AUTOSAR describes the electric/electronic architecture of complete vehicles with all configuration details: software, topology, bus communication and ECU-specific parameters. For this purpose, the methodology universally uses process blocks, which are largely organized into activities and work products (Figure 1). By connecting these process blocks, the methodology indicates the way from the creation of the system configuration to the generation of an executable file for a particular ECU. The system-wide view is firmly anchored in the AUTOSAR standard and shapes the methodology throughout.

From Abstraction to Software Component

The methodology starts with the design activity "Develop an Abstract System Description". Here, the functional architecture of a vehicle is described in detail without going into the technical details and the implementation. The result is an abstract system that reflects the formal idea of the overall vehicle system as a network of logical functions and their interfaces and connections. The benefits for further development are obvious: an abstract description makes system dependencies and possible optimizations much easier to analyze. Through the linking of logical functions and all system components, the design always remains transparent. Logic errors or redundancies in functions are detected early and avoided. The creation of the "Abstract System Description", however, is included only as an optional step in the AUTOSAR methodology and only a handful development tools can implement it truly universally up to now. The additional effort for this process step is still worth it because of the enormous efficiency gains. Architects and designers are given an ideal work basis for subsequent distribution and implementation of functions among specific ECUs (Figure 2).

Whether or not the abstract system description comes first, the functional design is implemented in the AUTOSAR software system architecture with the concept of the "Virtual Functional Bus" (VFB). The activity "Develop a VFB System Description" defines the software functions in the vehicle more precisely. The resulting work product is the "Overall VFB System", a system of software components connected by ports that communicate with one another via defined interfaces. It makes no difference whether the information exchange subsequently takes place later via a physical bus or within an ECU – the VFB abstracts this
communication. The individual software components within the VFB system are described as a “Delivered Atomic Software Component” – or frequently also as: Software Component Description (Figure 3).

If the overall system is not continuously kept in mind during this phase, incompatible interfaces are often observed later during integration. Here, for example, multiple interfaces were then accidentally defined for the same software components – with the same logical meaning but with different form and technical details. If these errors are not identified until later in the development process, expensive correction measures are the result.

Modern tools for electric/electronic development, such as PREEvision, provide an effective means of preventing this. Thanks to an integrated connection to the "Abstract System Description" and the implementation of the AUTOSAR-VFB concept in the tool, inconsistencies and redundancies in the design are detected at an early stage and avoided. Ideally, all components, interfaces, and data types are also created only once and kept in a central software library. They are thus available across projects and are reused as required.

**Topology, Design and Standardization of System Communication**

With the "Design System" activity, the software components defined in the VFB system are distributed among a topology of networks and ECUs. System architects normally perform this task. They either take the topology of an existing vehicle and expand it to include new functions as needed, or they create a new topology from the ground up.

In both cases, they define the bus technology (CAN, CAN-FD, LIN, FlexRay or Ethernet) and hardware properties, assign the software components to the ECUs, and analyze the expected communication load within the system. The objective is optimal load distribution on the bus systems and ECUs (Figure 4).

The communication on the individual bus systems is described in the AUTOSAR system design by the activity "Design Communication". A major advantage of the AUTOSAR methodology over conventional formats such as LDF or DBC is that gateways can also be described without difficulty in the communication matrix. Gateways enable transparent transmission of signals and messages from one bus system to another. The route of the signal from the sender to the receiver is always traceable system-wide – independent of the utilized transmission technology (Figure 5).
Transport protocol, network management, and subnets are additional features on the communication layer that are standardized by AUTOSAR. With the transport protocol, large data packets are segmented into smaller data packets and thus transferred over conventional bus systems. After their transfer, the data packets are reassembled by the receiver. This function is supported in AUTOSAR across gateways and is used mainly for diagnostic purposes but is also available for application data.

With "Network Management" and "Partial Networks", system-wide subnets are created that can wake up, keep awake, or shut-down a group of components in order to save energy. The resulting network management configurations can comprise multiple communication clusters. If required, other aspects are also implemented in the AUTOSAR system design:

- If large or very complex data units will be transmitted, a data transformation is defined. Data on the sender side are serialized and then deserialized on the receiver side. It is no longer necessary to map complex data types onto individual signals, so the complexity of the configuration is reduced.
- If multiple ECUs in a distributed system work together in an explicitly defined sequence, a system-wide time synchronization (Global Time Synchronization) is necessary.
- Timing constraints, events, event chains, and information from topology, software distribution, and signal mappings are used to define mandatory time requirements for the system.
- System variants of a vehicle are defined using conditions and binding times on elements of the system description.
- Each system element can be referenced by safety information or provided with an ASIL level.
The result of the system design is the work product “System Description”: it includes a complete vehicle design, which is exchanged between organizations and development tools in XML file format. The AUTOSAR methodology pays attention to different exchange points and recommends the appropriate scope for the respective data delivery. The “Deliverable” generated in this way is then an AUTOSAR XML file (or group of AUTOSAR XML files).

AUTOSAR – Development by the Book?
For those who want to implement the AUTOSAR system view consistently, the question remains: Is a defined development process prescribed in the AUTOSAR standard? The answer is an unequivocal “no”. AUTOSAR merely describes typical blocks from which a development process can be put together. The goal of AUTOSAR is a standardized data exchange between organizations. Only the portion of the system design needed in each case is transferred at the exchange point and then used seamlessly to perform typical vehicle development tasks (Figure 6).

An example of this is the extract of a “Software Component Description” from the “Overall VFB System”: It is used for model-based development and simulation or in code generators. A “System Extract” can, in turn, be created from the “System Description”. It contains a subsystem with one or more ECUs, the description of software components running on the ECU(s), and the communication configuration for the interface to the bus system. The “System Extract” is suitable both for internal use, such as simulations or tests of subsystems as well as for passing on to third parties.

For communication between OEM and supplier, the “ECU System Description” is often also used. It contains the software architecture in its original structure and provides a full description of an ECU with all configuration aspects. The commonly used “ECU Extract”, on the other hand, describes an ECU using atomic software components. The configuration of the ECU is then derived directly from it.
All formats always remain subsets or derivations of the system description. This acts as the central and only database for all AUTOSAR process steps and exhibits so to speak the AUTOSAR system view.

A Standard that Shapes Development Processes

Originated more than 10 years ago, AUTOSAR is today a standard in the automotive industry that is significantly shaping development processes. It never remains static. On the contrary, it is always open to new additions and technical developments. Prominent examples of these include integration of IP technology and Ethernet in AUTOSAR 4.

Still, what will the future bring and is AUTOSAR ready for it? If you talk to manufacturers today, terms like “autonomous driving” and “Car-to-X” are already being used for the next vehicle generation. Vehicles will be controlled with different levels of autonomy. The vehicle will communicate more intensively with its environment for reasons of safety and comfort and will be further networked with services and its environment.

One thing is clear: the complexity of needed electronic systems and the transported data volume will continue to increase rapidly. To control this, it will be increasingly important to keep an eye on the whole system during the development process.

This will be achieved by already consistently implementing the AUTOSAR methodology for current vehicles today. And, in order to technically master the looming digitization of the automotive industry, AUTOSAR is bringing the “Adaptive Platform” into play. The AUTOSAR Consortium has been working since the start of 2015 on this new platform, whose technical objectives include the following points:

- Dynamic adaptation of systems during operation
- Dynamic start of applications and services – even from previously unknown external clients
- Specification of communication paths only during system start or during actual run-time of an application or service
- Safeguarding of communication in the vehicle and with nodes from the environment (cybersecurity)

The “Adaptive Platform” must also ensure by means of methodology that classic ECUs can interact with adaptive ECUs as well as with ECUs that were not developed according to the AUTOSAR standard. For this, they must be represented together in one system.

Figure 5: PREEvision network topology with Ethernet switches in detail.
With these objectives, a paradigm shift in the system concept of AUTOSAR is taking place. To develop service-oriented software architectures in which different nodes subscribe dynamically to services, a different type of system description is needed. The necessary, service-oriented way of thinking in the system design has already been indicated in the last AUTOSAR versions. With the introduction of “Scalable Service-oriented Middleware Over IP” (SOME/IP) in AUTOSAR 4, the cornerstone has been laid for this.

The “Adaptive Platform” will continue this trend and thus strengthen and further establish the systems thinking. It will help developers of vehicle electronic systems even more to keep an eye on the whole system. That has always been a major benefit of development according to the AUTOSAR methodology. And it will remain so in the future.

**Figure 6:** Typical design and development process with blocks of the AUTOSAR methodology.

Translation of a German publication in Elektronik Automotive, issue 10/2016

Image rights: Vector Informatik GmbH