

AUTOSAR: New Paths to ECU Software

AUTOSAR in Practice – Life Cycle of AUTOSAR ECU Software

The first part of the article covers the structure of AUTOSAR-conformant ECU software, the AUTOSAR development method and helpful auxiliary tool support. The second part of the article presents realistic scenarios illustrating how AUTOSAR ECU software is maintained over its life cycle.

During the development of an AUTOSAR ECU, code generators are used to adapt the basic software (BSW) and runtime environment (RTE) to specific ECU requirements. Generation is based on the following extensive description files mentioned in the first part of the article:

- > The “ECU Extract of System Description” contains the ECU-specific section of the overall system.
- > The “SWC Description” describes the interfaces and structure of the AUTOSAR software components (SWCs), and it may already be included in the “ECU Extract of System Description”.
- > The “ECU Configuration Description” contains the configuration of the BSW modules and RTE.
- > The “BSW Module Description” describes the configurable parameters of a BSW module.

The challenge for the developer is to keep these description files consistent while avoiding the need to recreate the whole configuration whenever a change is made. In performing this task, it is helpful to use configuration tools that support iterative work processes during an ECU development project. Typical scenarios and change cases are described as well as having work steps explained based on the Vector tools DaVinci Developer and DaVinci Configurator Pro.

New development of software components (SWCs)

Depending on the OEM, the “ECU Extract of System Description” might already contain the interfaces of SWCs. In some cases, these SWCs are still incomplete. For example, OEM-specified application interfaces may be included, but not the implementation description (runnable entities) or interfaces to the standard services of the BSW modules. The Tier1 developer can add this missing specification with the help of a tool such as DaVinci Developer. The Tier1 may also create its own SWCs (Step D1 in Figure 4).

There are two different ways to implement the SWCs: either manually based on a generated code template or with the help of model-based development (MBD) tools such as MATLAB®/Simulink® EmbeddedCoder® or TargetLink. The “SWC Description” is imported into the MBD tool where it serves as a basis for modeling the SWC. Code generators are used to automatically generate SWC implementations.

If SWCs are already available at the Tier1, they can also be integrated in the ECU. This involves importing the relevant “SWC Description” and linking the SWC to the other SWCs of the ECU (Step D2).

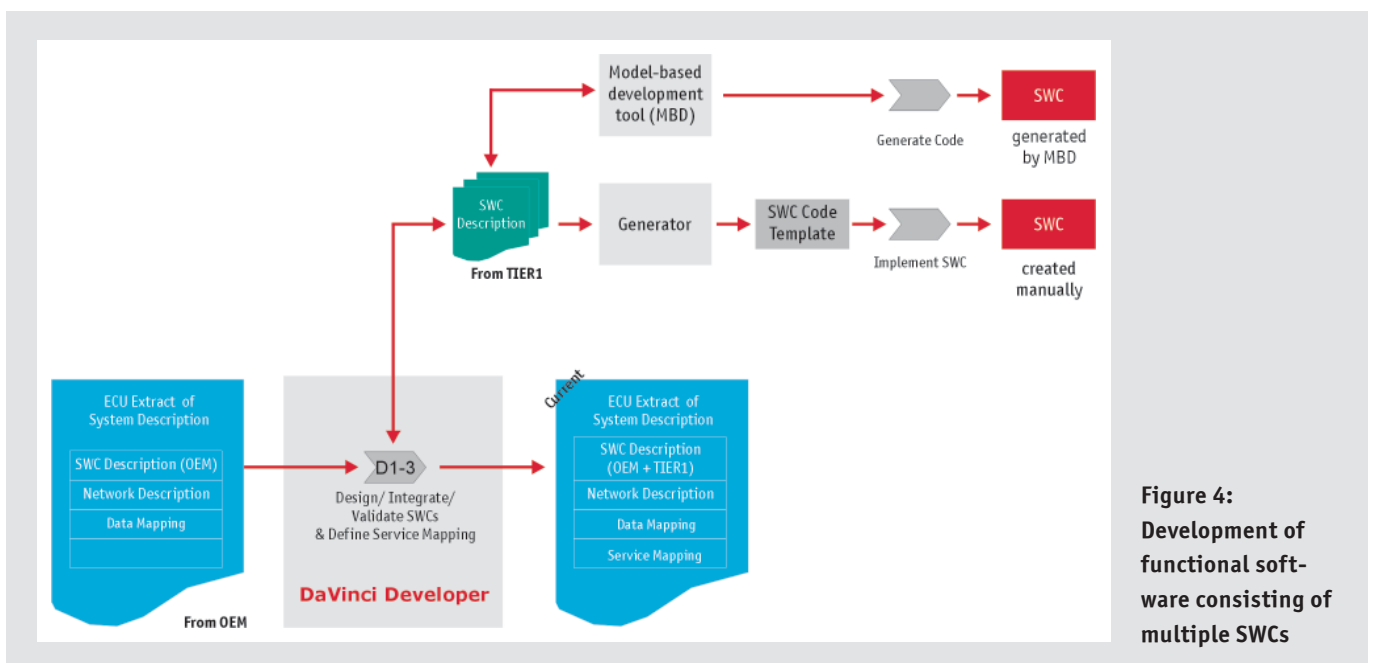


Figure 4: Development of functional software consisting of multiple SWCs

In a separate integration step, data elements of the SWC interfaces are assigned to bus signals (data mapping) – provided that the OEM has not already defined such a mapping in the “ECU Extract of System Description”.

Typically, the “ECU Extract of System Description” will change a number of times over the course of a project. When the Tier1 receives a new version, the SWCs it contains might also be modified. A special Update function makes it possible to accept changes in DaVinci Developer without losing any extensions previously made by the Tier1, such as implementation descriptions or interfaces to standard services.

Configuration of standard AUTOSAR services

One challenge arising in the configuration of AUTOSAR ECUs is how to configure the standard services of the BSW modules. Within the “ECU Extract of System Configuration”, the services are represented by special SWCs, the so-called service components. These service components are created and integrated with the SWCs by what is referred to as Service Mapping, which can be performed automatically with tool support (Step D3).

In the “top-down” approach, the need for services is determined from the entire functional software, and the services of the BSW modules are configured accordingly. This approach makes sense for those services that are not typically specified by the OEM in detail. Examples are services of the NVM (Non-Volatile Memory Manager) or the ECUM (ECU Manager).

In the “bottom-up” method, the service is first configured in the BSW module, e.g. based on OEM requirements. The SWCs are then extended to match the service configuration. An example of this is the DCM (Diagnostic Communication Manager).

Just like the services, the ECU’s I/O interfaces are also represented by a special SWC – the I/O Hardware Abstraction

Component. This SWC can also be created by a “top-down” or “bottom-up” approach.

At the end of the integration process, the Tier1 gets an updated “ECU Extract of System Description”. Besides the OEM’s original version, the file now also contains content defined by the Tier1 and is fully validated.

Modification of the RTE after changing SWCs

If only the implementation of a SWC changes without affecting the SWC’s interfaces or structure, neither the RTE nor the basic software would need to be reconfigured. However, if the structure of a SWC changes, e.g. due to the addition of a new runnable entity, the RTE configuration must be modified. This involves assigning the newly added runnable entity to a task. After this modification has been made (Step D4 in **Figure 5**), the RTE configuration is validated (Step D5).

The DaVinci Developer supports this process with detailed error messages and recommended corrections. For example, inconsistencies are reported so that the Tier1 can correct the “SWC Description” or RTE configuration.

Incorporating changed communication data from the OEM

A typical change scenario is updating the communication data of an ECU, e.g. because the distribution of signals to messages has changed. Such a change is only relevant to those BSW modules related to communication and does not require any changes to the RTE or SWCs.

Figure 6 shows the work steps that are performed to accept the changed communication data. The BSW modules can be adapted automatically: First, a new “ECU Configuration Description” is generated, and the contents of the old “ECU Configuration Description”

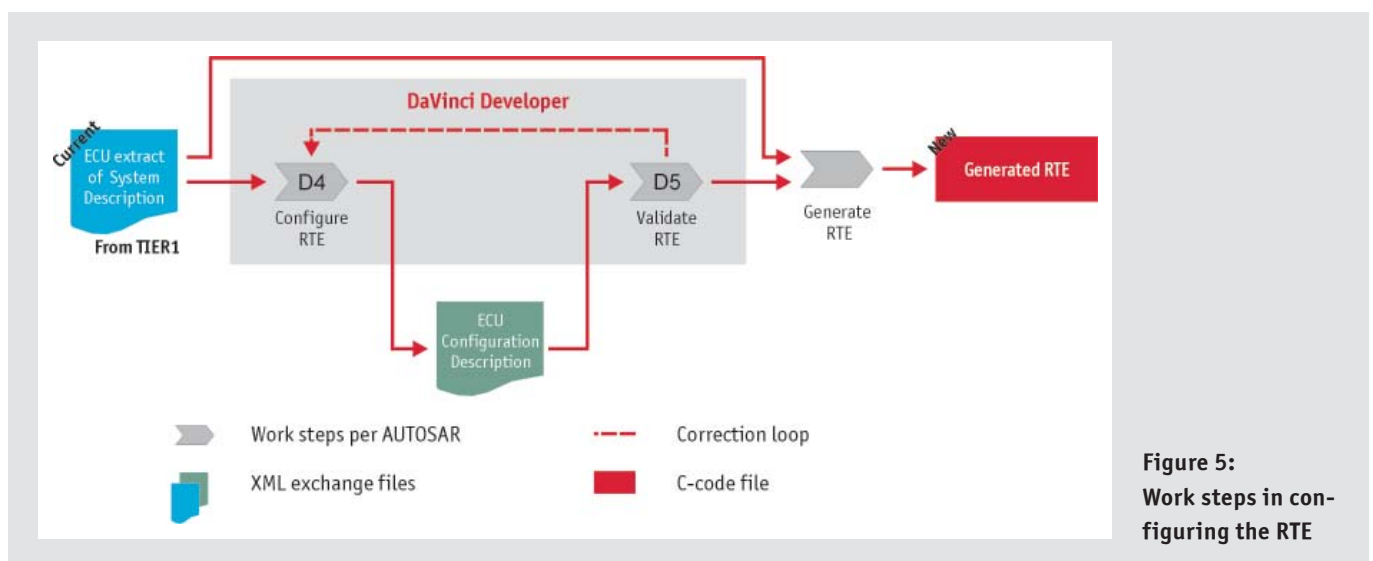


Figure 5:
Work steps in configuring the RTE

are converted to the new “ECU Configuration Description” (Step C1-2). All parameter values unaffected by the change are automatically accepted. Only the parameters of the new or modified signals or messages are then configured in Step C3. Finally, to ensure that all parameter values are consistent, the new “ECU Configuration Description” is validated in Step C4.

As a supplement to the AUTOSAR standard, Vector has implemented an “Update” function and validation in DaVinci Configurator Pro. If the OEM does not provide an AUTOSAR-conformant ECU Extract of the System Description, the Tier1 can instead use the familiar network description formats (DBC, LDF or FIBEX) as inputs to the DaVinci tools.

Switching over to a different processor or processor type

Thanks to the systematic hardware abstraction offered by AUTOSAR the TIER1 only needs to replace the affected MCAL modules when switching over to a different processor device or processor type. This transition is performed with tool support: The old MCAL modules are removed and the new MCAL modules are added to the project by selecting the new BSWMD files. The new platform-dependent parameter values that were added are checked in DaVinci Configurator Pro and reconfigured by the Tier1 as necessary (Step C3 for each changed MCAL module). Consistency of the “ECU Configuration Description” is restored by final validation (Step C4).

Especially useful to the Tier1 is the ability to extend the tool environment, e.g. to support the MCAL modules of future

processors or own BSW modules. This is enabled by the DaVinci Configurator Kit, which is used to create the Tier1 BSWMD files and module interface files. They contain definitions of specific configuration interfaces, validation rules or generators for the BSW modules (Steps C5 and C6).

Replacing OEM-specific diagnostics

If an existing ECU configuration is to be used for a different OEM, the ECU’s diagnostic basic software must be replaced, because it is OEM-specific. This affects the DCM and DEM (Diagnostic Event Manager) BSW modules.

Figure 7 shows how this replacement is made. The Tier1 obtains the new CDD or ODX file from the OEM. These widely used formats contain the diagnostic description data for the ECU. They can be generated with tools such as CANdelaStudio from Vector. DaVinci Configurator Pro utilizes information from these files to automatically configure diagnostic BSW modules for the ECU (Step C7). Analogous to the modified diagnostic BSW modules, the DCM and DEM service components must also be modified and made known to the application SWCs in a “bottom-up” process. For this purpose, DaVinci Configurator Pro takes the “ECU Configuration Description” and generates the “SWC description” files which include service components for DCM and DEM (Step C8).

The Tier1 can now use DaVinci Developer to generate the service mapping for all diagnostic services of the new OEM. Validation of the SWCs ensures that no service is forgotten in the change

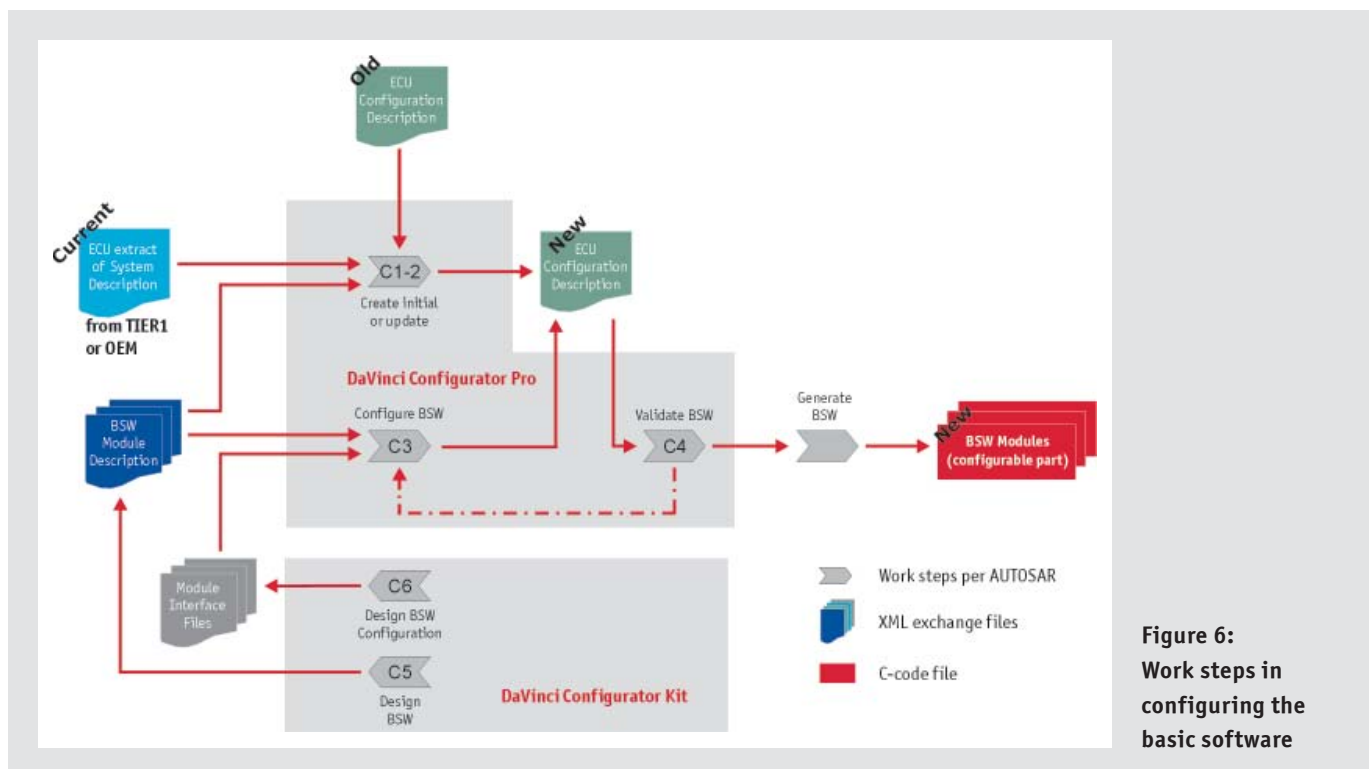


Figure 6: Work steps in configuring the basic software

process. If the application SWCs does not yet offer the diagnostic services required by the OEM, they must be extended (Step D1-3), which in turn requires modification of the RTE (Step D4-5).

Changing I/O signals

In case a new sensor needs to be connected to the ECU the new I/O signal that it uses must be added to the “ECU Configuration Description”. Therefore the Tier1 extends the configuration of the I/O hardware abstraction in DaVinci Configurator Pro by adding the new signal (Step C3 in Figure 6) and modifies the pin mapping in the I/O drivers in the MCAL layer. After successfully validating this configuration change, an updated SWC is generated for representation of the I/O Hardware Abstraction. By importing this SWC into DaVinci Developer, the Tier1 can extend the SWCs of the functional software so that they can access the new sensor.

Advantages of the AUTOSAR configuration process

The AUTOSAR principle “configuring instead of programming” enables early validation of the ECU software architecture. This prevents errors from occurring in a late phase. Nonetheless, the AUTOSAR formats are extraordinarily complex. Good tool support is essential to handle the changes that are typically required over the course of a development project.

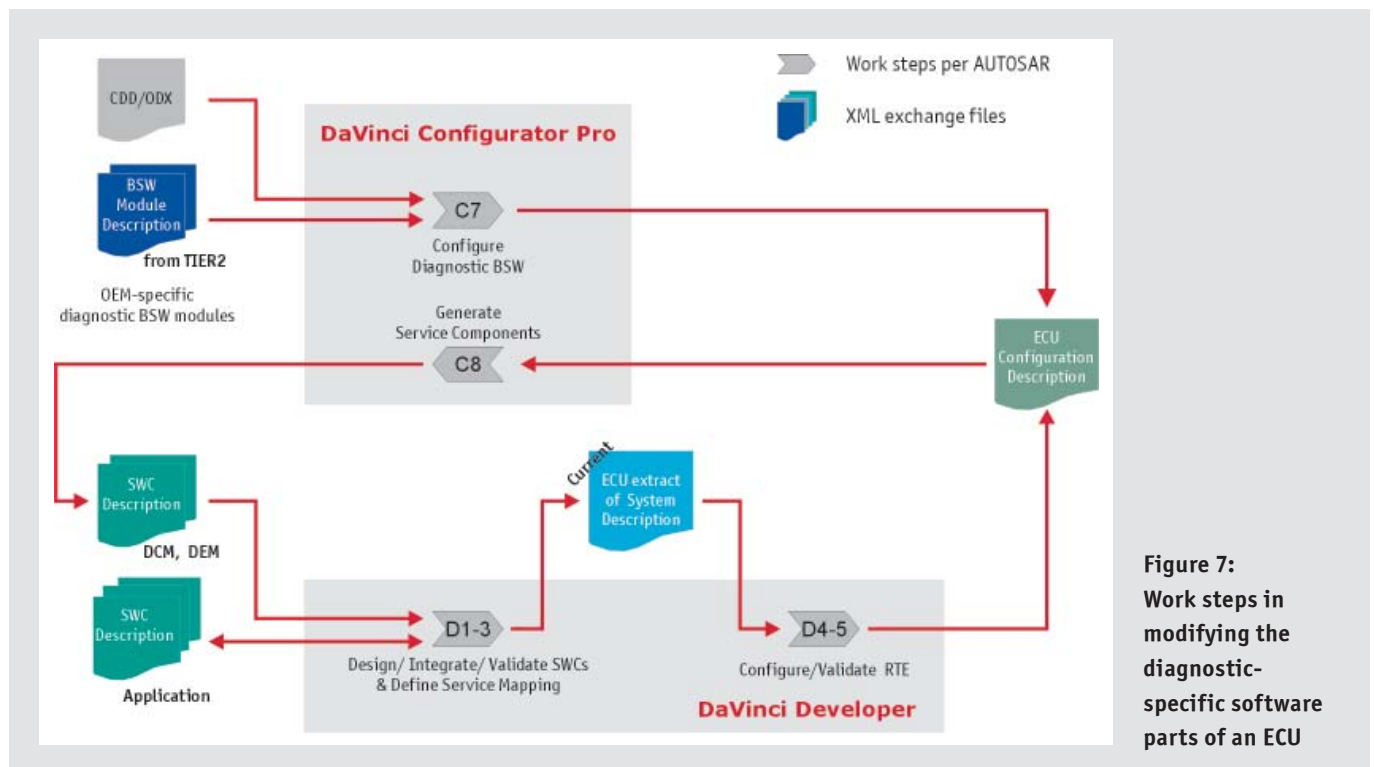


Figure 7: Work steps in modifying the diagnostic-specific software parts of an ECU



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