373). IEEE. https://doi.org/10.1109/IV48863.2021.9575802

Heyman, K. (2024, January 25). Shifting left using model-based engineering. Semiconductor Engineering. Retrieved from https://semiengineering.com/shifting-left-using-model-based-engineering/

International Council on Systems Engineering. (2023). Systems Engineering Handbook 5th Edition, A guide for system life cycle processes and activities. WILEY.

International Organization for Standardization. (2022). ISO 21448:2022 – Road vehicles — Safety of the intended functionality (SOTIF)

International Organization for Standardization. (2021). ISO/SAE 21434:2021 – Road vehicles — Cybersecurity engineering.

International Organization for Standardization. (2018). ISO 26262-1:2018 – Road vehicles — Functional safety — Part 1: Vocabulary.

Jiang, S., et al. (2024). Vehicle E/E architecture and key technologies enabling software defined vehicle (SAE Technical Paper 2024 01 2035). SAE World Congress Experience. https://doi.org/10.4271/2024-01-2035

Kirchner, S., Purschke, N., Wu, C., Khan, M. A., Dixit, D., & Knoll, A. C. (2025, March). AUTOFRAME – A software driven integration framework for automotive systems (arXiv:2503.04928). arXiv. https://doi.org/10.48550/arXiv.2503.04928

Mauser, L., & Wagner, S. (2024, September 16). Centralization potential of automotive E/E architectures: Quantifying prospects via industry insights. arXiv preprint arXiv:2409.10690. https://doi.org/10.48550/arXiv.2409.10690

Mou, J., Liu, S., & Zhang, H. (2023). Model checking in the loop model-based testing for automotive operating systems [Preprint]. arXiv. https://doi.org/10.48550/arXiv.2310.00973

SAIC. (2021, April 9). 三千亿创新投入 上汽向高科技企业全面转型 智能汽车打头炮 “银河全栈解决方案”出炉. Retrieved from SAIC Group: https://www.saicmotor.com/chinese/xwzx/xwk/2021/55350.shtml

Shetiya, S. S, Vyas, V., & Renukuntla, S. (2024). Zonal architecture development with evolution of artificial intelligence. arXiv preprint arXiv:2412.01840. https://doi.org/10.48550/arXiv.2412.01840

Toghuj, W., & Turab, N. (2023). Cybersecurity of automotive wired networking systems: Evolution, challenges, and countermeasures. Electronics, 14(3), Article 564. https://doi.org/10.3390/electronics14030564

Vignesh, N., Kumar, M., Achuthan, B., Badade, S., Shivakumar, P., & Keshri, A. (2023). Centralized E/E architecture and evolution. SAE Technical Paper Series, 2023-01-1511. SAE International. https://doi.org/10.4271/2023-01-1511

Wen, L., Pan, F., Lin, J., Zhang, Y., Betz, T., Rickert, M., & Knoll, A. (2024). Virtualization & microservice architecture for software defined vehicles: An evaluation and exploration. Electronics, 13(21), Article 4283. https://doi.org/10.3390/electronics13214283

朱学斌(Zhu Xuebin), 张军伟, 等. (2025). 汽车整车正向设计技术发展与展望. [Development and prospects of forward design technology for complete vehicles]. 汽车工程 (2). Retrieved August 2025, from https://wap.cnki.net/touch/web/Journal/Article/QZKJ202502005.html

Biography

YI Xia

Yi Xia is a Systems Engineer specializing in Body Domain systems design and Model-Based Systems Engineering (MBSE) at SAIC Motor Corporation. With several years of experience in automotive systems engineering, she focuses on the analysis and development of vehicle body control functions and system architecture. She also

serves as an internal MBSE instructor within the company, promoting the application of MBSE methodology and tool implementation to enhance system design efficiency and traceability across development projects.

ZHANG Jianqiong

Zhang Jianqiong is a Chief Architect and Director of the Electrical and Electronic Architecture Department at SAIC Motor Corporation, where she oversees system architecture strategy and technical innovation for next-generation intelligent vehicles. With extensive experience in vehicle electronics, systems integration, and model-based development, she has led the implementation of MBSE methodologies across enterprise-wide projects. Her work focuses on defining architecture governance, enabling digital continuity from requirements to verification, and fostering collaboration between system, software, and hardware engineering disciplines.

LI Feng

Li Feng currently works at SAIC Motor R&D Innovation Headquarters, where he manages the design and development management of automotive electronic systems. Having worked in the industry for more than 10 years and extensive experience in automotive electronic function and system development, he is able to draw upon a wide range of experiences, which enables him to lead the system development team, promoting model-based system engineering development methods within the team, and improving the quality and efficiency of the development of complex automotive electronic functions and systems.

WU Lingyan

Wu Lingyan is a CATIA Cyber Systems Industry Process Expert Senior Specialist at Dassault Systèmes. She has over 10 years of experience and expertise specializing in digital engineering transformation. As a business consultant and OMG certified systems engineer, she has participated, assisted, and led many successful digital transformation and MBSE projects in the aerospace and automotive industries in China, ranging from requirements engineering to architecture modeling, system analysis, multidisciplinary simulation, Verification & Validation etc. Her responsibilities in this role include understanding and working on MBSE methodology and enterprise architecture to support customers.

ONG Ho Kit Robert

Robert Ong is an Industry Process Consultant Director at Dassault Systèmes. A seasoned MBSE practitioner with more than 25 years of experience working with clients in a fast-moving environment, Robert brings to his position a well-established background and

strengths in MBSE/MBRE, project management, requirements engineering, IT solutions finding, domain and business process analysis, business process reengineering, and Enterprise Architecture. Outside of his professional endeavors, Robert is deeply committed to giving back to the community. He actively participates in INCOSE as the President of INCOSE Singapore Chapter to advance the practice, education, and knowledge of systems engineering to address complex challenges.

WANG Guangyong

Wang Guanyong is a CATIA Systems Industry Process Consultant, CATIA System at Dassault Systèmes. As a consultant with more than 6 years of work experience, Guangyong heavily focuses on the application and deployment of MBSE technologies. His experience in this business involves roles at the technical consultant level, applying his expertise and knowledge, participating and supporting multiple MBSE projects across the marine, automotive, and industrial equipment industries, providing technical expertise in requirements engineering, system architecture, and model-based process implementation. His objective is to accelerate digital transformation through system modeling and enterprise collaboration using CATIA Magic and the 3DEXPERIENCE platform.