SIEMENS

Ingenuity for life

Siemens Digital Industries Software

The advantages of MBSE-driven E/E architecture development

Executive summary

Vehicles in all sectors are growing in complexity as OEMs develop sophisticated platforms with growing levels of automation and connectivity. To cope with this growing complexity, automotive, aerospace and commercial vehicle OEMs must evolve their architectural design processes to leverage MBSE and the digital thread. Today's E/E system engineering solutions help companies implement MBSE by providing robust data continuity, advanced automation capabilities and closed-loop verification and design optimization. With these solutions, engineers can use existing functional models to generate vehicle architectures and more detailed system designs, continuously building on data from upstream processes to ensure traceability from functions to implementations and actual components or systems. This traceability will be crucial to demonstrate regulatory compliance and the safety of advanced vehicle platforms.

Hans-Juergen Mantsch, Siemens Digital Industries Software

siemens.com/ee-systems

Contents

Introduction

Vehicles in all sectors are growing in complexity as OEMs develop sophisticated platforms with growing levels of automation and connectivity. To cope with this growing complexity, automotive, aerospace and commercial vehicle OEMs must evolve their architectural design processes to leverage MBSE and the digital thread. Today's E/E system engineering solutions help companies implement MBSE by providing robust data continuity, advanced automation capabilities and closed-loop verification and design optimization. With these solutions, engineers can use existing functional models to generate vehicle architectures and more detailed system designs, continuously building on data from upstream processes to ensure traceability from functions to implementations and actual components or systems. This traceability will be crucial to demonstrate regulatory compliance and the safety of advanced vehicle platforms.

Original equipment manufacturers (OEMs) in the automotive, aerospace and heavy-duty/off-highway industries are beginning to face common challenges as vehicle technologies converge towards greater levels of autonomy and connectivity (figure 1). These advanced technologies are expected to improve the safety, productivity and capabilities of passenger cars, aircraft and agricultural and other heavy equipment. Enabling these technologies, however, is driving a rapid increase in the complexity of vehicles in all segments due to the sophisticated electrical, electronic and electro-mechanical systems required.

The architectural evolution of vehicles has been driven, so far, by demand for greater vehicle functionality through new and more sophisticated features. For example, consider the evolution of the automotive electrical and electronic (E/E) architecture over the last 20 years. Vehicle architectures used to consist of a few dozen ECUs connected by low-bandwidth networks and low-fidelity signals. These architectures supported basic vehicle features and functions such as the stereo, electric windows, cruise control and anti-lock braking systems. In contrast, a modern mid-range vehicle contains about 90 ECUs connected by a variety of high and low-speed networks to dozens of sensors and actuators. These modern architectures have grown in size and complexity to support new features such as advanced driver assistance systems (ADAS), sophisticated infotainment systems, navigation and more.

Now, more sophisticated vehicle automation, electrification and connectivity systems are driving vehicle OEMs to incorporate new technologies into their vehicles. In particular, manufacturers are attempting to integrate new communications technologies to connect their vehicles to 5G networks, WiFi and to enable vehicle-to-everything (V2X) communications. These communications technologies will allow OEMs to perform over-the-air (OTA) updates to vehicle software to unlock additional functionality even after the sale of the vehicle. But, they also require additional infrastructure in the vehicle architecture to provide security and prevent safety threats from

Figure 1: Vehicles in the automotive, aerospace, and heavy-duty and off-highway industries are becoming more connected and automated.

outside the vehicle and the networks to which it is connected. This will be especially important as vehicle automation increases.

The result of these evolutionary steps is an explosion in vehicle complexity across multiple domains. In 2014, Deutsche Bank conducted a study in which they measured rising vehicle complexity based on the software lines of code (SLOC) and the number of network signals implemented within a typical vehicle at various times. The study predicted that the average vehicle in 2020 would contain 30 million SLOC and 10,000 network signals, both of which were at least double what was reported for a vehicle from 2012. This prediction, however, has proven to fall short of reality (figure 2).

According to conversations with our customers, the typical vehicle in 2020 has 150 million SLOC and 20,000 or more network signals. The rapid development and integration of new vehicle technologies is driving a growth in complexity that has outpaced our predictions from just a few years ago. In addition to more complicated vehicles, OEMs have to manage the lifecycles of all of these software, network and electrical components, along with all of the other vehicle systems and parts. This involves the coordination of the development cycle to support the launch of the lead program, while ensuring that requirements from later programs have been incorporated. This is, of course, very complex. As OEMs manage the production life cycle of the software and components, they must ensure components are appropriately used in the production and service of each vehicle variant and configured to match each vehicle specification. This involves multiple teams in very different parts of the organization.

A small error in the partitioning of functions in the vehicle definition could lead to safety related functionality being dependent on systems with insufficient integrity in the design and development process. This can lead to costly late program changes or, worse, in field recalls to update multiple ECUs after field failures occur. Incorrect component configurations can also lead to incorrect or lost vehicle functionality, causing customer dissatisfaction and additional cost to identify the population of affected vehicles in production and in the field. Managing this complexity and ensuring traceability throughout the development process will be crucial to creating advanced automated and connected vehicles and to bringing these vehicles to market on the tight timelines common in today's automotive, aerospace and commercial vehicle industries.

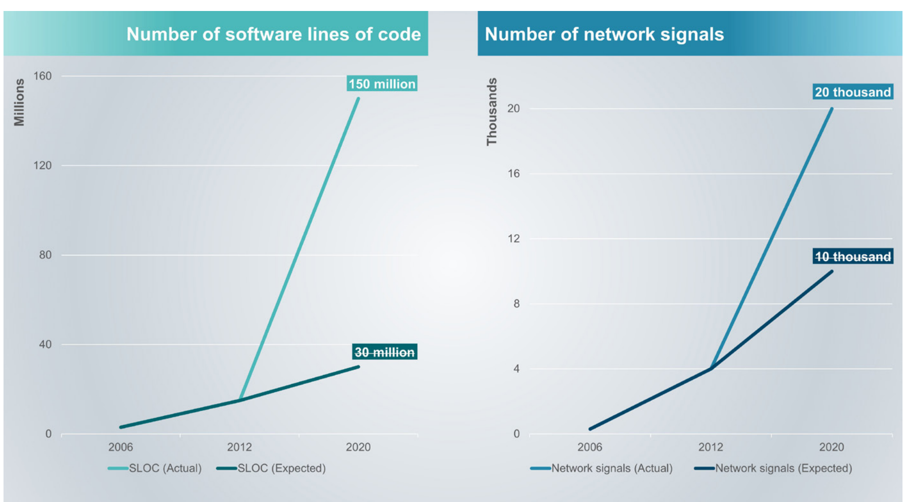

Figure 2: Predictions from a 2014 study conducted by Deutsche Bank have fallen short of the actual increases in SLOC and network signals.

Model-based systems engineering for E/E architecture design

The exploding complexity of modern vehicles demands a change in design methodologies. Where traditional methods relied on manual operations and separated engineering domains into silos, future engineering strategies must employ automation and support collaboration through robust data integrity and integrated solutions. Model-based systems engineering (MBSE), with an advanced portfolio of engineering software solutions, can provide these crucial capabilities for developing the increasingly sophisticated E/E architectures of modern cars, planes and commercial vehicles.

A model-based system engineering flow begins with functional models. These models describe the functionality of various vehicle systems and sub-systems. For example, a functional model may describe a specific software component, electrical signal, or electronic hardware component, such as an analog to digital converter. These models frequently are constructed in a variety of tools from different vendors. The source data behind these models is, therefore, distributed and disconnected. Such disconnected data can become a major pain point in downstream processes, especially in the case of design changes.

The source data for these various systems has to be normalized to a common data model and consolidated into network, software, electrical, hardware and other function types. The normalization of these various inputs to a common data model is critical because it makes the source data for vehicle systems accessible throughout the E/E architecture and other downstream development flows.

System architects can use the normalized data to create E/E architectures, complete with logical, network, hardware and software information. Then, engineers from each domain can interrogate the E/E architecture for domain-specific information to use in the creation of detailed system designs. For example, an electrical engineer can extract logical schematics from the architecture, which they can refine and enhance with additional

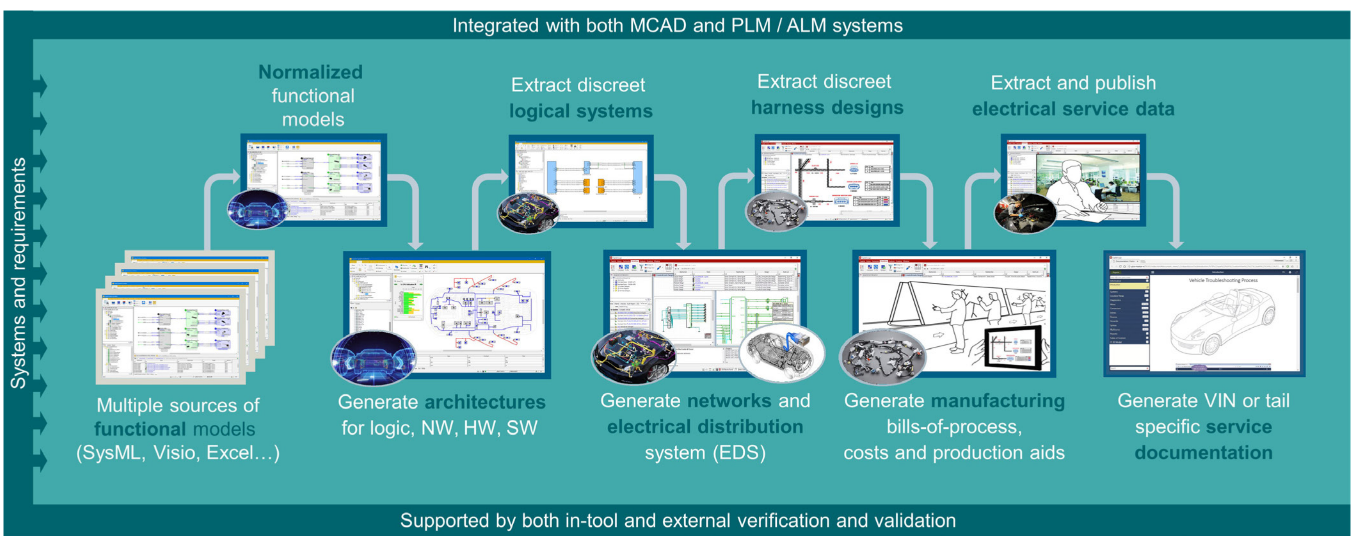

Figure 3: Data continuity enables data from each stage of development to feed the next, ensuring traceability and accelerating development cycles.

information to create the vehicle networks and electrical distribution system (EDS). Then, the network and EDS designs can be used to extract discreet harness designs, which are used, in turn, to create manufacturing data that then leads, finally, to the creation of service data and vehicle-specific service documentation.

A robust data model is critical to enabling a model-based engineering flow, where data remains continuous from definition through manufacturing and service. Data continuity allows the work products from each stage of EDS, network and software development to be used as input for the next step in the flow (figure 4). This continuous digital thread of data up and down the development flow enhances the ability of engineers to manage and implement changes as well. The full impact of design changes can be understood before implementation because each domain works from the same digital thread. Once a change is verified, it can be propagated throughout affected domains and designs quickly.

Furthermore, data continuity provides traceability from functional models to the implementation and documentation of these functions in the vehicle software, networks and EDS. This traceability ensures that engineers can quickly identify the functional source of any component within the vehicle architecture, or (in reverse) find the specific ECUs, networks signals, or pins involved in the implementation of a function.

MBSE enables engineers to leverage existing functional models and engineering data from various environments in the creation of vehicle architectures and more detailed system designs. By continuously building on data from upstream processes, MBSE ensures traceability and streamlines change management and implementation. Much of the processes involved in consolidating models, creating architectures and maintaining traceability, however, are still accomplished manually. Modern E/E systems engineering solutions can improve these processes by automating manual tasks and providing a unified database to ensure data continuity throughout E/E architecture and system design.

Key software capabilities to enhance MBSE for E/E architecture design

Consolidating functional models into a single vehicle platform is, currently, a manual process that consumes a lot of time and energy. Functional models exist in a variety of different systems engineering tools, each with its own method of abstracting vehicle functionality. Refining just one of these models in a traditional systems engineering tool to represent its implementation in the vehicle platform with sufficient detail requires extensive manual effort. Multiply this effort across the hundreds or thousands of models usually required to define a vehicle platform, and the immense scale of the task becomes clear.

Today, E/E systems engineering solutions, such as Capital from Siemens Digital Industries Software, can automate much of this work. Instead of adding the necessary domain detail in the systems engineering environment, advanced E/E systems engineering software can import the functional design abstraction, allowing engineers to embellish this abstraction, at the platform level, with domain details for the software, hardware, networks and EDS (figure 4). The software then uses rule-based synthesis to deploy functionality in the vehicle platform with the granularity required for subsequent domain engineering processes.

These specialized E/E engineering solutions also feature built-in metrics for both logical and physical key performance indicators (KPIs), including cost, bandwidth utilization and more. These metrics can drive early optimizations and, coupled with rule-based synthesis, rapid iteration on the E/E architecture. Design rule checks can then identify violations or issues in the physical design abstraction, such as excess bandwidth or ECU utilization.

For example, consider a functional design for a software component that was built in Excel. The E/E engineering solution can import this design, along with the hundreds or thousands of others needed for this vehicle, and deploy the functionality to create a vehicle platform. After deploying functionality, the built-in metrics can show how much RAM is being used per ECU in the architecture

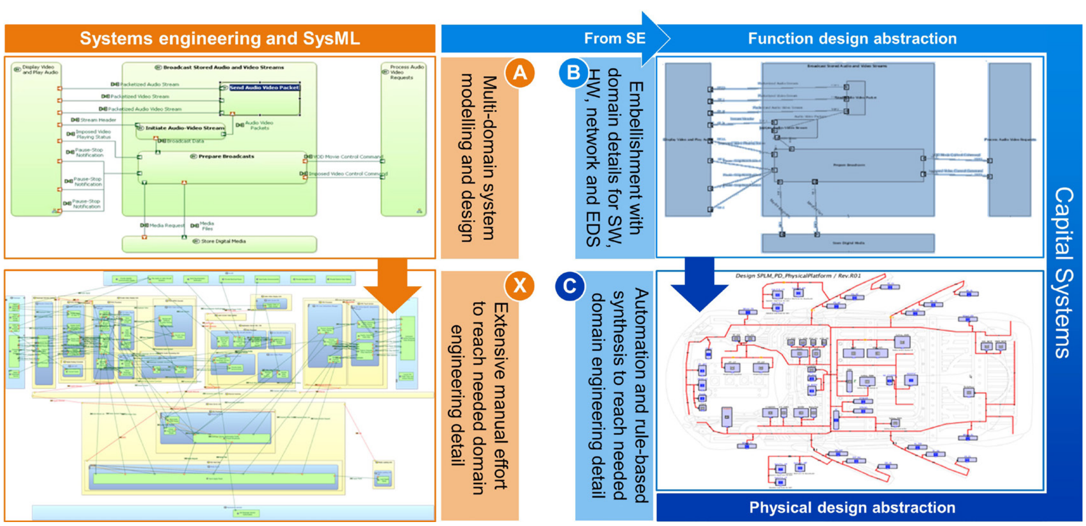

Figure 4: Advanced E/E systems engineering solutions accelerate architectural and domain engineering processes by allowing engineers to embellish the functional abstraction at the system level.

allowing engineers to make trade-off's. Additionally, the engineer can quickly see how much CPU load the current functional allocation will create in each of the ECUs around the architecture, adjusting the allocation if necessary. Once an adjustment is made, the engineer can synthesize an updated architecture, and continue to refine in this manner.

The result is a verified multi-domain vehicle platform that is already optimized for platform KPIs. As the platform is generated, the E/E engineering solution ensures data continuity and traceability from the platform back to the source functional designs. This data continuity will also support collaboration between teams and traceability as they perform the detailed design work to create the logical, networks, software and hardware systems. Each element will automatically be tied to its functional abstraction and physical implementations. A single pin on a connector in a wiring harness can be traced back to the wiring diagram, logical schematic and functional design in which it was defined.

Advanced E/E engineering solutions, like Capital, take this a step further. Integrations with product and application lifecycle management solutions (PLM/ALM), such as Teamcenter and Polarion, allow for the digital thread to extend all the way back to the product configurations, requirements and constraints (figure 5). This comprehensive traceability ensures that engineers can understand each component or function in the architecture and how it was implemented, but also why it exists in the first place. As a result, vehicle OEMs can produce detailed and accurate compliance documentation automatically, without needing to hunt for information.

Data continuity and traceability also streamline the management, implementation and propagation of engineering changes. Sometimes, even changes made for the right reasons can result in an electrical problem in a vehicle in the field. Capital's data model allows engineers to view and understand the impact of changes on the vehicle level. This visibility prevents engineering changes from creating issues in other related systems. Automated change management capabilities can also propagate design changes to all affected systems, ensuring they are implemented correctly across the vehicle.

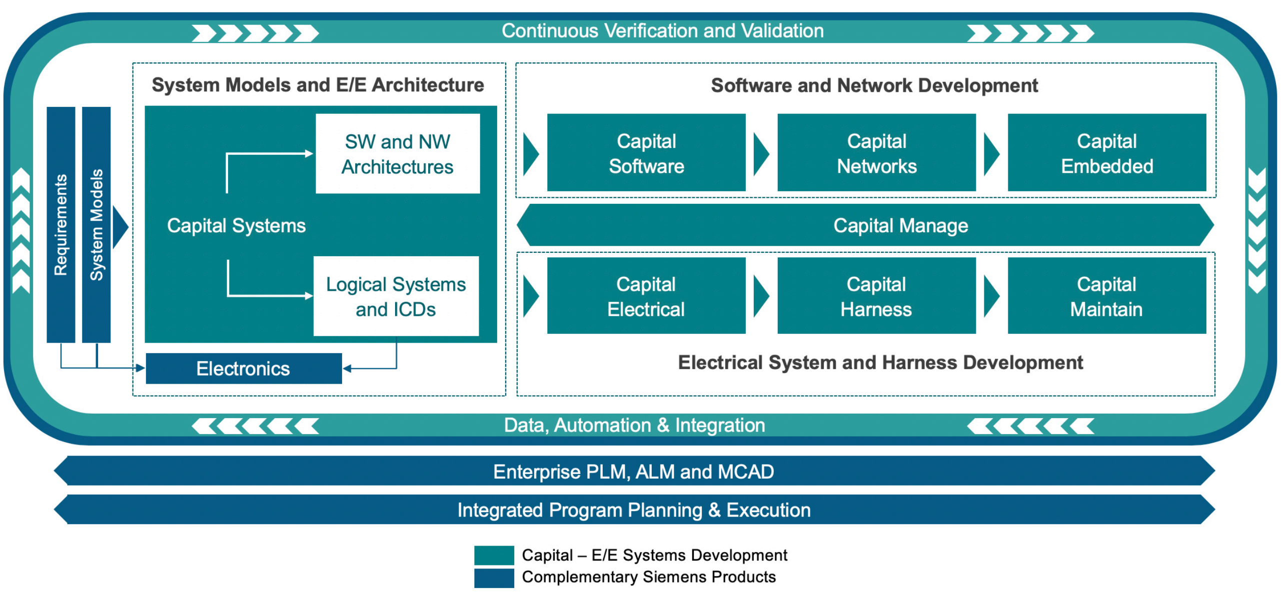

Figure 5: Integrations with ALM and PLM solutions enable the digital thread to extend all the way back to product requirements and definitions.

Connecting software development and integration

Embedded software applications are now critical to enabling vehicle functionality in passenger cars, aircraft and commercial vehicles. The growing importance of software means that engineers must account for these applications during architectural design and functional allocations. A recent addition to the Capital suite, Capital Software Designer enables software engineers to collaborate and synchronize with the architectural design directly to develop embedded software application requirements within the context of the system definition (figure 6). With these requirements, engineers can orchestrate software models and control algorithms, and validate functionality before code is implemented.

Capital Software Designer features automation and contract-based software design, allowing software engineers to refine the high-level architecture into digital components that will be implemented in software applications. Major automotive, aerospace, and heavy equipment OEMs also possess a lot of legacy software. Engineers can import existing code in C as well as SysML, Matlab, and others, to update legacy software and enable its reuse in new vehicle platforms. In either case, the necessary software components are connected to the overall vehicle architecture, enabling model-based synthesis of software architecture models.

Integrated formal verification technology can mathematically prove the absence of inconsistencies in the software architecture and software component implementation. Capital Software Designer can also connect with various simulation environments to conduct model, software, hardware, and vehicle-in-the-loop (MiL/SiL/HiL/ViL) tests of embedded applications. This enables closed-loop verification of the software architecture and components within various vehicle abstractions, ensuring that

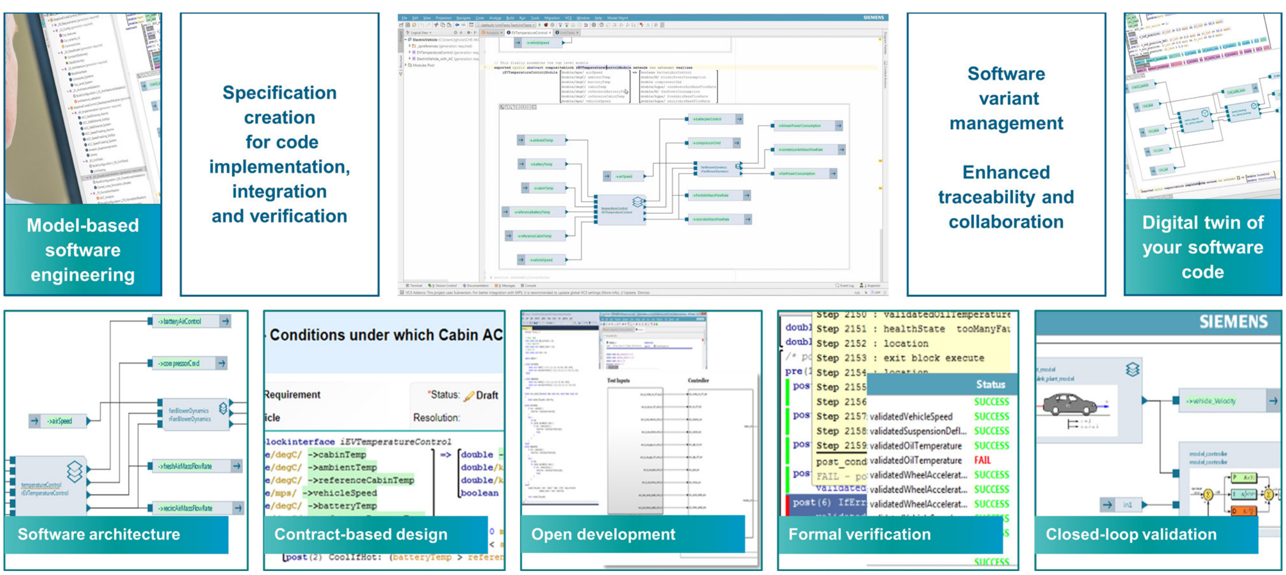

Figure 6: Capital Software Designer enables the integration of software architecture and component design into the E/E systems engineering flow.

software is integrated into the architecture effectively. At the same time, ECU specifications are derived from the E/E architecture, allowing application code to be developed in the context of virtual ECU models. The basic software configurations for these ECUs can also be developed and tested from the extracted ECU specifications.

Function allocations and signal definitions can be used to derive and design the vehicle networks using Capital Networks, considering the full network topology and its multiple technologies such as LIN, CAN, CAN-FD, FlexRay and Ethernet. Comprehensive timing models can then be used to validate that the behavior of the network design meets functional requirements, enabling the delivery of correct-by-construction output files for internal teams or Tier 1 partners.

Next, AUTOSAR basic software can be configured using Capital VSTAR Integrator. This process leverages the model data from earlier in the flow to structure and generate configured software templates, validating them against internal rules and models to ensure the outputs are ready to use. The software can then be run virtually before target hardware is available with Capital VSTAR Virtualizer to allow behavior verification and debugging to take place early in the development flow, saving valuable time on early sample hardware.

An ALM framework, based on Polarion, ties the development of embedded software together. ALM tracks software development to ensure that applications meet platform-level requirements, providing traceability along the way. ALM also coordinates software verification and validation. From the ALM solution, engineers can trigger functional mock-up simulations that combine each of the available vehicle abstractions to see how a virtual model of the entire vehicle responds in a virtual environment. Rather than creating a physical prototype, this approach integrates MiL, SiL, HiL and ViL simulations to provide a multi-domain picture of the vehicle functionality. This saves considerable cost and time compared to physical prototyping for early verification and validation. Finally, ALM solutions track the configuration and delivery of applications to each vehicle variant, ensuring that delivered software supports the specific mix of features on each vehicle.

Conclusion

Vehicles in all sectors are growing in complexity as OEMs develop sophisticated platforms with growing levels of automation and connectivity. In particular, the integration of these technologies is driving a rapid expansion in the size and complexity of vehicle architectures as additional electronics hardware, software applications, networks and other architectural components are required to support the advanced functionality. The complexity of these modern E/E architectures makes them extremely challenging to design and verify using traditional, labor-intensive procedures that are in place at many manufacturers. In addition, maintaining traceability of vehicle requirements, functions and implementations through each vehicle abstraction becomes nearly impossible.

To cope with this growing complexity, automotive, aerospace and commercial vehicle OEMs must evolve their architectural design processes to leverage MBSE and the digital thread. Today's E/E system engineering solutions help companies implement MBSE by providing robust data continuity and advanced automation capabilities. With these solutions, engineers can use existing functional models to generate vehicle architectures and more detailed system designs, continuously building on data from upstream processes to ensure traceability from functions to implementations and actual components or systems. These solutions also provide closed-loop verification and optimization of architectures and system designs to improve vehicle performance at the architecture level.

As vehicles become more automated, the traceability provided by an MBSE flow with advanced E/E engineering tools will be crucial to demonstrate regulatory compliance and the safety of a vehicle platform. A continuous digital thread throughout vehicle development and manufacturing will enable the generation of accurate documentation directly from vehicle design data. Such a digital thread can also track test results and show that vehicle systems behave as expected. Such a capability will be imperative as OEMs are being asked to demonstrate the safety and reliability of their systems before, during and after production.

As vehicles become more automated, the traceability provided by an MBSE flow with advanced E/E engineering tools will be crucial to demonstrate regulatory compliance and the safety of a vehicle platform.

Siemens Digital Industries Software

Headquarters
Granite Park One
5800 Granite Parkway
Suite 600
Plano, TX 75024
USA
+1 972 987 3000

Americas
Granite Park One
5800 Granite Parkway
Suite 600
Plano, TX 75024
USA
+1 314 264 8499

Europe
Stephenson House
Sir William Siemens Square
Frimley, Camberley
Surrey, GU16 8QD
+44 (0) 1276 413200

Asia-Pacific
Unit 901-902, 9/F
Tower B, Manulife Financial Centre
223-231 Wai Yip Street, Kwun Tong
Kowloon, Hong Kong
+852 2230 3333

About Siemens Digital Industries Software

Siemens Digital Industries Software is driving transformation to enable a digital enterprise where engineering, manufacturing and electronics design meet tomorrow. Our solutions help companies of all sizes create and leverage digital twins that provide organizations with new insights, opportunities and levels of automation to drive innovation. For more information on Siemens Digital Industries Software products and services, visit siemens.com/software or follow us on LinkedIn, Twitter, Facebook and Instagram. Siemens Digital Industries Software – Where today meets tomorrow.

siemens.com/software

81781-C4 4/20 A