A new approach to automotive electric/electronic engineering life-cycle management

Managing engineering data and processes using a single source of truth
Introduction

In 1977, the Oldsmobile Toronado had a single computer unit for spark-plug timing. Today, cars may contain 50 or more computerized devices sensing and processing hundreds of variables. Radar sensors measure the car’s distance from objects in the front and back and provide this input to various safety systems, such as predictive collision avoidance systems. Rain sensors monitor the intensity of rain, and an electronic control unit (ECU) operates the wiper blades automatically. Motion sensors monitor the vehicle’s position, providing real-time data to airbags and driver assistance systems. This type of embedded hardware and software content in vehicles, also referred to as electric/electronic (E/E) content, may have once been considered a bonus or a luxury, but now it’s a key differentiator that can make or break a brand. As consumers come to expect more—more safety features, more comfort and more environmental friendliness—automotive original equipment manufacturers (OEMs) must continue to innovate to succeed in what has become a fiercely competitive marketplace. And the majority of these innovations likely come from E/E features.

What this all means is that cars that used to contain a few thousand lines of code of embedded software in a couple of ECUs now require millions of lines of code running dozens of E/E systems at the same time has become increasingly strategic and increasingly complex. E/E is essentially the battleground on which OEMs and their suppliers fight for differentiation and strive to deliver the most innovative automobiles to the marketplace. E/E may represent nearly half the value of a new midclass automobile. Getting E/E engineering wrong can lead to huge financial losses through recalls and damaged brand image. Lagging behind in E/E basically paves the way for the competition.

E/E development demands strength in systems and software engineering

Both automotive suppliers and OEMs are struggling because their current engineering practices are unable to keep up with the epic expansion from traditional mechanical engineering to electronic and embedded software development. It has become clear that the automotive industry must embrace a combination of software and systems engineering to succeed. Software engineering is essential to deliver the innovation provided by the E/E components. Systems engineering, which can be defined as a multidiscipline approach to creating a system or product that includes mechanical components, hardware, software, people and information, is essential for a number of reasons:

- It empowers organizations to manage the links within E/E development as well as in relation to mechanical, hydraulic and fluid engineering components.
- It serves as an umbrella discipline to hold the relevant workflows together because the different engineering disciplines have different processes and attributes. For example, software can be modified up until practically the last moment, whereas mechanical components cannot.
- It facilitates collaboration throughout the development supply chain to help prevent compatibility issues from arising later in production. This is even more essential when products require matching capabilities from OEMs and their related suppliers.
In the past, managing automotive development was simpler because the focus was on managing single components that each delivered a set of functions and features. But now, OEMs must manage not only the components but also the functions and features that may be delivered via several interconnected components. While it's still necessary to look at each component, successful automotive development now requires multiple views of the entire logical, functional and physical product architecture. Architectural frameworks that provide these views therefore make it easier to communicate and validate the product, resulting in faster time to market. Such frameworks enable a comprehensive systems engineering methodology, which provides a more abstract, model-based approach that's well suited to today's complex systems.

**Definition**

**Engineering life-cycle management** refers to the holistic, structured and consistent management of all processes or activities and related artifacts or data that are generated and used during the product development life cycle.

**A multitude of processes and workflows**

E/E engineering includes a staggering number of processes and workflows because of the vast number of different engineering disciplines and application areas. OEMs and suppliers alike must manage their product portfolios, which consist of various product lines and variants developed through a huge number of engineering programs and projects.

Although the processes followed by the various engineering programs and projects and spanning different engineering disciplines and organizations are complex enough by themselves, OEMs and suppliers must additionally adhere to process guidelines such as Capability Maturity Model Integration (CMMI) and Automotive SPICE (Software Process Improvement and Capability dEtermination), and they must comply with necessary safety regulations such as the upcoming ISO 26262. Additionally, many are migrating to standardized platforms such as AUTomotive Open System ARchitecture (AUTOSAR).

The separation of tasks between OEMs and suppliers varies depending on the nature of technology and the level of interaction and supply chain management. However, it's safe to say that, in general, OEMs’ processes also include the following:

- Defining and detailing customer features and creating feature catalogs using context diagrams
- Deriving the related requirements as well as logical and functional blocks for building up the E/E architecture
- Creating test cases to test defined customer features as part of quality management processes
- Detailing logical and functional blocks; defining interfaces; and visualizing vehicle functions, their interrelations and their behavior
- Specifying timing and performance constraints using a functional network or a domain-specific or proprietary modeling language
- Developing an initial E/E architecture by defining the in-vehicle network topologies and the ECUs within them
Defining and specifying software components and mapping them to ECUs

- Updating and refining the E/E architecture, using trade-off analysis to determine the best possible architecture
- Developing an E/E network layout and defining bus messages for the various bus systems
- Detailing and updating software component specifications, often according to the AUTOSAR methodology
- Integrating different models from different tools into a single, coherent architecture
- Establishing traceability across hardware and software development; enacting process workflows
- Demonstrating compliance with applicable regulations, including ISO 26262

Suppliers must contend with processes such as the following:

- Collaborating with OEMs on component requirements and in several other process areas
- Negotiating and finalizing mechanical component, ECU and software requirements
- Developing, testing and integrating mechanical components, ECU hardware and software
- Packaging and delivering completed components and ECUs

The challenge of managing E/E engineering data and processes

All of the processes and workflows in E/E development generate massive amounts of data. Managing and documenting all of the possible configurations and variants of a single model line increasingly requires skilled labor to handle the vast number of possible automobile setups. In addition, more stakeholders are involved in the process than ever before—OEMs, suppliers, engineering teams, sales teams, marketing, finance departments and purchasing departments. Keeping everyone informed and on the same page is not an easy task, particularly because companies are using multiple, disparate tools for E/E engineering. Many of these tools provide proprietary user interfaces, logic and storage repositories, which makes it difficult to integrate them and therefore drives up maintenance costs.

Ultimately, E/E engineering data and process management practices must go beyond the engineering life cycle. In fact, only part of the E/E engineering data is even related to the final product. Much of the data generated during engineering is interim data only used in development, such as simulation models, simulation results and trade-off studies. A majority of these development assets remain within the engineering department to be reused and added to the general knowledge base.

So E/E engineering data management is about much more than product data management. E/E engineering data management must span the entire vehicle life cycle. After engineering is complete, vehicles are monitored, controlled and linked with multiple business processes such as spare parts management. Each ECU and its associated embedded software must be managed throughout the entire life of the vehicle because these configurations can change. Software updates during maintenance are common, allowing suppliers to resolve known issues as well as update and improve existing features.
If a quality issue arises, OEMs and suppliers must be able to trace the root cause back to each single variant and configuration so they can take corrective action. Only a closed loop between product development and service management processes will allow OEMs to address virtually any issue that arises.

But if OEMs and suppliers continue to use existing heterogeneous, nonintegrated and proprietary solutions to manage E/E data and processes, they will continue to see issues including loss of information, inconsistencies, lack of reuse and delays in the development cycle. Ultimately, proprietary E/E data structures prevent a holistic view into the E/E development process.

**An alternative to a single, common repository**

Many automotive E/E engineering teams attempt to solve the issues of managing E/E data and processes by establishing one common repository for all data. However, this approach often fails in the long term because it's extremely costly to rip and replace an entire E/E engineering solution if an upgrade or extension is required. In addition, this approach requires a massive effort up front to migrate existing engineering data into the new repository. And sharing data that's in a proprietary repository format can be challenging, hindering the adoption of new engineering technologies. The reality is that automotive engineering has become so complex and so massive in scope that any engineering data management solution must include support for multiple vendors' products and data.

Just about the only way that automotive E/E processes can be properly controlled is if data objects from multiple systems are consistently defined and are linked to each other. Only by doing this can companies monitor configuration and change throughout the entire life cycle of the vehicle, well beyond the engineering life cycle. Configuration, version, variant and change management must be embedded in all functional development processes, such as order, sales, design, validation, test, quality assurance, product documentation, manufacturing engineering, supplier collaboration, service, maintenance, warranty management and compliance management.

**A comprehensive E/E engineering life-cycle management strategy from IBM**

The IBM Rational® software platform for automotive systems is an open, standards-based, extensible collaboration and integration platform that supports the management of the artifacts generated within automotive E/E engineering as well as the management of related development workflows and processes. The IBM solution provides a comprehensive set of offerings that can help engineering teams effectively implement an E/E engineering life-cycle management solution. It includes domain-specific process templates supporting compliance with ISO 26262 and domain-specific architecture frameworks based on EAST-ADL2 and AUTOSAR.
A new approach to automotive electric/electronic engineering life-cycle management

Use modeling to validate requirements, architecture and design throughout the development process

Manage all system requirements with full traceability across the life cycle

Collaborate across diverse engineering disciplines and development teams

Automotive quality management with an integrated, automated testing process

Figure 1: The Rational software platform for automotive systems uses modeling to validate requirements and architecture.

The platform enables engineering teams to define, model, specify, simulate, develop, test and integrate E/E systems, ECU's and embedded software. The Rational software platform consists of integrated tools that can help distributed teams collaborate quickly and easily, helping reduce design errors, control labor costs and speed the development of in-vehicle systems. It also offers tools for automating time-consuming, labor-intensive manual tasks such as coding and testing. These tools deliver the following capabilities:

- Embedded software development and test
- Configuration and change management
- Asset management and reuse
- Quality management
- Measurement and reporting

Plus, the platform from IBM also enables companies to integrate existing third-party authoring systems, so they don't have to rip and replace the tools they've already invested in.

- Product portfolio management
- Requirements engineering
- Model-driven systems development (MDSD)
Based on open technologies and standards

The Rational software platform for automotive systems is based on the IBM Jazz™ platform and Open Services for Lifecycle Collaboration (OSLC), which means that it is open and extensible, easily integrates with third-party solutions, and supports data migration from other solutions. The IBM Jazz platform is a major initiative supported by IBM to help create a scalable, extensible and collaborative systems life-cycle management platform. It serves as an integration architecture that enables mashups and products not based on the Jazz platform to participate. There’s an active community at the Jazz Community Site, where numerous Jazz platform–based products are built.

Initially proposed by IBM in 2008, OSLC is an industry initiative aimed at simplifying collaboration across the software and systems delivery life cycle. The initiative aims to enable teams to use disparate tools and share life-cycle resources, whether the tools are from IBM, other vendors, open source projects or in-house development. The OSLC initiative describes an integration architecture and set of web protocols and services for life-cycle collaboration based on the principles of representational state transfer (REST). More information, including an initial set of descriptions for life-cycle resources such as requirements and test cases as well as protocols and services for accessing these resources, can be found at www.open-services.net
The IBM solution provides an open environment with multiple integration levels supporting an open ecosystem of collaboration vendors and customers. Because it’s based on open standards, the IBM solution can be integrated with mechanical engineering systems and with product life-cycle management (PLM) systems.

**Enables teams to centrally manage artifacts of different types, from different tools**

The Rational software platform and its E/E engineering life-cycle management solutions provide an E/E architecture framework that serves as an organizational umbrella to help logically manage artifacts of different variants and product lines. The artifacts can be stored in their original format and can leverage IBM artifact life-cycle management capabilities including version control and baselining, change management, stakeholder approval, workflow automation, work item tracking, reporting with metrics, project planning, and team collaboration. In other words, artifacts from third-party or proprietary tools do not have to be exported and stored in the IBM platform to benefit from its capabilities.

**The IBM solution is designed to enable engineering teams to access and manage virtually all types of E/E engineering data and artifacts—from a variety of tools and formats.**

**Supports a wide variety of resources**

The E/E engineering life-cycle management solution from IBM provides data management capabilities for key resources including textually and graphically represented requirements; test cases and test suites; E/E architecture models; ECU specifications; AUTOSAR input descriptions; and C, C++ and Java™ source code, to name just a few. Supported types of resources include the following:

- Unstructured files, such as source code, make and debug files, text files, binary files, and documents
- Structured artifacts and their composing objects such as Systems Modeling Language (SysML) or Unified Modeling Language (UML) models and AUTOSAR XML files
- Change and workflow artifacts, such as user tasks, enhancements, defects and other types of work assignable to a user
- Project/iteration schedules and work breakdown structures
- Configurations of related artifacts (components), work streams, baselines and workspaces (all forms of resource containers)
- Metadata definitions that show how the E/E data model is defined and managed

**Includes a customizable automotive E/E engineering metamodel**

The Rational software platform provides a generic automotive E/E metamodel based on the EAST-ADL2 metamodel and supporting the AUTOSAR methodology as well as the
upcoming ISO 26262 standard. This provided metamodel can be used as-is or adapted to address companies’ unique needs. As shown in figure 3, the metamodel defines layers in a hierarchical structure and the content of the E/E architecture framework, in which users can search, query and navigate data elements from various artifacts; use the traceability features to manage relations between data elements in different artifacts; and launch the appropriate authoring tools to edit the artifacts. So unlike conventional source code configuration management tools, the IBM solution allows a user-defined, metadata-defined hierarchy of artifacts that enables engineering teams to build their own automotive domain model.

The automotive E/E metamodel is defined and managed separately from the actual artifact storage. Artifacts can be stored inside the Rational software platform, in Jazz repositories or outside of the platform in a variety of internal resource formats according to the particular authoring tool. The authoring tools can retain their own internal data model, but information is extracted from those artifacts along with information about relationships between artifacts and data elements.

By separating the storage of metamodel and managed artifacts, new tools can be integrated with the system incrementally. By mapping the original artifacts into an appropriate representation in the specific automotive domain model and by keeping a reference to the original data source, the IBM platform can present a single, unified view of the artifacts and their relationships while still preserving the rich semantics of the original authoring tools. This is an efficient management of what we call a web of loosely coupled engineering data. Following this approach, the IBM solution provides a single source of truth rather than a single repository.

A single source of truth

The IBM engineering life-cycle management solution doesn’t require teams to force all of their data into a single repository. Instead, it offers a single source of truth, rather than a single repository.
A summary of the solution benefits

The IBM Rational software platform for automotive systems is an open, extensible platform that allows engineering teams to integrate third-party and custom-built tools along with their engineering data. The platform enables teams to map artifacts to an architecture framework and does not require complex data migration or import/export. The platform also allows, but does not require, all data to be persisted within Jazz servers, giving a significant degree of flexibility. Because authoring tools can retain their own rich semantics, companies can use the best-in-class authoring tools and map different authoring tool artifacts to a common metamodel—without losing semantics or the richness of the original data.

With these engineering data management capabilities, it’s easy to group versions to configurations and baselines and link them to work items, which can have a different status depending on defined workflows. Work items and workflows are elements within project and iteration planning, which are by themselves assigned to a predefined vehicle, an E/E system and ECU product development processes. The platform enacts engineering processes and strongly supports team collaboration throughout the E/E life cycle by providing the user the right data in the right context within the right teams.

Figure 4: A federated integration architecture enables flexible engineering data management.
The IBM solution for engineering life-cycle management can logically connect vehicle features and functions, product requirements, E/E architecture, ECU hardware and software functional designs, implementations, and related test cases. Making these connections can help establish the required traceability between the various artifacts, analyze the impact of changes and show the status of verification and validation efforts. These connections can also help OEMs and suppliers achieve compliance with ISO 26262.

The solution enables engineering teams to define multiple viewpoints on the E/E architecture. It also supports filtered views and queries across engineering artifacts, which can be grouped and organized in many hierarchical ways. Ultimately, by providing multiple viewpoints across different artifacts and relationships, it becomes easier for vehicle and product engineers to access the right set of artifacts at one time, reducing the amount of time wasted searching for data. The IBM approach can also help minimize miscommunication and the probability of using wrong data. Fortunately and finally, the manual and error-prone methods for connecting data from various tools and artifacts are stepwise becoming a thing of the past.

**Conclusion**

The IBM Rational software platform for automotive systems provides a scalable, robust, collaborative engineering life-cycle management solution. It doesn’t require companies to rip and replace existing tools. Instead they can be adopted or integrated incrementally. It provides rich support for collaboration with built-in process guidance as well as change management capabilities with comprehensive process traceability and auditing. It also delivers strong support for process measurements and metrics along with integrated project planning and control features that deliver continuous, real-time project insight.

The provided AUTOSAR, ISO 26262 and EAST-ADL2 templates, workflows and process elements can be configured to address an organization’s unique needs using the metamodel, speeding the solution’s time to value, allowing teams to quickly improve quality and reducing the cost of implementing the solution. Companies using tools from the Rational engineering life-cycle management solution have realized a number of benefits:

- An increase in productivity of up to 40 percent
- A reduction in defects of up to 75 percent
- Improved project management
- Better use of remote staff
- Improved team utilization
- Greater collaboration
- More transparent knowledge sharing
- More effective transfers of work
- Greater product quality
- Less risk of project failures
- Greater adherence to best practices
For more information
To learn more about the IBM Rational software platform for automotive systems, contact your IBM sales representative or IBM Business Partner, or visit: ibm.com/software/rational/solutions/automotive/

Additionally, financing solutions from IBM Global Financing can enable effective cash management, protection from technology obsolescence, improved total cost of ownership and return on investment. Also, our Global Asset Recovery Services help address environmental concerns with new, more energy-efficient solutions. For more information on IBM Global Financing, visit: ibm.com/financing

© Copyright IBM Corporation 2010
IBM Corporation
Software Group
Route 100
Somers, NY 10589
U.S.A.

Produced in the United States of America
November 2010
All Rights Reserved

IBM, the IBM logo, ibm.com, and Rational are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at “Copyright and trademark information” at ibm.com/legal/copytrade.shtml

Java and all Java-based trademarks and logos are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

References in this publication to IBM products or services do not imply that IBM intends to make them available in all countries in which IBM operates.

The information contained in this document is provided for informational purposes only and provided “as is” without warranty of any kind, express or implied. In addition, this information is based on IBM’s current product plans and strategy, which are subject to change by IBM without notice. Without limiting the foregoing, all statements regarding IBM future direction or intent are subject to change or withdrawal without notice and represent goals and objectives only. Nothing contained in this documentation is intended to, nor shall have the effect of, creating any warranties or representations from IBM (or its suppliers or licensors), or altering the terms and conditions of the applicable license agreement governing the use of IBM software.

IBM customers are responsible for ensuring their own compliance with legal requirements. It is the customer’s sole responsibility to obtain advice of competent legal counsel as to the identification and interpretation of any relevant laws and regulatory requirements that may affect the customer’s business and any actions the customer may need to take to comply with such laws.

Please Recycle