Vehicle Computers – Automotive Software Development Rethought

AUTHORS

Dr. Detlef Zerfowski
is Vice President Vehicle Computer and Security at Etas GmbH in Stuttgart (Germany).

Jürgen Crepin
is Senior Manager for Marketing Communication at Etas GmbH in Stuttgart (Germany).

Vehicle Computers are becoming more and more important in modern vehicles, alongside conventional control units. The new Autosar Adaptive standard is being developed at the same time. Both innovations will fundamentally change E/E architectures, but much more is at stake: Software development is being reorganized, and with it, value creation. OEMs and automotive component suppliers have to restructure both their technologies and their organizations in order to cope with the upheaval. In this article, Etas shows which new methods and tools are available.

CONWAY'S LAW

The introduction of the Autosar standard was hardly love at first sight. It took almost a decade for OEMs and component suppliers to fully align their development projects with the standard. Thus it may seem surprising that Bosch and Etas are already offering tools and methods for integrating the new Autosar Adaptive standard, even though standardization is not yet complete. What is the rush?

Part of the answer can be found in a 1968 essay by US computer scientist Melvin E. Conway on how organizations develop systems. His observation is now known as Conway's Law: "Any organization that designs a system will produce a design whose structure is a copy of the organization's communication structure." Put simply, every company is trapped within its own communication and organizational structures when changing systems. The company reproduces its own

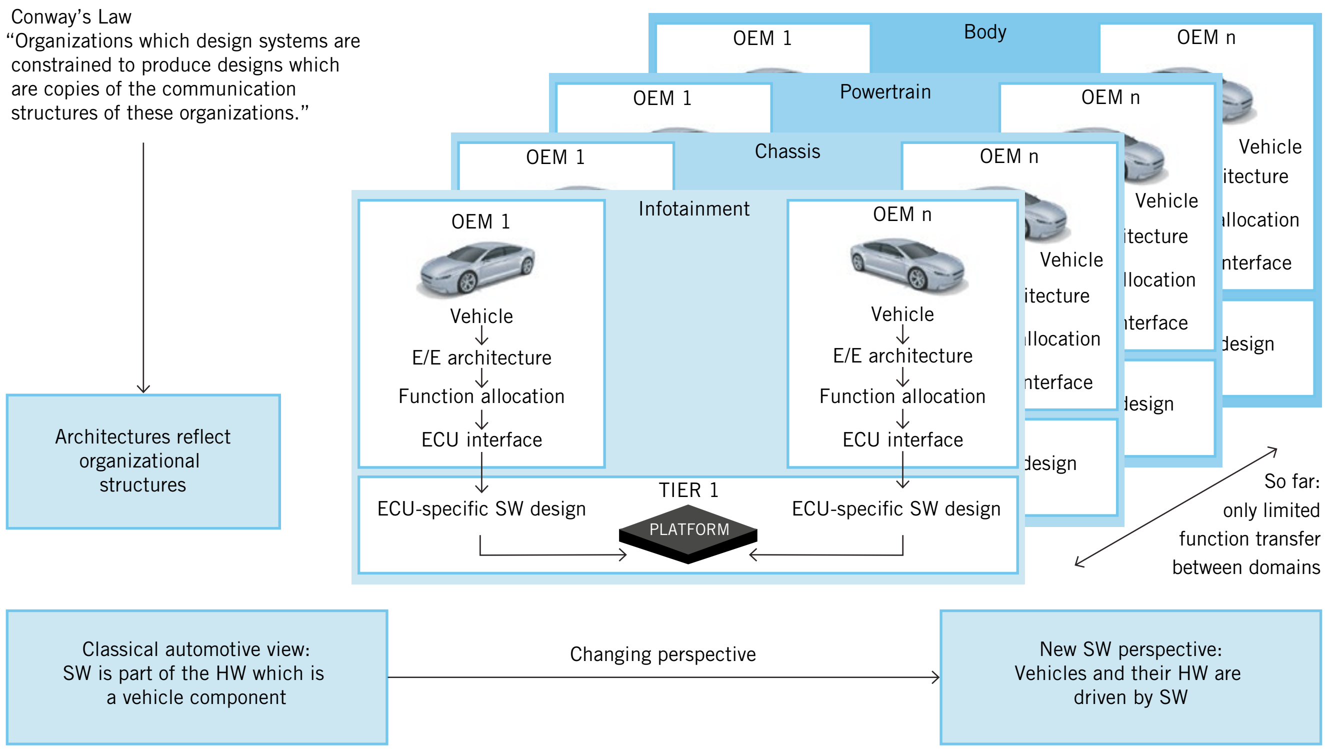


FIGURE 1 Conway's Law: The Software (SW) architecture of a system maps to its organizational structure (© Etas)

structures in terms of the system designs of its products, **FIGURE 1**. And it does so successfully because it has aligned its organizational and communication structures with market requirements. This only becomes a difficulty when a system change is imminent in the market.

THE AUTOMOTIVE INDUSTRY IS REINVENTING ITSELF

And that is exactly the case right now. The transition to electric mobility, connecting vehicles to the Internet and to cloud services, together with rapid advances in the automation of driving, are pushing existing solutions and strategies to their limits. A glance at today's E/E architectures is enough to understand this: A typical modern vehicle uses up to 120 decentralized, highly specialized Electronic Control Units (ECUs), often with similar functions resulting in redundancies. The complexity of the distributed infrastructure is an obstacle on the way to the networked, highly electronic and increasingly automated driving of the future.

The automobile as a software-dominated system, a fully networked robot that must meet the highest Automotive Safety Integrity Level (ASIL) D safety requirements, presents an enormous challenge to developers. They have to integrate an increasing number of code and differing software types while proving functional safety, Freedom-from-Interference (FfI) as well as the security of the networked systems. A comparison will help illustrate the magnitude of the task: A typical automobile already has a scope of over 100 million lines of code today, around 100 times more than in the Space Shuttle and over four times as much as in a Boeing 787.

Greater centralization of E/E architectures is the key to solving the upcoming technical challenges. Powerful and flexible Vehicle Computers (VCs) will complement conventional ECUs in the future Autosar Adaptive architectures and will support new possibilities such as Firmware updates Over-the-Air (FOTA) and connectivity.

CENTRALIZED E/E STRUCTURES WITH VEHICLE COMPUTERS

Deeply embedded ECUs based on microcontrollers (µCs) specialize in cyclical execution of clearly defined control algorithms in real time. The ECUs operate injection systems, Electronic Stability Programs (ESPs), and chassis control systems working in defined millisecond cycles with high functional reliability. Classic communication channels such as CAN and FlexRay transmit the necessary signals and support real-time implementation. The advent of VCs has fundamentally changed these architectures.

VCs employ microprocessors (µPs) or System-on-a-Chip (SoC) hardware with several CPU cores, co-processors, and powerful graphics cards, letting them perform far more complex computing operations. Unlike µC-based ECUs, VCs

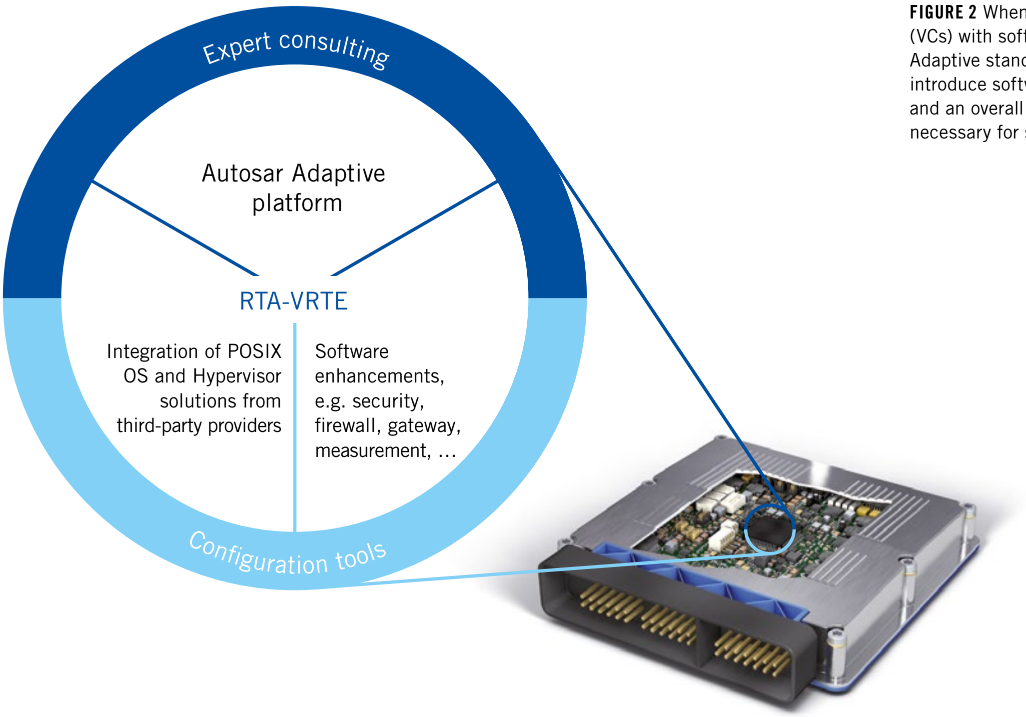


FIGURE 2 When introducing Vehicle Computers (VCs) with software according to the Autosar Adaptive standard, it is not enough just to introduce software and tools; expert knowledge and an overall view of the live-cycle are also necessary for success (© Etas)

access external memory, with communication standards such as Ethernet ensuring the necessary transfer rates. However, VCs are not real-time capable in the narrower sense. Although they are faster than µC-based hardware in most operating phases, the deterministic cyclic sequence is no longer guaranteed per se.

The increased computing power makes new, virtualized approaches possible: Hypervisors rigorously partition VCs so that each partition works as a virtual machine, ultimately assuming the function like an independent control unit. Bosch and Etas have developed an Autosar-based Vehicle Runtime Environment (VRTE) for these architectures which integrates POSIX-compliant operating systems such as Blackberry QNX or Linux in order to provide the most efficient solutions possible. This combination of IT-oriented hardware and operating systems opens the door to comprehensive system changes in automotive software development. It is precisely at this level that the system change driven by megatrends is already manifesting today. If classic automotive players do not immediately make every effort to internalize the new systematics related to VCs and modern software development methods, they will remain trapped in their historically evolved structures, as predicted by Conway's Law.

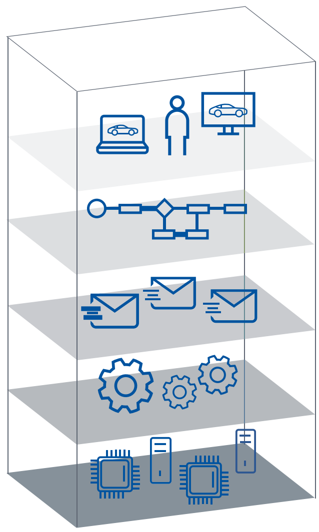
NEW BEGINNINGS IN SOFTWARE DEVELOPMENT

Today software development follows a fixed pattern: OEMs plan the functions, the allocation and networking of the ECUs in the vehicle, derive specific software designs from these factors, and pass them on for implementation to Tier-1 suppliers who in turn adapt their own ECU platforms to match OEM specifications. The Tier-1 suppliers develop the hardware and the complete software range, including application software from various specialty areas, to which Etas in many cases contributes the RTA-OS operating system and the RTA-BSW basic software.

This established business model is fundamentally changing for connected, automated mobility concepts, since architecture with encapsulated virtual machines and central operating systems makes it possible to completely separate software development from hardware development. Software from the widest possible variety of vendors can run on the encapsulated partitions. In the future OEMs will be able to purchase software separately. In addition, the integration of vehicles in the Internet of Things (IoT) is behind another upheaval: The ability to update and upgrade over-the-air at any time and to install after-market apps is driving the introduction of new agile software development processes and methods. These methods are pushing the standardized V model back into those areas in which very high safety and security requirements require retention of µC-based Autosar Classic methodologies.

ENORMOUS MARKET POTENTIAL AND HIGH RISKS FOR TRADITIONAL BUSINESS MODELS

In the future VCs will simultaneously run software from dozens of different vendors. While the amount of ECU software has been around 8 MB up to now, it will rise in the future to as much as 80 GB in VCs, increasing by a factor of 10,000. Typical automotive players (OEMs, Tier-1 and other component suppliers) are likely completed by new partners, often from outside the automotive industry. But even if the established automotive players only deliver a part of the software, this will still be much more code than before.



Application services	Functions/applications
Layer 5 Vehicle-dependent platform services	Services managing the ECU grid of the vehicle
Layer 4 ECU-dependent platform services	Services managing one specific ECU
Layer 3 Communication middleware (service-oriented)	Manages control and data flow between software components
Layer 2 OS-dependent infrastructure software	Software that complements the actual OS kernel (aka scheduler) and abstracts OS-specific properties towards higher layers
Layer 1 Hardware-dependent infrastructure software	Software that interacts directly with hardware and abstracts it towards the higher layers
Hardware	Microcontroller (µC), microprocessor (µP), virtual machine (VM)

FIGURE 3 Layers of the Etas RTA-VRTE software framework (© Etas)

In order to exploit these opportunities, automotive players will have to quickly adapt their expertise and structures to the new market system. This calls for business units that exclusively develop and sell software: production without a manufacturing infrastructure. This will challenge the automotive suppliers and will pit them against new, agile competitors from the IT sector. The issue is nothing less than the redistribution of value creation. Conway's Law seems like a dark shadow. By now the answer to the initial question "What's the rush?" is clear.

FAMILIARIZATION WITH THE NEW AUTOSAR ADAPTIVE METHODOLOGY

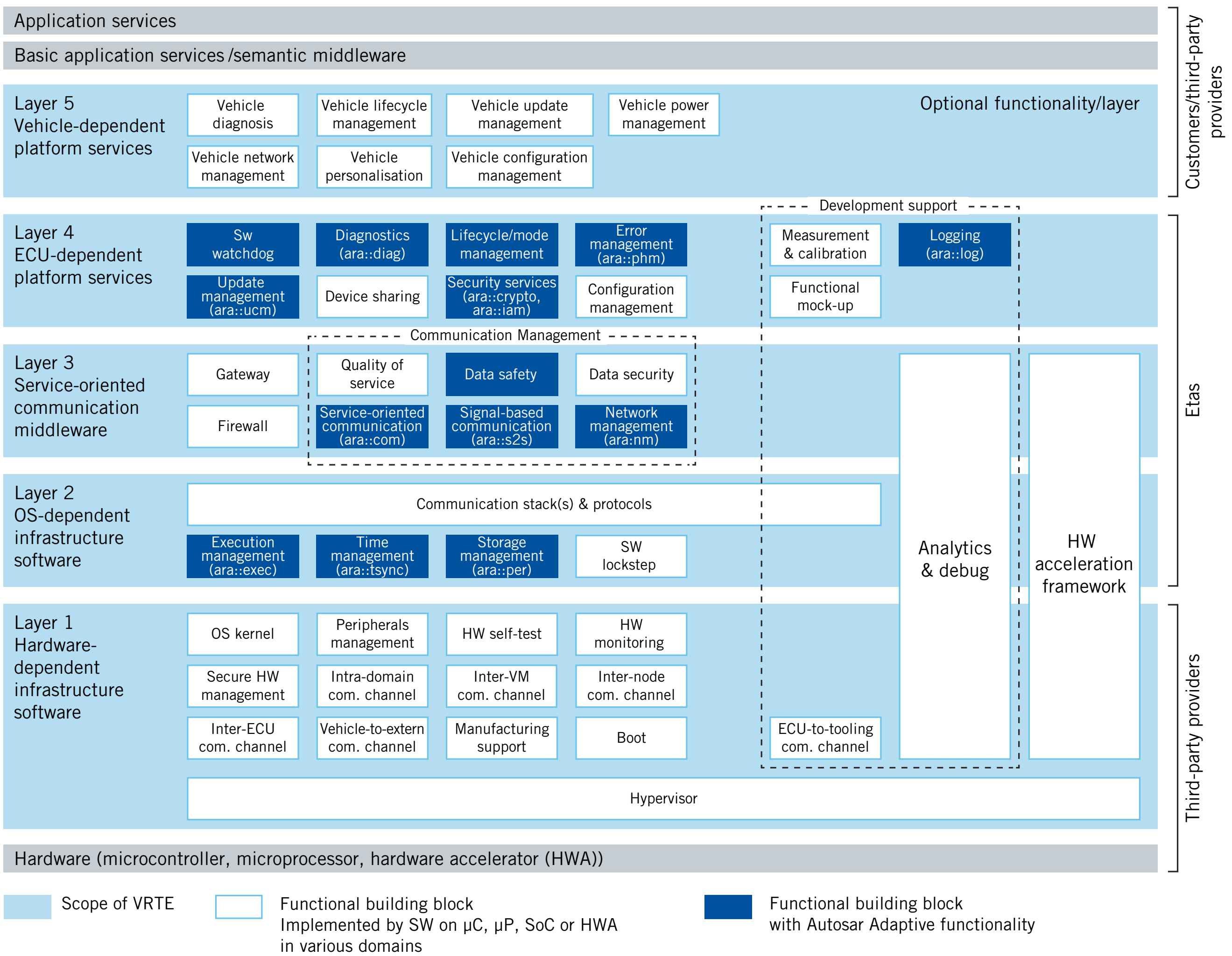
The new Autosar Adaptive standard is still being developed, so there is still time to prepare for the impending paradigm shift. This is precisely why Bosch and Etas have set up the Autosar-based VRTE (Vehicle Runtime Environment) platform software framework and are now offering customers an Early Access Program which uses previous progress in standardization as a foundation for providing developers with training, consulting, and a "ready-to-go" Software Development Kit (SDK). This lets customers immediately start prototyping with VC architectures. The modular SDK features a platform software framework and configuration tools, **FIGURE 2**. The same high quality standards of the Autosar Classic platforms are also applied to this early implementation of Autosar Adaptive.

The new E/E architectures with a partitioned VC and POSIX operating systems require powerful communication middleware. As part of the Early Access Program, this middleware controls and regulates interaction among the various levels and thus guarantees the smooth operation of the encapsulated software. Users can test the new architectures in detail and can develop and debug their own operational prototypes. In addition, software solutions can be integrated that are not (yet) compliant with Autosar Adaptive, whether firewalls or gateway management solutions from the security sector. Users now have the flexibility they need to bundle software from a wide variety of sources to form a reliable, functionally secure whole. The VRTE already offers early starters a solid foundation for configuration, which Etas brings to life with the Real-Time Application Vehicle Runtime Environment, or short RTA-VRTE.

GLOBALLY DISTRIBUTED, COMPLETELY VIRTUALIZED DEVELOPMENT PROCESS

The partitioning of VCs into virtual machines paves the way for highly parallel and completely virtualized software development processes. Developers can already learn their methodologies with the RTA-VRTE now, since as a multi-layer platform, the RTA-VRTE is completely decoupled from the VC hardware that will be used later, regardless of whether the hardware is from NXP, Renesas, Qualcomm, Nvidia, or Intel. Pre-installed virtual machines (x86_64 bits and Armv8) are preconfigured using hypervisors. In the RTA-VRTE these virtual machines function as virtual ECUs, which developers can simulate on conventional desktop PCs. The ECUs are networked via Ethernet and can therefore communicate with each other. Finally, Etas provides users of the Early Access Program with a complete Virtualbox image of the VC featuring a five-layer VRTE architecture, **FIGURE 3** and **FIGURE 4**. This gives users uncomplicated access to the completely virtualized development environment, which they can test on their own PCs. Virtualbox is a virtualization solution for PCs from vendor Oracle.

FIGURE 4 Software components of the Etas RTA-VRTE software framework (© Etas)



SUMMARY

Bosch and Etas now offer users the opportunity to familiarize themselves with the logic and methods of the new standard, parallel to the Autosar Adaptive standardization project, in the RTA-VRTE and the Early Access Program. The potential goes well beyond familiarization with modified tool chains. Software development teams can practice new communication structures within their companies, the prerequisite for establishing agile development processes for VC software. This also calls for overcoming structures that were in the past strictly divided according to domains. Companies will lack the flexibility to leverage the opportunities of the changes as long as functional areas stick to their accustomed technical terms and mindsets and have difficulties communicating with other departments. However, those who embrace the new processes and collaborative methods and practice them with the new virtualized tool chain according to Autosar Adaptive can achieve a perfect point of departure for the redistribution of added value as an early starter. Time is of the essence, but it is not too late yet.