

AUTOSAR as a Key Enabler for Collaborative Product Development

2010-01-2341

Published
10/19/2010Mark Jensen and Tony Mascolo
Vector CANtech, Inc.

Copyright © 2010 SAE International

ABSTRACT

Whether it be in highly visible features like fascinating new infotainment systems or hidden behind the scenes in complex new hybrid powertrain controls, in-vehicle software is rapidly changing the way the automotive industry engages its vehicle-buying customers. In every application where a compelling new electronic solution is emerging, it is enabled by the convergence of in-vehicle software developed by different collaborating partners. As more and more component suppliers, vehicle OEMs, and technology vendors enter into collaborative software development projects with each other, a new set of technical and business challenges are showing collaborative software development to be a very distinctive proposition than traditional stand-alone development. This paper will highlight key AUTOSAR concepts that provide an open-standard framework for collaborative development of in-vehicle software that provides natural boundaries for both the technical and business interfaces between partners and their software.

INTRODUCTION

Automotive electronics and in-vehicle software are fertile ground for innovation. Where innovative and compelling new electronic product content is emerging, it is enabled by the convergence of in-vehicle software developed by different collaborating partners, including key technology companies from the consumer electronics, telecommunications, information technology, and defense industries.

In traditional stand-alone software development, all software developers work together in a single team under a single leadership structure to deliver a product that does not interact with other products from other organizations. The challenges of these teams are well-documented by vast bodies of work in both academia and industry. There are a variety of software

development philosophies and technologies that successfully address these challenges. However, this independent stand-alone software development scenario is almost non-existent in today's automotive industry.

With almost no exceptions, software development in today's automotive industry involves multiple development teams working in different companies, in various geographic locations with unaligned processes to deliver software that directly interacts with software from other teams. Most modern vehicles contain software from dozens of such development teams, compiled from millions upon millions of lines of source, and distributed over dozens of networked electronic controls. This pervasive interaction between the vast collection of software components in the vehicle means that collaboration is not a special case at all, but rather a norm that is already in place across the entire industry - even if it is not formally recognized in most cases. With the realization that collaboration is already an integral part of the way the automotive electronics industry does business comes the requirement to master it. Collaborative software development comes in several forms. Each form of collaborative development faces challenges that have yet to be effectively and consistently mastered and managed by the automotive industry.

AUTOSAR is an emerging standard software framework designed especially to meet the needs of automotive electronic systems. AUTOSAR is developed by a voluntary consortium of companies from the automotive electronics industry. The AUTOSAR consortium specifies the AUTOSAR software framework in a collection of documents that all consortium member companies can utilize. While the AUTOSAR consortium strives to achieve a multitude of goals, enabling efficient collaborative software development is a central theme and objective.

AUTOSAR offers significant opportunities for the automotive electronics industry to address the challenges of collaborative software development and even master it. With AUTOSAR, it is possible to enable collaborative product development through a common context that all industry participants can understand and work within. Software and software development processes in the AUTOSAR context are inherently collaborative in nature and implicitly eliminate many of the challenges facing less-structured approaches to collaborative development.

This paper will examine three points -

- Software is the single biggest and key enabler for collaborative product development in automotive electronics.
- Traditional ad-hoc forms of collaborative software development face serious challenges to efficiency, productivity, and severely limit the value of collaboration.
- AUTOSAR provides a context that implicitly enables the collaborative development of automotive electronics products and eliminates many of the challenges facing traditional collaborative development efforts.

SOFTWARE AS THE KEY ENABLER TO COLLABORATIVE DEVELOPMENT OF AUTOMOTIVE ELECTRONICS

Proof of the central role and value of software in product innovation is easy to ascertain from some of the current high-profile innovations in the automotive industry -

- Adaptive cruise control starts with a typical cruise control system and adds a radar system. Software implementing algorithms for pattern recognition and drivetrain management binds the two systems together.
- Hybrid propulsion starts with a traditional internal combustion engine and adds an electric motor. Controls software coordinates the use of the two power plants to deliver an optimal combination of performance and operating efficiency.
- Popular new infotainment systems start with a traditional audio system and add a microphone, a GPS receiver, a Bluetooth wireless transceiver, and a serial data port like USB. Voice recognition software, digital maps, navigation software, MP3 playback software and an overarching software application glue all of the hardware components together to deliver integrated entertainment, telephony, and navigation capabilities through a single intuitive and consumer-friendly interface the vehicle-buying customer sees as a single system.
- Connected vehicle technologies will start with traditional vehicle electronics and add WiFi or other wireless

connectivity to transform the entire vehicle into a single node in an ad-hoc network. Vehicle nodes and smart infrastructure nodes will discover each other and interact through software-enabled services that each node can provide or use.

In each of these examples, two, three, or more partners are jointly and yet independently working to create new and innovative vehicle electronic systems and solutions that none could effectively develop and deliver on their own. In all of these examples, in-vehicle software is the key enabler that joins these complementary technologies and partners together. As software continues to expand in scope and importance in the automotive electronics domain, collaborative software development is critical to innovation.

But high-profile futuristic features are not the only place collaboration takes place in the development of vehicle electronic systems. Software-based collaboration takes place at multiple levels in the vehicle.

VEHICLE SYSTEM LEVEL

Electronic control units (ECUs) for traditional vehicle functions like engine control, transmission control, anti-lock braking, airbag control, central door locking, and lighting still require collaboration, but of a different kind than the direct collaborative development examples described in the previous section. The behaviors of the individual ECUs in the vehicle are coordinated through interaction over in-vehicle communication networks. While the developers of these traditional electronic control units do not necessarily directly collaborate, the vehicle OEM acts as a collaboration agent that works to seamlessly align the network interfaces between the ECUs and integrate them into cohesive systems. Like the explicitly collaborative efforts seen in hybrid propulsion and new infotainment systems, this looser and informal approach to collaboration is also enabled by software. These systems are bound together by the software stacks that implement the communication protocols and data exchange mechanisms that all participants on the network capitalize on to interact and perform in a coordinated way at the vehicle level.

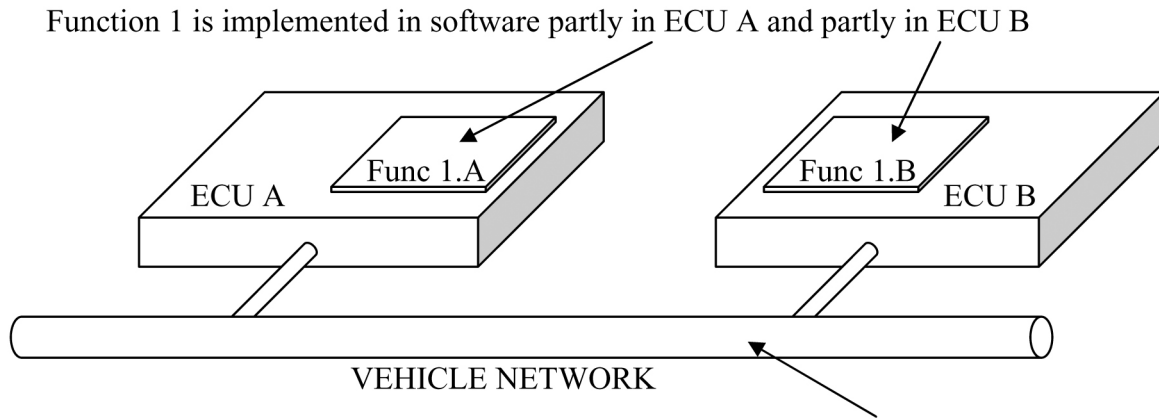
COMPONENT LEVEL

At the component level, collaboration also occurs through software, but at an even lower level inside of individual ECUs. As electronic control functions for sophisticated behavioral algorithms, complex I/O drivers, power management, network communication, flash memory drivers, and diagnostic monitors become more and more specialized, there is a natural progression toward having multiple software partners with different areas of specialization provide software components that work together as a single software system inside the ECU. In this form of collaboration, it is the ECU manufacturer that acts as the collaboration agent that seamlessly aligns the interfaces between the software components and integrates them into the ECU system. Where



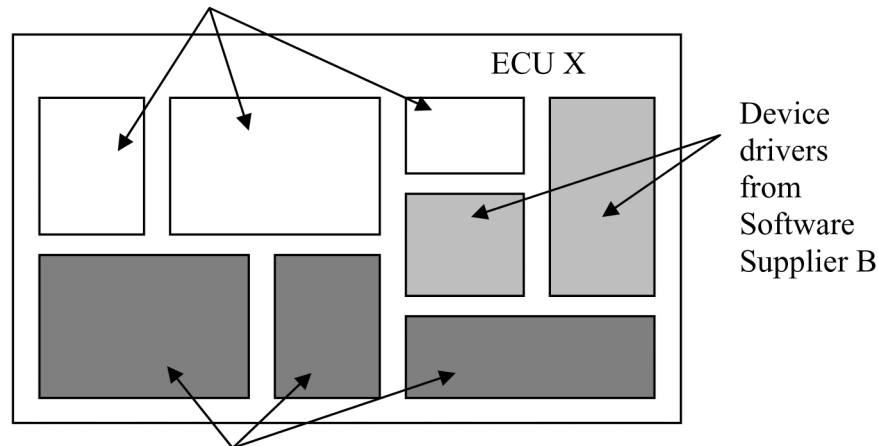
Documents

Document Sharing Hub



ECU A and ECU B coordinate by communicating over the vehicle network

Proprietary application software



Basic software from Software Supplier A

an ECU's functional behavior is considered a core competency or a competitive advantage for the vehicle OEM (as is frequently true with powertrain controls and central body/gateway controls), the ECU manufacturer is sourced to provide the ECU hardware and basic software while the vehicle OEM provides its own application software.

So, whether it is in game-changing new innovations, networked vehicle systems, or inside the ECU, software is the single biggest enabler for collaborative product development in the world of automotive electronics.

CHALLENGES TO COLLABORATIVE SOFTWARE DEVELOPMENT

Collaborative development does not just happen on its own. Collaborating partners must have a framework that allows their respective contributions to come together and provide

value greater than the sum of the parts. This framework for collaboration must effectively and efficiently harmonize these aspects of the software development process:

- software design
- software re-use
- implementation specification
- inter-organizational communication

An exploration of each of these points will demonstrate the challenges that must be addressed in order for collaborative software development to not just succeed, but to become a robust and repeatable process that consistently delivers higher value than any collaborating partner could produce for itself at a similar cost.

SOFTWARE DESIGN

Design for assembly is a well-known concept on the mechanical side of the automotive industry. Design for

assembly challenges designers of mechanical parts to look beyond the functional requirements and physical performance of their part and to consider how that part will be physically placed and fastened into the vehicle assembly. A good design for assembly can significantly improve production efficiency, reduce operating costs, and improve the bottom line for the vehicle manufacturer. In order to succeed with a design for assembly initiative, the vehicle manufacturer must orchestrate informal collaborations at many points along the product development process and across the complete supply chain.

Collaborative software development activities share many common themes with the development of mechanical systems. The software counterpart to design for assembly is “design for integration”. Design for integration asks software designers to look beyond the functional requirements and behavioral performance of their software and to consider how the software will be integrated with the other software in the vehicle system. A good design for integration approach must address several challenges in order to deliver optimal value.

The time and effort that goes into designing new algorithms that deliver innovative functionality is a necessary investment of human resources and as much art as it is science. However, given the huge impact this innovation can have on market success, this investment of resources to design new functional algorithms offers an opportunity for a huge return on investment that makes it a high-value-added activity. The key to maximizing this return on investment is finding a design approach that allows the creative new software to be easily and consistently integrated with all of the other software going into the collaboration.

Stepping back and taking a broader view, software from one partner that was not designed to interact with the software from other partners creates integration challenges that drive non-value-added rework for every partner. Defining system-specific interfaces so that all software in the system can interact with the rest of the software in the system is non-value-added work. This concept applies to the source code interfaces between software components inside a single ECU as well as to the network communication interfaces between ECUs in a vehicle system.

This scenario represents a serious challenge to the efficiency of collaborative product development and leads to a strong need for a standardized design for integration approach that the automotive electronics industry can align on. A standard design for integration approach that all partners are already using would allow all software to be quickly and easily inserted into a standardized system framework and would eliminate the need for non-value-added definition of system-specific interfaces.

SOFTWARE RE-USE

As described above, many new and innovative systems are often a new combination of existing technologies. A “design for re-use” approach can realize the highest possible value from the existing technologies brought into the collaboration by each partner. The key to maximizing this value is finding a design approach that allows existing software from all partners, both standardized and pre-standardization, to be easily and consistently re-used with all of the other software going in the collaboration.

Just as a design for integration approach asks software designers to look beyond the bounds of their own software parts, a design for re-use approach requires partners to look beyond the bounds of the current development project. A good design for re-use approach must address several challenges in order to deliver optimal value.

Looking to the past, a design for re-use approach must facilitate the integration of existing legacy software into the new system. Software that was not designed to interact with the software from other partners creates integration challenges that often limit re-use and drive non-value-added rework for every partner. Modifying the designs of existing software parts and designing new software parts so that all software components in the system are explicitly aware of each other and can interface to each other exactly as needed is non-value-added work. Further, such a here-and-now perspective serves only the current collaboration and does nothing to avoid repeating such reworks in future collaborations with different partners.

Looking to the future, a design for re-use approach must ensure that software from the current project will be inherently re-usable in future collaboration projects with different partners. This approach must include all of the legacy software repackaged for the current project as well as all new software developed for the current project. This approach must apply to the source code interfaces between software components inside a single ECU as well as to the network communication interfaces between ECUs in a vehicle system.

This scenario represents a serious challenge to the efficiency of collaborative product development and leads to a strong need for a standardized design for re-use approach that the automotive electronics industry can align on. A standard design for re-use approach that all partners are already using would allow existing software parts from previous projects to be quickly and easily inserted into new systems and would eliminate the need for non-value-added reworks of existing software that implements carry-over functionality needed in the current collaboration and also in future collaborations with different partners.