

AUTOSAR on its Way to Production

To master the growing complexity of software in modern vehicles, automotive OEMs are increasingly developing their electronic systems based on AUTOSAR. The standards created by this development partnership simplify development processes and make ECU software reusable. Since its introduction in 2004, this innovative and pioneering technology has been tested in many evaluation projects and is now entering an implementation phase in production ECUs. AUTOSAR standard software covers the current state of technology and is undergoing continual advanced development in new releases.

The automotive industry is being confronted with new times. Growth in the number of complex vehicle functions is making the development of automotive electronics increasingly more extensive and complicated. Customer wishes, e.g. for more variants and equipment variety in the vehicle, as well as non-functional requirements such as diagnostic capability and availability, are requiring more intensive ECU development processes. Vehicles, in particular luxury vehicles, may have more than approx. 1,000 software functions, several vehicle electrical networks and more than 70 ECUs. Due to the variety of hardware platforms used in this field, ECU software dependencies on the hardware and system configurations being used has become problematic. Each change to one of these constraints requires reprogramming or at least modification of the software.

To make the complexity of ECU software development manageable, the AUTOSAR development partnership has defined a practice-tested software architecture that serves as a foundation for developing reusable applications. This open standard for system architecture – created by automotive OEMs, suppliers and other companies in the global software, semiconductor and electronics industries – lets users avoid proprietary solutions that would result in increasingly high development cost and effort.

AUTOSAR subdivides the electronics architecture into a number of layers and modules. With the simultaneous definition of interfaces, AUTOSAR creates standards for easy exchange of software components or hardware platforms. The development partnership not only provides specifications for the base software modules. It also offers a methodology for developing applications and distributed systems. This methodology begins with a model-based description of the software applications and the distributed system, and ideally ends in an automatically generated and reproducible test. This formal approach simplifies the use of a universal tool chain.

A full three years after its start, the AUTOSAR development partnership published Release 2.1 at the beginning of 2007. A stable level was reached with this release, and it has been tested in several validation projects for its practical utility. At many businesses, the action item of “AUTOSAR evaluation” has been successfully completed. Now it is ready to be implemented in concrete production ECUs.

AUTOSAR Architecture

To realize the objectives of AUTOSAR – separation of the application software from basic modules and functions – the vehicle electronics

is abstracted and subdivided into several layers (Figure 1). The connection to the actual microcontroller, i.e. the physical foundation, is represented by the Microcontroller Abstraction Layer, which maps the microcontroller’s functions and periphery. It defines interfaces for memory, the I/O driver and its communication connections. In this layer, features that the microcontroller does not offer may also be emulated in software. The second layer that lies above this is the ECU Abstraction Layer. It establishes die ECU-specific hardware layout and provides drivers for the periphery external to the ECU, for example. In another layer, the Services Layer, various basic services are represented such as network services, memory management, bus communication services and the operating system. This layer is already largely independent of the hardware. The RTE represents the fourth layer. This is where the actual separation of application and basic software (BSW) is made. The RTE handles integration of the application software and regulates the exchange of data between the application and the BSW. It is only at this layer that reuse of already written application software is possible, because the defined interfaces make it possible to develop the application software without special knowledge of the hardware to be used at a later time, and the software can be used for any other AUTOSAR-conformant ECUs.

The so-called Virtual Functional Bus (VFB) forms the basis for configuration of the layers. All components of the vehicle’s electronics communicate in an abstracted view via this bus. The software components use ports for this, whose interfaces can be defined as port interfaces. Connectors connect the ports. It is irrelevant to the VFB whether this is a connection within an ECU or a connection via an external bus. This is not decided until the final system layout is made and specific functions are assigned to a specific ECU.

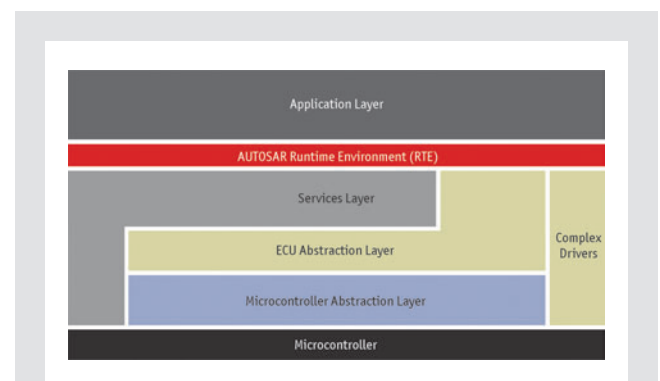


Figure 1:
AUTOSAR layer model for ECU software.

The software component itself does not require any knowledge of this later distribution, and it can therefore be developed independent of it. The component is subdivided into executable units, so-called Runnable Entities, which are executed like procedures when a specific event occurs. Such an event might be the arrival of a new sensor value or a periodic activation, for example. The description of the electronic system formulated from the perspective of the VFB defines the interfaces of the specific software components. As a result, the application components can be developed independent of the specific ECU.

The “counterpart” on the ECU is the RTE. It guarantees access to I/Os, memory and other basic services. Using the model-based description, the RTE is generated ECU-specifically, which means that it can be adapted to different requirements while economizing on resources.

Methodology

Besides defining the ECU’s software architecture, the AUTOSAR standard also defines a methodology to help in the development of AUTOSAR systems. Conformance to a structured creation process is recognized as an important precondition in creating error-free software. Deficiencies in the requirements list are detected early, reuse and porting of software components is simplified, and the overall system is more reliable. Nonetheless, this methodology also allows certain freedoms: for example, users can decide whether to use a top-down approach or a bottom-up development.

The AUTOSAR Initiative provides universal support for the software development process by tools. Mature tools are used for structured implementation of requirements and their consistent management; configurations are systematically created, and complex dependencies are automatically taken into account.

The first step involves formal description of the three primary constituents: Software (SW Components), ECUs (ECU Resources) and System Constraints.

Suitable editors are used to create a configuration description of the complete system (Figure 2). This system configuration serves as the basis for the ECU Configurations that the user creates for the individual basic software modules with the help of configuration tools. At the end of the process, multiple generators supply the ECU-specific implementation of the RTE and basic software.

All design and configuration data produced in the development process are described in a uniform format. AUTOSAR has defined an XML-based format for this purpose. On the one hand, it guarantees universality of the development process, and on the other it also simplifies seamless integration of necessary and auxiliary tools.

Migration

The software architecture of an AUTOSAR ECU is not monolithic, rather it consists of a number of standard modules with well-defined interfaces. This enables implementation of migration solutions that offer a risk-free transition to AUTOSAR. Such a migration solution must typically be worked out project-specifically. In prac-

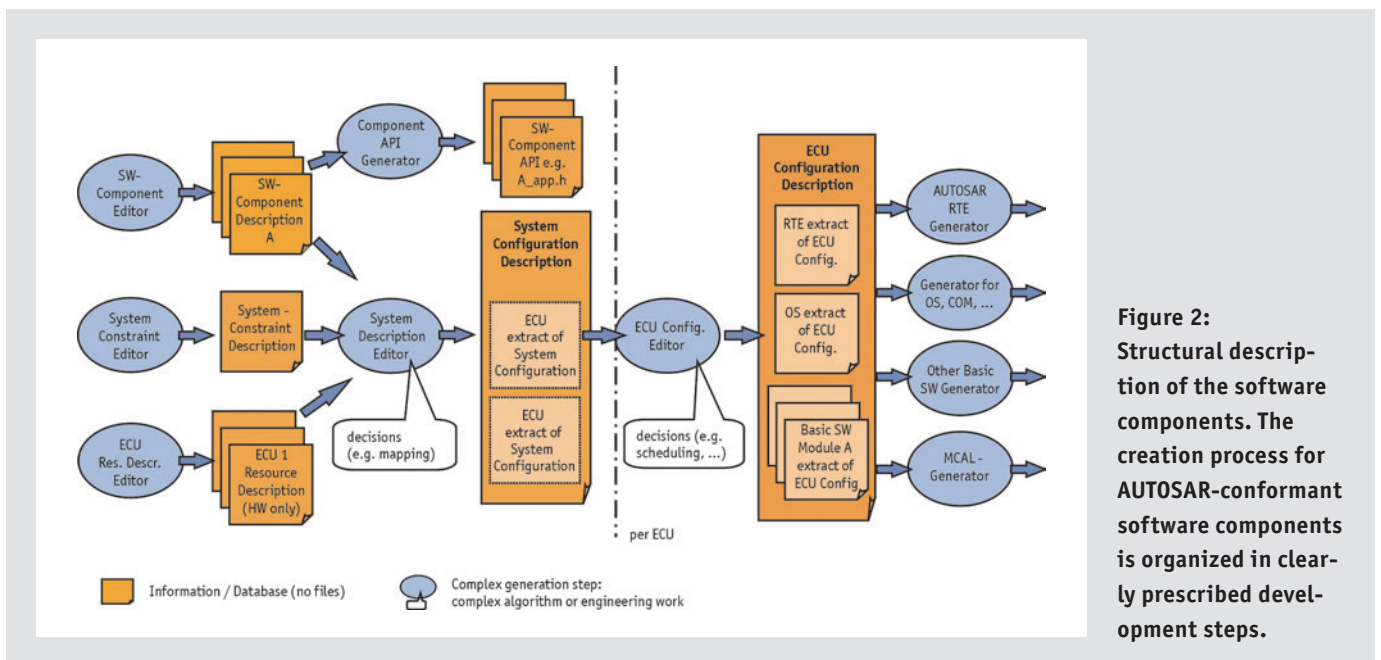


Figure 2: Structural description of the software components. The creation process for AUTOSAR-conformant software components is organized in clearly prescribed development steps.

tice, this may involve a mixed configuration of standard AUTOSAR modules and proprietary software.

To work out a migration solution, one begins by comparing the existing custom software and the AUTOSAR architecture. After an analysis of overlapping functionalities and integration options, a decision is made regarding which modules will remain and which ones can be replaced by standard software.

So, it is advisable, for example, to introduce a separation layer between the application and basic software. A potential scenario in this case is to prepare the applications as AUTOSAR software components already in early migration phase, and to integrate them via an RTE. Beneath the RTE, a specific adaptation layer is used for interfacing to the existing basic software (Figure 3).

If parts of the existing basic software are to be replaced by AUTOSAR basic software, the emphasis should lie on the use of a uniform configuration tool for both worlds. Vector offers suitable tools

here, which are flexible enough to be used to configure proprietary basic software too. Non-AUTOSAR modules can now be replaced by AUTOSAR modules in successive steps, without putting the overall architecture at risk or requiring reprogramming to other modules.

Outlook

The current AUTOSAR Release 3.0 represents another step taken to enhance the quality of the AUTOSAR standard. The continuing involvement of participating companies clearly demonstrates commitment to the goals of the AUTOSAR development partnership. Ideas are currently being introduced that should become a reality in the framework of the future Version 4.0 of the AUTOSAR standard.

New ideas related to AUTOSAR are also being generated by tool suppliers. Current developments at Vector are aimed at making AUTOSAR-based ECU development as convenient and efficient as possible. One example is a PC-based test tool for AUTOSAR application components, which serves as an emulator for the AUTOSAR ECU environment on the level of the VFB. This makes it easy to test the implementation code of the AUTOSAR software components on a development PC. Widely used standard tools such as Vector's CANoe may be used for test execution, visualization during or after testing, and generation of test reports. Vector supports all phases of the ECU development process with its full set of AUTOSAR basic software and a universal tool chain of design and development tools (Figure 4). The available Vector solution has been tested in practice through its use in evaluation projects, and it is production-mature for AUTOSAR Release 2.0 or 2.1 (3.0 too starting in the 2nd quarter of 2008).

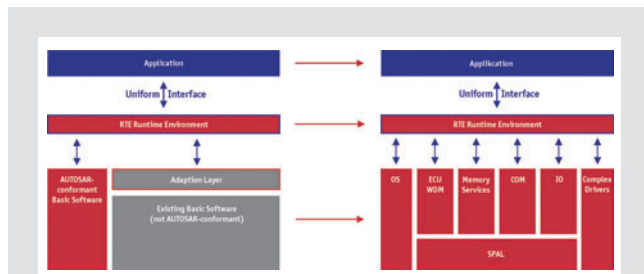


Figure 3:
Early introduction of a uniform interface and RTE simplifies migration.

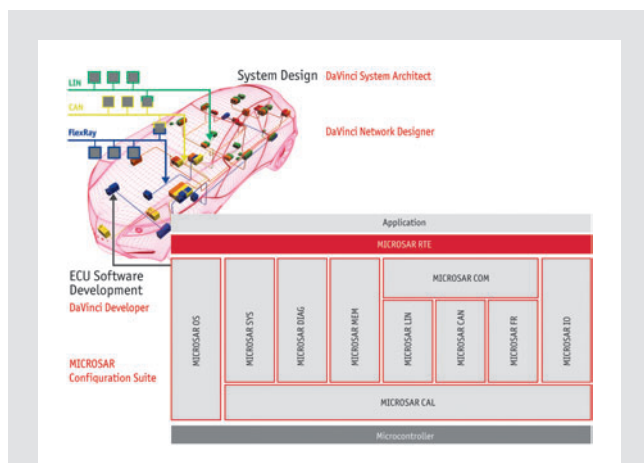



Figure 4:
The Vector AUTOSAR solution: From system description to standardized ECU software.

Summary

AUTOSAR is becoming a reality. Various OEMs have concrete plans for implementing AUTOSAR in upcoming vehicle production programs. Vector offers a comprehensive product lineup for this, as well as basic software and tools related to AUTOSAR. Not only does this enable the development of pure AUTOSAR systems; it can also support a stepwise migration toward AUTOSAR.



Matthias Wernicke
(Graduate engineer) is responsible for product management of the DaVinci Tool Suite and is also actively engaged in AUTOSAR standardization work.