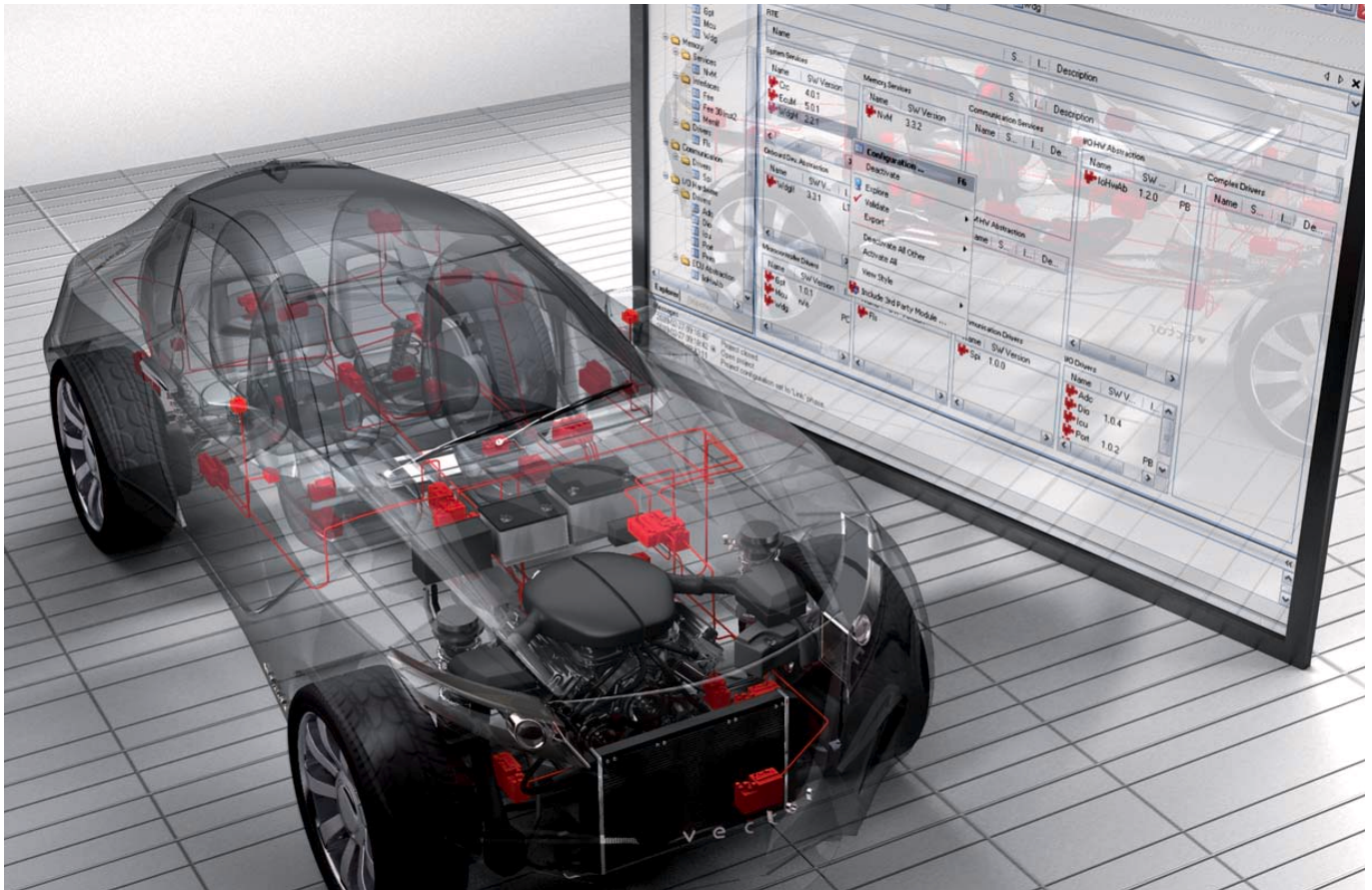


AUTOSAR: New Paths to ECU Software

Iterative collaboration between OEM, TIER1 and software supplier



A primary reason for introducing AUTOSAR, besides standardizing the basic software, is to increase reusability of the functional software. This affects the cooperation between the partners involved as well: OEM, TIER1, semiconductor manufacturer and software supplier. This first part of a two-part series describes a foundation for successful collaboration: AUTOSAR-specific exchange formats and tools. In the second part, you will learn about the significance of AUTOSAR for everyday work in developing ECU software for production projects.

Each OEM has its own functional requirements for the ECUs in its vehicles, especially when it comes to communication and diagnostics. These requirements are implemented in OEM-specific software. If a TIER1 supplies an ECU to several OEMs, it must manually modify the ECU software for each project. Even if the functional software is already decoupled quite well from the hardware, so that it can be adapted to the OEM-specific requirements, this modification effort is still work intensive and prone to errors. Figure 1 shows how unmodified functional software is adapted to different vehicle projects without AUTOSAR.

One goal of AUTOSAR is to minimize these adaptation efforts in software integration. Therefore, AUTOSAR focuses on consistent abstraction of the software from the hardware and partitioning of the software into modules with defined functional scope and precise interfaces. These modules may be combined and, most significantly, they can be substantially configured to cover the requirements of different OEMs. This eliminates manual modification of

the software and facilitates its reuse. Defined interfaces make it possible to replace OEM-specific software components (e.g. for diagnostics) with just minimal effort.

AUTOSAR reference architecture

The AUTOSAR reference architecture is described in the document AUTOSAR Layered Software Architecture [1]. In this document, the ECU software is organized into the three parts shown in Figure 2:

- > The functional software consists of software components (SWCs). The SWCs are created, independent of the ECUs, based on a Virtual Function Bus (VFB), and they communicate with one another via interfaces.
- > The Runtime Environment (RTE) is used for executing the SWCs and it includes the technical implementation of the VFB in a real ECU.

> The basic software (BSW) modules handle the basic functions of an ECU. They also offer higher-end standard services such as management of ECU states and diagnostic services.

The RTE is the layer between the functional software and the basic software modules. It provides all interfaces the SWCs need to access data and services of the BSW modules. Examples are signal values from the communication network (CAN, LIN or FlexRay), I/O signals and standard services of the BSW modules. The interfaces originate from the "SWC Description" files. Moreover, the RTE handles execution of the SWCs and communication among the SWCs with the help of the operating system.

The BSW modules are subdivided into three layers per the AUTOSAR architecture [1]:

- > Service Layer
- > ECU Abstraction Layer
- > Microcontroller Abstraction Layer (MCAL)

The BSW modules of the Service Layer play a special role here, because they contain standard services for the functional software that are accessed via special interfaces within the RTE. The second part of this article, in the next issue, will describe the configuration of these services in greater detail.

The AUTOSAR Release 3.0 defines approx. 50 different configurable basic software modules; some of them are very complex. The majority contains functions that were already usually present in previous software architectures, but now they are more precisely distinguished from one another. Consistent partitioning of

functions into individual software modules is what assures the desired hardware abstraction and scalability for different types of ECUs. The use of such standard modules increases the quality of the ECU software. In most cases, this standardization covers the interfaces as well as functions of the BSW modules. Representing an exception here are the BSW modules for diagnostics. Since diagnostic processes are very dependent on manufacturing and after-sales processes at the OEMs, AUTOSAR only defines the interfaces of the diagnostic modules. This allows OEM-specific implementations of the diagnostic modules. Vector provides these modules for many OEMs, and it is the task of the TIER1 supplier to configure and integrate the specific variant.

Both the BSW modules and the RTE are available as software products from various software suppliers (TIER2), such as the MICROSAR products from Vector, which offer coverage of all BSW modules and the RTE per AUTOSAR Release 3.0. Although they are standard software products, the BSW modules and the RTE must be adapted to project-specific constraints (OEM, vehicle model, ECU variant). This involves use of relevant PC-based tools during the configuration process. For example, the RTE may be configured with DaVinci Developer and the BSW modules with DaVinci Configurator Pro from Vector.

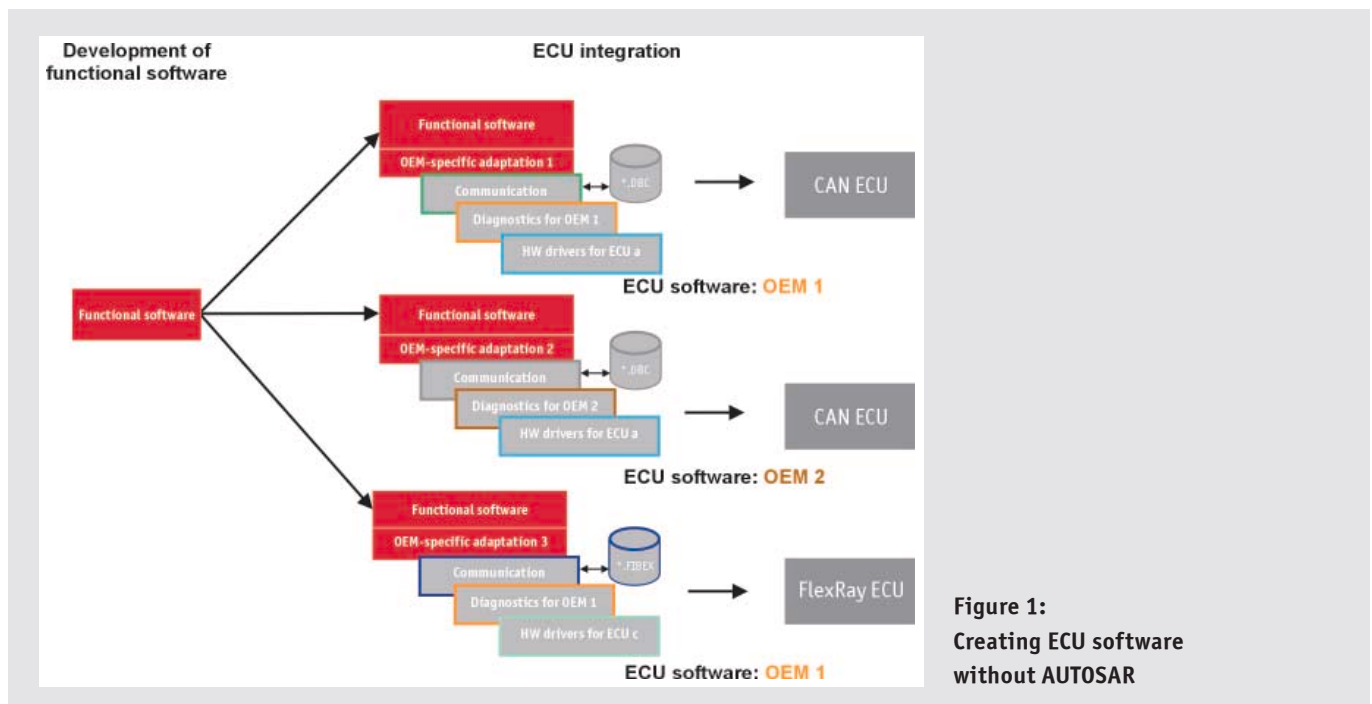


Figure 1:
Creating ECU software
without AUTOSAR

AUTOSAR Methodology

The AUTOSAR Consortium has defined a method for developing ECU software, the AUTOSAR Methodology [2]. This document essentially subdivides the development process into three activities and standardizes data exchange between development partners with a set of XML files:

> Activity: “Component implementation”

The TIER1 or OEM defines the SWCs. For this purpose, it creates an XML file for each SWC, the so-called “SWC Description”. This file describes the SWC’s interfaces and resource requirements. Afterwards, the TIER1 or OEM creates the related C files for the implementation of the SWC.



> Activity: “System configuration”

The OEM first defines the functional scope of the entire vehicle based on the SWCs, independent of the ECUs. The next step is to design the communication networks and distribute SWCs to the available ECUs. The result is saved in the “System Description”.



For each ECU, the OEM reduces the “System Description” to an “ECU Extract of System Description” which the OEM can pass to the supplier (TIER1) of the relevant ECU. This file replaces the DBC, FIBEX or LDF files previously used to configure the BSW modules.



> Activity: “ECU design and configuration”

Starting with the “ECU Extract of System Description”, the TIER1 integrates its own SWCs. This results in a complete and up-to-date “ECU Extract of System Description”, which now contains the description of all SWCs (from OEM and TIER1) of an ECU.



Another prerequisite for ECU configuration is the existence of the “BSW Module Description” files, which contain the definition of the data structures and all configurable parameters of a BSW module. These files are implementation-specific and – along with the generators and the static code – are part of the BSW modules.



Afterwards, the TIER1 creates the initial “ECU Configuration Description” (activity 2 in Figure 3) based on the current “ECU Extract of System Description” and the “BSW Module Description” files. Then the TIER1 begins to configure the ECU and documents it in the “ECU Configuration Description”. The TIER1 uses suitable tools for configuring and checking parameters of the BSW modules and the RTE for this purpose (activities 3 and 4 in Figure 3). The “ECU Configuration Description” is the foundation for ECU-specific generation of the RTE and the BSW modules by the relevant generators.

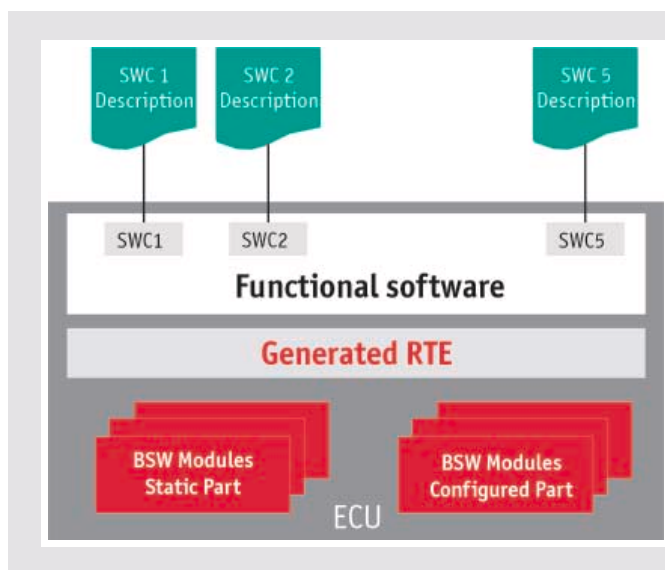


Figure 2: AUTOSAR architecture of an ECU

The AUTOSAR method is flexible and suitable for satisfying the practical requirements of different projects or different OEMs. For example, use of SWCs in the “System Description” is optional.

Figure 3 shows – based on the example of the tools DaVinci Developer and DaVinci Configurator Pro from Vector – how the “ECU design and configuration” activity can be implemented with tool support.

Configuring and integrating all software components

During the configuration process defined in AUTOSAR, the TIER1 selects – from its component collection – those SWCs it needs for the ECU’s functionality. Afterwards the TIER1 integrates them in its ECU, together with the BSW modules and RTE. This shifts the primary work of integrating the ECU software from manual code adaptation to tool supported configuration of the BSW modules and the RTE.

Since the current level of the AUTOSAR specifications still has some room for interpretation, from today’s perspective it is advisable to procure either the entire BSW package, or at least defined BSW clusters, from a single source. The advantage is that the software supplier (TIER2) can already perform an integration test on these modules. However, it is also possible to procure individual modules from different TIER2 suppliers or use modified BSW modules. This increases integration effort, however, since both functional integration and integration in the configuration tools need to be performed.

Essentially, all MICROSAR BSW modules are tested within systematic integration tests. As an integration partner, Vector can extend its integration services to software modules from third-party producers, such as MCAL drivers, upon request.

For configuration of the BSW modules, the TIER1 needs the support of a universal tool with user-friendly functions. That is why Vector developed DaVinci Configurator Pro. It supports three use cases:

- > Configuration of MICROSAR BSW modules from Vector
- > Configuration of AUTOSAR BSW modules from third-party manufacturers
- > Configuration of software modules you have created yourself

MICROSAR BSW modules are configured by using a graphical user interface that shows the complex interrelationships of the configuration parameters in simplified form. Furthermore, parameter selection is limited to valid input values, and this prevents setting implausible values.

The Generic Configuration Editor (GCE) defined in AUTOSAR is included with DaVinci Configurator Pro to configure the BSW modules from third-party producers. As an alternative, the TIER1 supplier may choose to develop a user-friendly and integrated configuration user interface for these modules itself. This may also be done with the newly developed DaVinci Configurator Kit. It is used to create “BSW Module Description” files for the software modules, define user-friendly user interfaces, establish validation rules and create code generators for generating the executable code. The TIER1 can also use this approach to configure its own BSW modules, e.g. complex device drivers.

Both DaVinci Configurator Pro and DaVinci Developer contain validation rules that supplement the AUTOSAR method. They ensure that individual parameters as well as complex parameter groups and their interdependencies are validated and that the “ECU Configuration Description” is generated consistently. This consistency is essential for subsequent generation of the RTE and the BSW modules.

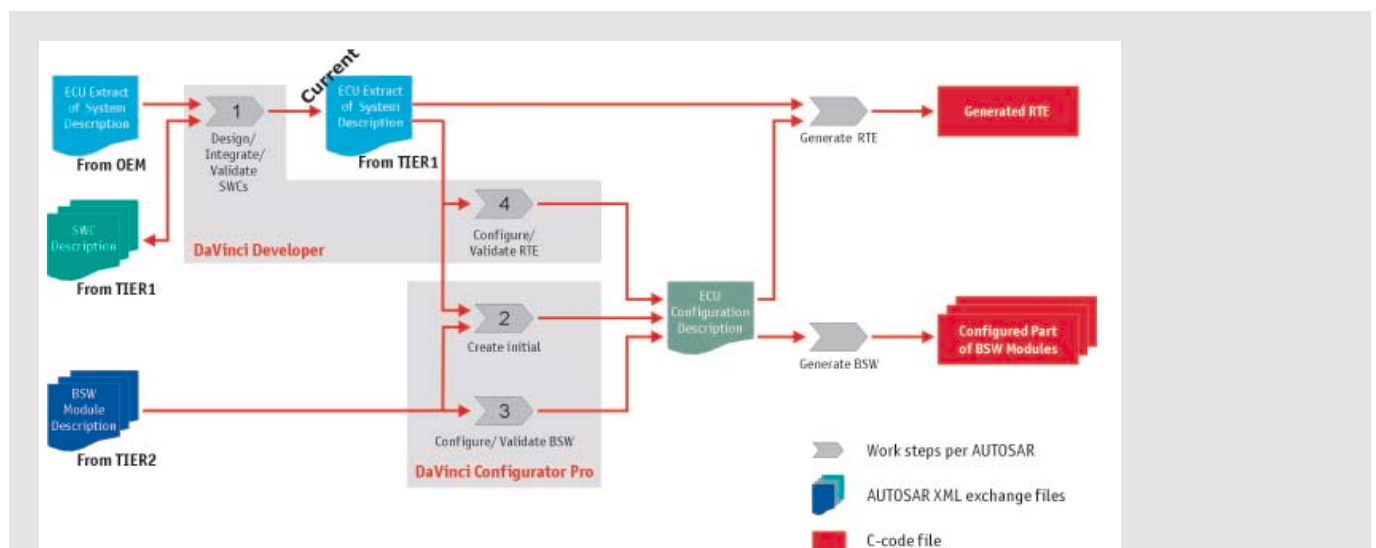


Figure 3: Tool-supported integration of SWCs and configuration of RTE and BSW modules per AUTOSAR methodology

In the second part of this article, you will learn – based on examples of selected use cases – how the exchange files and configuration tools are used in practice. The process of creating a complete set of AUTOSAR-conformant ECU software for a specific OEM is explained, and the article describes how to maintain the software over time or modify it for a different OEM.

**Translation of a German publication in
Elektronik automotive, 11/2009**

All Figures:
Vector Informatik GmbH

Literature:

[1] Layered Software Architecture:
http://www.autosar.org/download/specs_aktuell/AUTOSAR_LayeredSoftwareArchitecture.pdf

[2] AUTOSAR Methodology:
http://www.autosar.org/download/specs_aktuell/AUTOSAR_Methodology.pdf

Links:

Homepage Vector: www.vector.com
Product information about AUTOSAR: www.autosar-solutions.com



Pascale Morizur (Dipl.-Ing.)
studied Physics-Electronics at the Grande Ecole ICPI in Lyon (France). After graduating in 1986, she worked for 10 years in advanced development for CAN, J1939 and diagnostics at MAN Commercial Vehicles. Since 2005, she has been employed at Vector as Product Manager in the area of Embedded Software Components.
Tel. +49 (0)711/80670-2211,
Fax +49 (0)711/80670-111,
E-mail: pascale.morizur@vector.com



Matthias Wernicke (Dipl.-Ing. (FH))
upon graduating in Industrial Electronics at the Polytechnical College at Ulm, was employed for four years at the Daimler Research Center in Ulm, Germany. Since early 2000 he has been working at Vector Informatik in Stuttgart, developing methods and tools for the design of distributed electronic functions in motor vehicles. Today he is responsible for product management of DaVinci AUTOSAR tools.



Justus Maier (Dipl.-Inf. (FH))
studied Computer Science in Regensburg. He began his professional career as a developer of standardized software in the insurance industry. For 8 years he was involved in design and advanced development of tools for ECU configuration in the AUTOSAR field. He has been employed at Vector since 2006 as technical Product Manager for DaVinci Configurator Pro.
Tel. +49 (0)941/20865-451,
Fax +49 (0)941/20865-111
E-mail: justus.maier@vector.com

>> Your Contact:

Germany and all countries, not named below

Vector Informatik GmbH, Stuttgart, Germany, www.vector.com

France, Belgium, Luxembourg

Vector France, Paris, France, www.vector-france.com

Sweden, Denmark, Norway, Finland, Iceland

VecScan AB, Göteborg, Sweden, www.vector-scandinavia.com

Great Britain

Vector GB Ltd., Birmingham, United Kingdom, www.vector-gb.co.uk

USA, Canada, Mexico

Vector CANtech, Inc., Detroit, USA, www.vector-cantech.com

Japan

Vector Japan Co., Ltd., Tokyo, Japan, www.vector-japan.co.jp

Korea

Vector Korea IT Inc., Seoul, Republic of Korea, www.vector.kr

India

Vector Informatik India Prv. Ltd., Pune, India, www.vector.in

E-Mail Contact

info@vector.com

