AUTOSAR – Challenges and Achievements 2005

Dr. **Th. Scharnhorst**, Volkswagen AG, Wolfsburg, (AUTOSAR Spokesperson);
Prof. **H. Heinecke**, BMW Group, München;
Dr. **K.-P. Schnelle**, Bosch, Heilbronn;
H. Fennel, Continental, Frankfurt,;
Dr. **J. Bortolazzi**, DaimlerChrysler, Sindelfingen;
L. Lundh, Ford Motor Company, Göteborg/S;
Dr. **P. Heitkämper**, General Motors, Rüsselsheim;
J. Leflour, PSA Peugeot Citroën, Vélizy-Villacoublay/F;
J.-L. Maté, Siemens VDO, Toulouse/F;
K. Nishikawa, Toyota Motor Company, Zaventem/B

1. Kurzfassung

Wesentlicher Treiber der Automobilelektronik ist die Entwicklung neuer, innovativer Funktionen. Zunehmend müssen allerdings Ressourcen dafür verwendet werden, existierende Lösungen an neue Hard- und Software anzupassen. Zudem nehmen Anzahl, Funktionsumfang und Vernetzungsgrad der Steuergeräte rapide zu. Die daraus resultierende Komplexität der Elektrik-/Elektronik-Systeme lässt sich auf herkömmlichem Wege nur noch mit großem Aufwand beherrschen.

Die Automobilindustrie begegnet dieser Herausforderung mit einem Paradigmenwechsel von einem komponenten- hin zu einem funktionsorientierten Entwicklungsprozess sowie mit der Standardisierung wesentlicher Elemente der Elektrik-/Elektronik-Infrastruktur. Vorangetrieben werden die Standardisierungsbestrebungen weltweit von der herstellerübergreifenden Initiative AUTOSAR (AUTomotive Open System ARchitecture, siehe [1], [2]). AUTOSAR wurde 2003 von OEMs und Tier 1 gegründet und umfasst heute darüber hinaus zahlreiche Tool- und Softwarehersteller sowie Halbleiterlieferanten. Ziel von AUTOSAR ist es, die Wiederverwendung von Software herstellerübergreifend zu ermöglichen. Um dies zu erreichen, spezifiziert AUTOSAR eine Methode zur Integration funktionsorientierter Softwaremodule als auch die gesamte Basissoftware und eine Abstraktionsschicht mit standardisierten Schnittstellen als Kommunikations-Layer zwischen Funktionsmodulen untereinander und mit der Basissoftware. Mit Hilfe dieser Abstraktion wird die Verwendung von „Commercial off the shelf“ Hardware wirkungsvoll unterstützt.

Aufbauend auf [2], [3] gibt dieser Artikel einen Überblick über die AUTOSAR Initiative, deren Ziele und Zeitpan. Inhaltliche Schwerpunkte bilden die Darstellung des technischen Konzepts und der Projektstatus’ im Juli 2005.

2. Abstract

Implementations of new innovative functions are the main drivers of today's automotive electronics. Indeed more and more resources are spent on adapting existing solutions to different environments. At the same time, due to the increasing number of networked components, a level of complexity has been reached which cannot be managed using traditional development processes.

The automotive industry addresses this problem through a paradigm shift from a hardware and component-driven to a requirement and function-driven development process, and a stringent standardization of infrastructure elements. One central standardization initiative is the AUTomotive Open System ARchitecture (AUTOSAR). AUTOSAR was founded in 2003 by major OEMs and Tier 1 suppliers and now includes a large number of automotive, electronics, semiconductor and software companies. AUTOSAR aims at facilitating the re-use of software components between different vehicle platforms, OEMs and suppliers. To achieve this, AUTOSAR defines a methodology that supports a distributed, function-driven development process and standardizes the software architecture for each ECU in such a system. AUTOSAR also specifies compatible software interfaces at application level. The abstraction encourages effectively the usage of "commercial off the shelf" hardware.

This article gives an overview of the AUTOSAR initiative, its goals, partners, and roadmap. It describes the AUTOSAR concepts, highlights the achievements to date and the challenges ahead.

3. Electronics and Software in Automotive

Electronics and software represent an ever increasing share in the added value in the automotive industry. Up to 90% of all automotive innovations are attributed to E/E. This trend is driven by safety, assistance, comfort, and legal or environmental requirements. All vehicle domains are affected.

As a consequence, the complexity of automotive E/E architectures is growing exponentially. Semiconductor and computer industries continue to advance at high speed, improving performance and cutting the costs of their components in short time intervals. Since automotive software is embedded in its hardware, the automotive industry has only partly been able to benefit from this process. SW functionality is frozen for a lifetime far beyond the manufacturing time of semiconductors originally decided.

Solution: The automotive industry has recognized that a technological breakthrough is required to address these challenges. Separation ("abstraction") is needed between applications (often implemented in software) and a standardized infrastructure (consisting of networked hardware and basic software). This approach will allow more decoupling between the development of applications and the underlying infrastructure and between the life-cycles of hardware and software. Furthermore it will allow an improved management of the system complexity. Increasing re-use of software will result in higher quality.

Since these issues can hardly be handled by individual companies and constitute an industry-wide challenge, leading OEMs and Tier 1 suppliers have jointly decided to establish an open standard for automotive E/E architecture, leading to the AUTOSAR partnership, which was formally launched in June 2003.

4. The AUTOSAR Partnership

AUTOSAR (AUTomotive Open System ARchitecture) is a partnership of automotive manufacturers and suppliers working together to develop and establish a de-facto open industry standard for automotive E/E architectures. The partnership has adopted a three-tier membership structure which currently consists of 10 Core Partners, 45 Premium Members and 15 Associate Members (Figure 1).

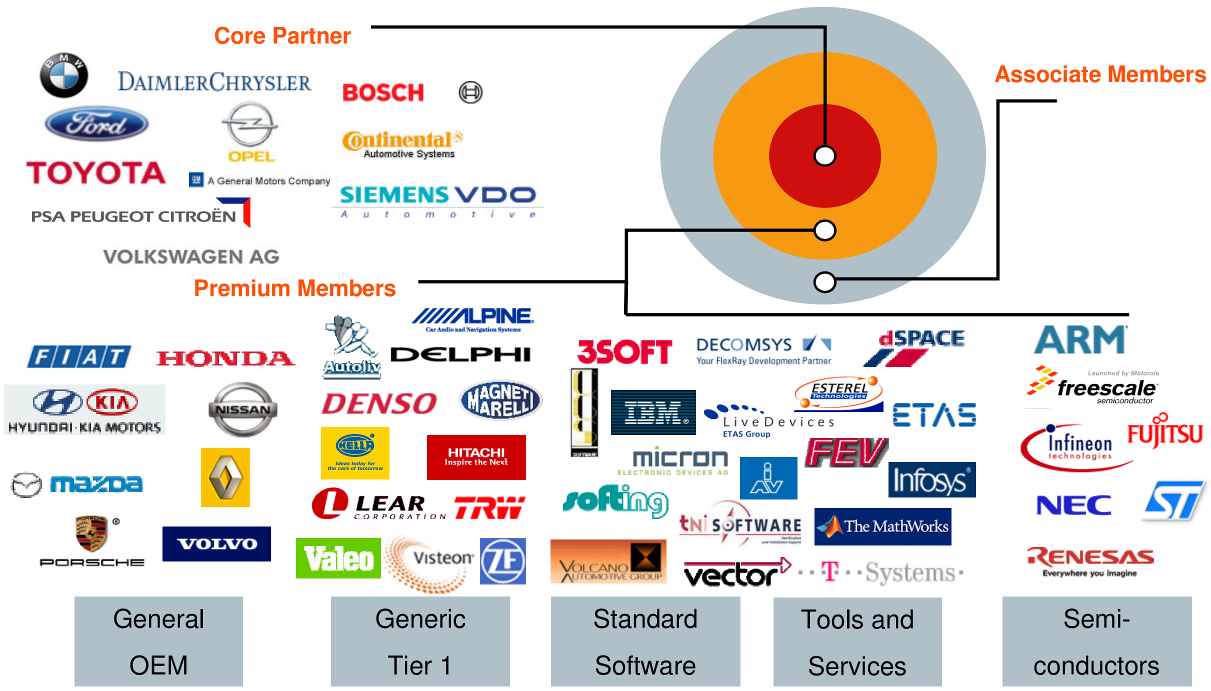

Fig. 1: Worldwide, OEMs and suppliers participate in AUTOSAR (status July 2005)

The common objectives of AUTOSAR are:

- Consideration of availability and safety requirements
- Redundancy activation
- Scalability to different vehicle and platform variants

- Implementation and standardization of basic system functions as an OEM wide “Standard Core“ solution
- Transferability of functions throughout network
- Integration of functional modules from multiple suppliers
- Maintainability throughout the whole “Product Life Cycle”
- Increased use of “Commercial off the shelf hardware”
- Software updates and upgrades over vehicle lifetime

The top priority of the AUTOSAR project is to ensure that proven AUTOSAR Architecture and high quality specifications are available by the end of 2006, see Figure 2. At this time the industrial maturity will be reached to support OEM product development. In order to control complexity and schedule release management has been introduced. Basic software specifications were split into 2 releases. Release 1.0 has been completed within the project and prototype implementations have been started to validate Release 1.0. Release 2.0 is planned to be completed at the end of 2005 based on the stable architecture of Release 1.0. Through feedback from implementation and integration later on the quality and maturity of the specifications will be verified and stabilized.

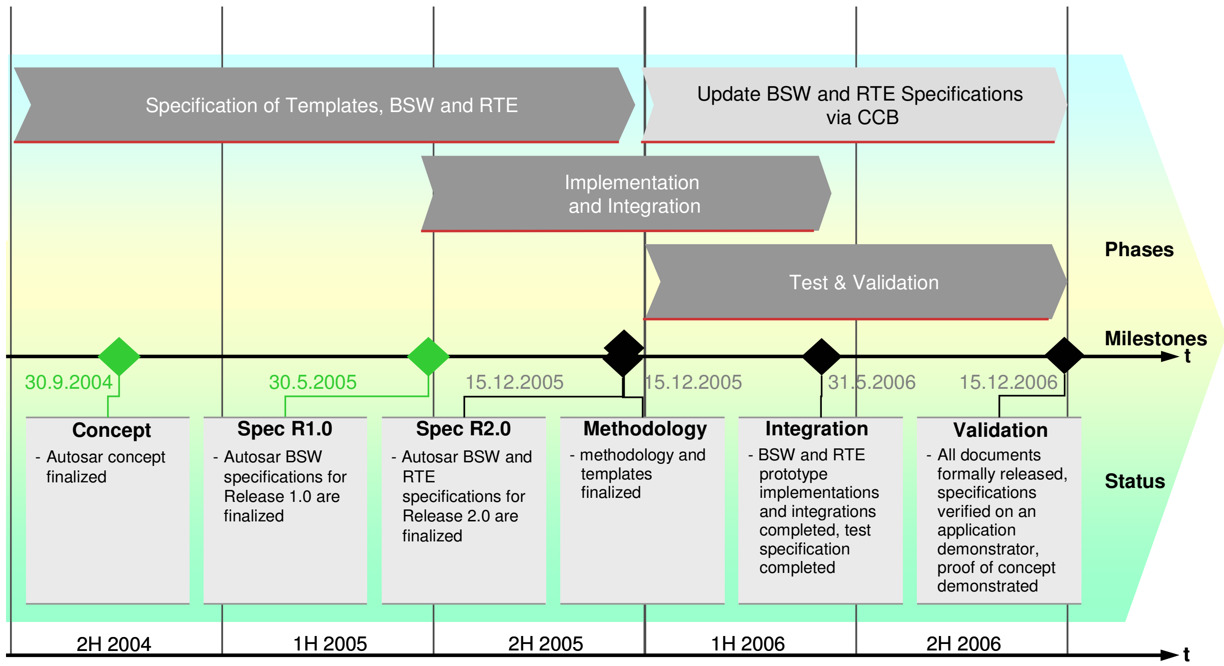

Fig. 2: The AUTOSAR standard will be completed and available to OEM product development in 2006.

5. The AUTOSAR concepts

This chapter describes the basic concepts of AUTOSAR: the “virtual functional bus” (VFB), the AUTOSAR Methodology and the AUTOSAR ECU Architecture.

5.1 The “virtual functional bus” (VFB)

The concept of a “virtual functional bus” was introduced to decouple applications from infrastructure. An application consists of interconnected “AUTOSAR Software Components” (SW-Cs). The “virtual functional bus” (shown in the top part of Figure 3) provides standardized communication mechanisms and services for these components. The VFB acts independently of the chosen mapping of these components on the infrastructure of interconnected ECUs (shown in the bottom part of Figure 3).

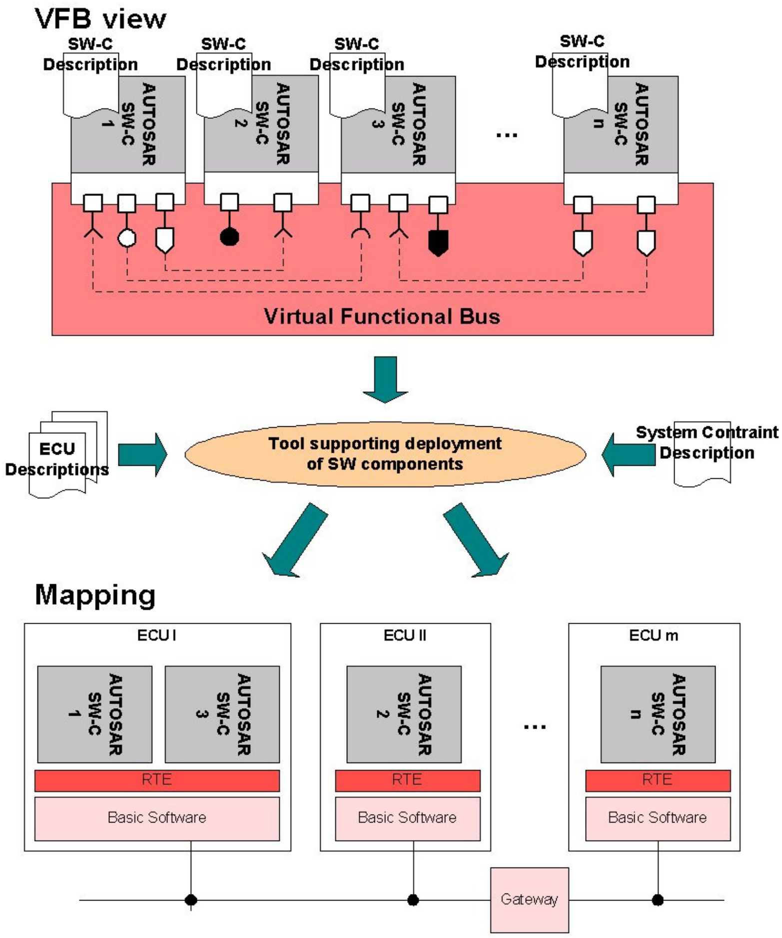

Fig. 3: The “virtual functional bus” (VFB) decouples application and infrastructure

5.2 The AUTOSAR Methodology

This section briefly describes the design method of AUTOSAR-conform systems (Figure 4). To specify the input parameters, engineering work is necessary: a first input is the “Software

Component Description", which defines all AUTOSAR SW-Cs. This includes detailed definitions of the ports, interfaces and connections between the SW-Cs. A second input is the "ECU Resource Description" which describes the available hardware resources (such as peripheral devices, memory and processing units). The third input to the "System Configuration Generator" is the "System Constraint Description". This includes the constraints given by an already existing network architecture.

The "System Configuration Generator" supports engineering decisions at system level. One of the most important decisions taken when configuring the system is the "mapping" of the components on ECUs in the topology. These mapping decisions should take the needs of the components and their communication links, the resources available and the constraints into account. These mapping decisions might also be influenced by other considerations, such as cost or packaging, which are some of the very important product line specific parameters.

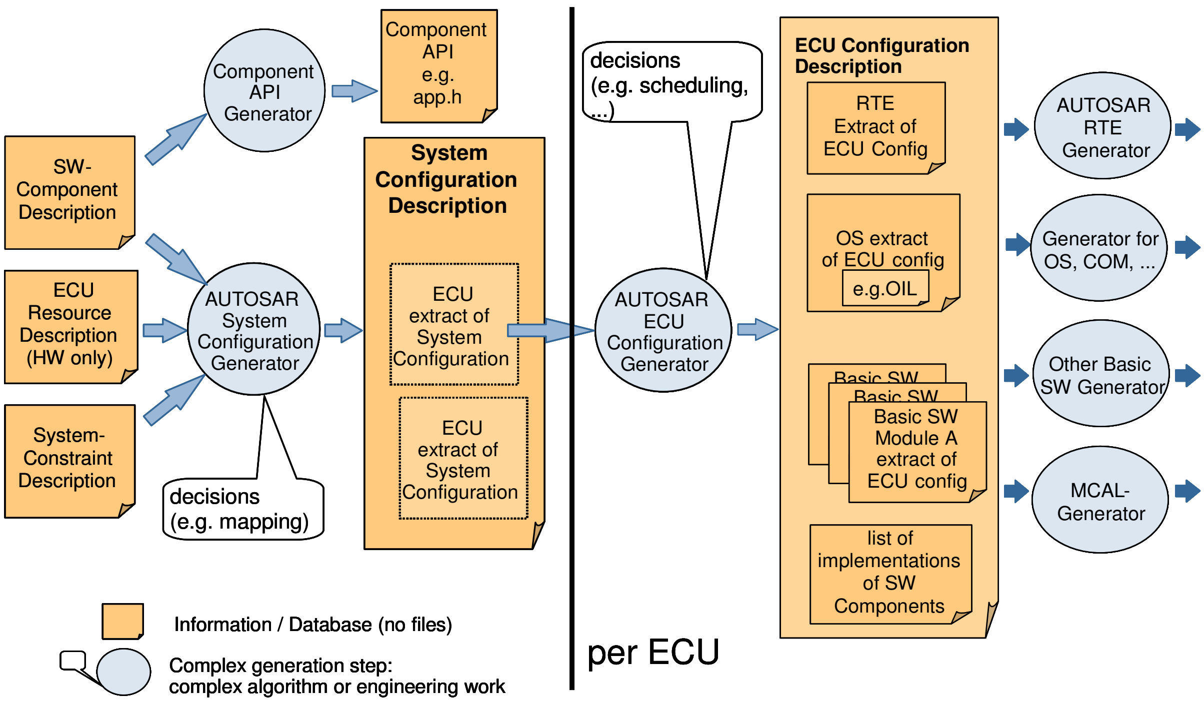

Fig. 4: The AUTOSAR Methodology

The resulting "System Configuration Description" contains the resulting "mapping", as well as a complete communication matrix, describing the timing and contents of all network frames.

From this system-level configuration, an extract can be generated for each ECU. The "ECU extract of the System Configuration" contains parts of the "System Configuration Description" which are relevant to configure and generate that ECU. The next engineering step uses that information to take important decisions on the local design of the ECU (e.g. what OS scheduling policies are used) and configures the entire local basic software (BSW). This step is supported by the “ECU Configuration Generator”. Based on a complete “ECU Configuration Description”, the BSW of the ECU can be configured or generated and integrated with the application, which is provided in the form of implemented AUTOSAR SW-Cs. This results in an executable image for the ECU.

5.3. The AUTOSAR ECU Architecture

The realization of the VFB concept of chapter 5.1 and the AUTOSAR Methodology of chapter 5.2 are only possible when each AUTOSAR ECU has the same BSW functionalities and interfaces. Figure 5 shows the layered architecture of an AUTOSAR ECU, which is principally applicable to nearly all vehicle domains.

The AUTOSAR SW-Cs that are mapped to a specific ECU are located in the ECUs “Application Layer”. The implementation of such an AUTOSAR SW-C is independent from the microcontroller and ECU to which the AUTOSAR SW-C is mapped and from the physical location of other components in the system. An AUTOSAR SW-C interacts with other SW-Cs (on the same or different ECUs) and with the services and resources available on the ECU via the AUTOSAR Runtime Environment (RTE). The RTE decouples the application SW-Cs from the infrastructure software and thus from the hardware.

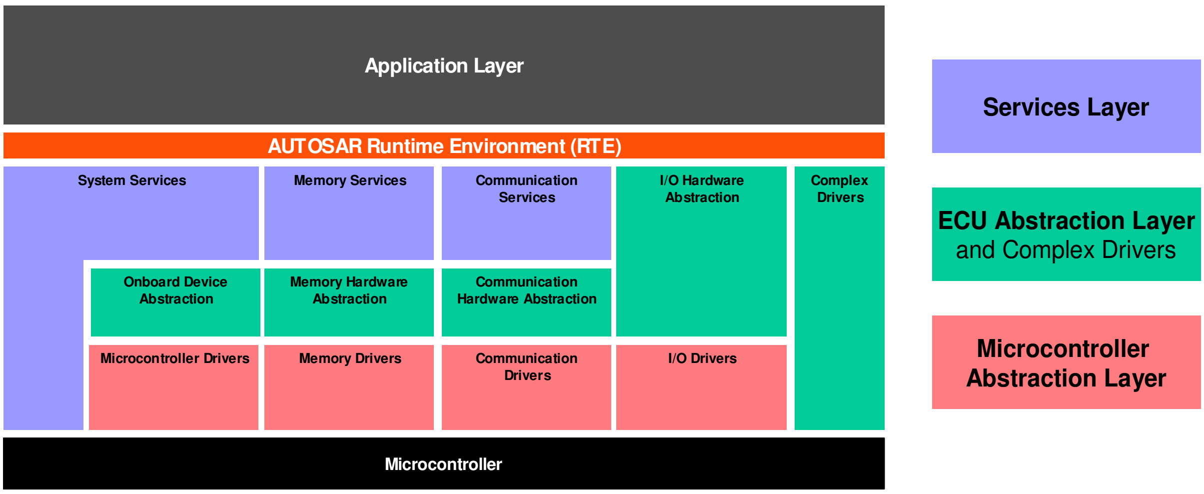

Fig. 5: The AUTOSAR ECU layered architecture

The additional non-functional services are provided by the BSW below the RTE. To distinguish between HW-dependent and HW-independent functionalities, the BSW is divided into the following layers:

- Services Layer
- ECU Abstraction Layer

- Microcontroller Abstraction Layer
- Complex Drivers

The Services Layer provides system services such as diagnostic protocols, non-volatile memory management, ECU mode management and operating system. Except for the operating system, the modules belonging to the Services Layer are HW-independent. The ECU Abstraction Layer abstracts the ECU layout (i.e. how peripherals are connected to the CPU) from the layers above. Although this layer is ECU specific, it is independent from the microcontroller. This independence is achieved by the Microcontroller Abstraction Layer, which contains microcontroller-specific drivers such as I/O drivers to control digital inputs or outputs and ADC-drivers to convert analogue signals to digital values. Modules that handle complex sensors and actuators with strong timing requirements are not standardized within AUTOSAR. These particular modules can be located in the Complex Drivers Layer, however with the strong boundary conditions of performing AUTOSAR interfaces.

6. Challenges and Achievements

6.1 AUTOSAR Methodology and Descriptions

Challenges

The AUTOSAR Methodology, presented briefly in chapter 5, had to be refined:

- A precise definition is required of the information that is provided or required by the activities in the AUTOSAR Methodology. For example: what information does an integrator (e.g. ECU-supplier) need from the system designer (e.g. OEM)? What information does an ECU-integrator need to be able to integrate several SW-Cs from several suppliers on a single ECU?
- In addition, AUTOSAR needs to define a formal exchange format, which allows different partners in the development process (e.g. OEMs defining the system and suppliers providing an ECU) and different tools (such as editors, configurators, generators or analysis tools) to exchange information.

This is not a trivial task due to the size and complexity of the required information structures. In addition, a good balance must be found between defining a precise and rigid methodology and leaving flexibility to tailor the AUTOSAR Methodology to OEM or supplier-specific processes. Finally, the allowed behaviour of tools must be specified precisely enough so that AUTOSAR-compatible tools from different vendors can work together.

Achievements

AUTOSAR has designed a methodology to address this problem and has successfully applied that method in defining major parts of the descriptions.

As shown in Figure 6, AUTOSAR uses a so-called “meta model” to define the structure of all the information that is produced and consumed when developing with AUTOSAR. This meta model is defined using UML and OCL, according to both a specific UML profile and a set of modelling patterns. This meta model approach and the associated tool support for specifying the AUTOSAR information model allow working at the right level of abstraction: an efficient tool is available to generate a precise and formal description of the work products in the AUTOSAR Methodology without being confronted with the issues of data exchange between tools.

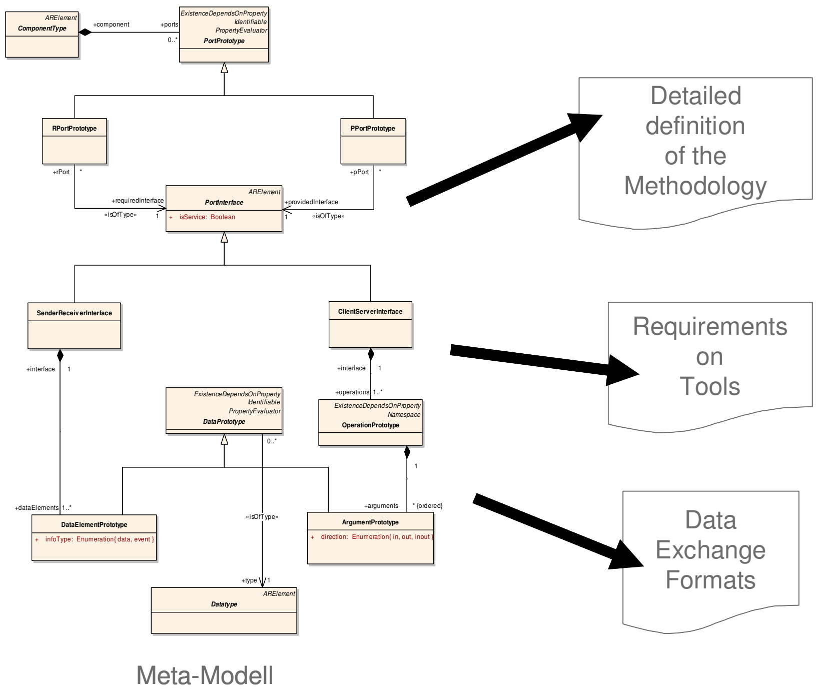

Fig. 6: The meta model defines the structure of the information

AUTOSAR has also specified a data exchange format (using XML) which is automatically generated right out of the meta model. This technology ensures that AUTOSAR conforming tools can easily export and import data.

The meta model currently developed by AUTOSAR covers a large part of the AUTOSAR Methodology, including the description of SW-Cs, their interfaces and resource needs, the description of HW resources, network topologies and communication matrices (covering the common automotive network technologies CAN, LIN and FlexRay). The meta model contains many hundred classes and attributes. Figure 6 illustrates the structured approach.

Ongoing and future work

The meta model is being extended to cover additional aspects such as ECU configuration, error handling and timing. Special care must be taken to ensure long life-cycle and stability.

6.2 Basic Software

Challenges

During the refinement of the BSW presented in chapter 5.3, all BSW modules have been identified. The challenge in AUTOSAR is to specify the interfaces, behaviour and configuration of these modules.

It has to be considered that:

- the BSW exists of 62 modules with their interfaces and dependencies. Communication between work packages has to be encouraged and supported accordingly,
- not the highest common multiple but the best solution has to be found and agreed among all partners,
- next to the harmonization of existing solutions several modules need to be designed from scratch (e.g. the “ECU State Manager” and all FlexRay relevant modules),
- different partners in AUTOSAR might have different criteria to evaluate competing solutions (functionality, performance, size), and
- the configuration process and format for the BSW must be specified according to the AUTOSAR Methodology.

Achievements

The first task was to identify and describe the functionality of the BSW and allocate the different modules of the BSW into the layered architecture. Requirements were defined by a BSW architecture team for each module. After this first decomposition, 9 parallel work packages with experts on the various modules were started. A BSW architecture team ensured the technical coordination and integrity of these work packages and took care of the APIs between the modules defined by different work packages.

The following example shows the data flow through different layers when storing data into an internal or external EEPROM (see Figure 7, compare with Figure 5). The key module taking care of non-volatile data is the “NVRAM Manager”. This module ensures the storage and maintenance of non-volatile data and covers typical automotive requirements. The services of the NVRAM Manager are used to store error information including OBD freeze frames, for example.

The NVRAM Manager located in the Services Layer manages the storage of data into the non-volatile memory. In this case the NVRAM Manager calls an API to access the EEPROM Interface module. A static configuration of the BSW modules, done in the ‘ECU Configuration Generator’ stage (see Figure 4) of the development according to the described AUTOSAR Methodology, is necessary to configure the ECU. In this stage the user knows if an external or internal EEPROM is available. In case of an external EEPROM that is linked to the microcontroller via SPI (Serial peripheral interface), the EEPROM Interface calls the SPI Handler/Driver with several APIs. In case of an internal EEPROM the EEPROM Interface calls the API of the EEPROM Driver. Finally, the EEPROM Driver stores the data into the memory.

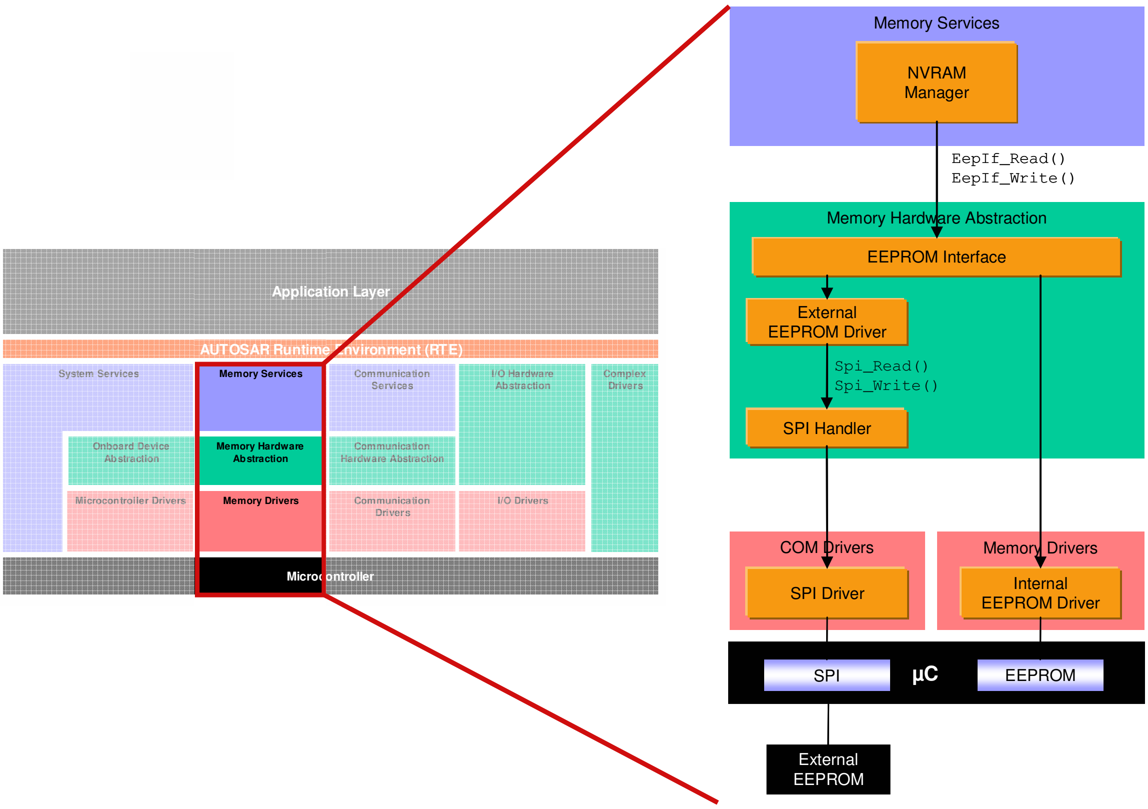

Fig. 7: Data flow through the layers, EEPROM handling, exemplary, extract from the layered architecture

This example shows how the abstraction in the ECU is achieved. The lower interface of the NVRAM Manager is independent of the design of the ECU. This enables the NVRAM Manager to be re-used for different ECUs.

For most features and modules of the BSW, competing solutions were already known and available. In this case, AUTOSAR has managed to harmonize these solutions into a de-facto world-wide standard. Furthermore, AUTOSAR has also specified completely new BSW modules, such as some of the modules inside the FlexRay communication stack, the PDU Router, which decouples the service layer from the different communication technologies, and the ECU State Manager, which allows a standardized start up and shut down of the ECU. Finally, AUTOSAR is completing the specification of the RTE as part of Release 2.0.

It has also been ensured that the overall AUTOSAR concept fits to the BSW. The system engineering approach based on the communication mechanisms used by the VFB therefore has to be implemented in the BSW and in the RTE. The BSW modules linked to the RTE have to provide AUTOSAR interfaces according to the VFB (see Figure 5).

Ongoing and future work

All BSW features were split into two releases. To check the concept of Release 1.0, fourteen AUTOSAR Partners/Members are implementing and integrating 34 software modules in validator 1. To achieve a high quality of the specifications all involved AUTOSAR work packages will incorporate the experiences of the implementation and integration phase into Release 2.0, which will be available during 2005. It is then planned to upgrade the implementations and set up a validator 2 to test the concept of Release 2.0.

In order to ensure and to maintain a stable standard the control of the conformance with respect to the valid set of specifications is mandatory. Conformance tests and processes are therefore being prepared.

The standardization of the configuration process and the XML based configuration format is still challenging. The different stages of the configuration need to be considered in the process as far as the dependencies of the modules are concerned.

6.3 Standardization of application-level interfaces

In addition to the standardization of the infrastructure it is advantageous to develop a classification strategy to deal with competitive functionality at application level. Of course, unanimity between AUTOSAR Partners has to be secured for both, the level of details of the interface description (which has to be standardized) and the level of functional breakdown. At present, AUTOSAR focuses on body, powertrain and chassis domains. A subset of functional interfaces has been standardized to agreed levels in each domain, e.g. external light, wiper/washer, driver request and adaptive cruise control.

6.4 Project organisation and project management

AUTOSAR is managed by its Core Partners who built up a distributed, virtual organization with consensus-based decision processes. Some 650 people are currently contributing to the development of the standard, representing a full time equivalent of 175 staff. This places high demands on effective and efficient project and resource management, accurate reporting, quality assurance and transparent communication.

As a result of the tight schedule, lots of AUTOSAR work packages work in parallel on many specifications. This requires intensive communication to assure the technical interaction of the standard as a whole. Alongside the AUTOSAR work packages, teams are arranged (e.g. system team, BSW architecture team) to control, coordinate and review the work done with respect to technical issues.
A formal review process based on a 4-milestones concept is also established for each module in order to guarantee high quality and consistency of all resulting specifications.

7. Conclusion

Having already completed some core elements, AUTOSAR continues to focus on completing a commonly accepted standard for the development of future automotive EE systems on schedule. The AUTOSAR Methodology has been defined and BSW specifications of Release 1.0 have been completed. The prototype implementation of Release 1.0 is in progress resulting in valuable feedback and quality enhancement for the forthcoming finalization of Release 2.0.
AUTOSAR pushes the paradigm shift from an ECU based to a function based attempt in automotive software development and enables the management of the growing E/E complexity with respect to technology and economics.

8. Abbreviations

ADC	analogue digital converter	OCL	object constraint language
API	application programming interface	RTE	runtime environment
BSW	basic software	SPI	serial peripheral interface
ECU	electronic control unit	SW	software
E/E	electric/electronics	SW-C	software component
EEPROM	electrically erasable programmable ROM	UML	unified modelling language
HW	hardware	VFB	virtual functional bus
NVRAM	non volatile RAM	XML	extensible mark up language

9. References

[1] www.autosar.org
Official website of the AUTOSAR partnership

[2] Heinecke, H. et al.: AUTomotive Open System ARchitecture - An industry-wide initiative to manage the complexity of emerging Automotive E/E-Architectures
Convergence 2004, International Congress on Transportation Electronics, Detroit, 2004

[3] Heinecke, H. et al.: AUTOSAR – An industry-wide initiative to manage the complexity of emerging Automotive E/E-Architectures
Electronic Systems for Vehicles 2003, VDI Congress, Baden Baden, 2003

[4] Scharnhorst, Th.: AUTOSAR - ein wichtiger Beitrag für die Automobilarchitekturen der Zukunft
Vision Automobil, Handelsblatt Jahrestagung Automobiltechnologien 2005, Munich, 2005

10. Contact

request@autosar.org