2008-21-0019

AUTOSAR on the Road

Gerulf Kinkelin, Alain Gilberg, Bertrand Delord, PSA Peugeot Citroën
Harald Heinecke, Simon Fürst, BMW Group
Juergen Moessinger, Andreas Lapp, Bosch
Ulrich Virnich, Stefan Bunzel, Continental
Thomas Weber, Noë Spinner, Daimler
Lennart Lundh, Daniel Svensson, Ford Motor Company
Peter Heitkämper, Fredrik Mattsson, General Motors
Kenji Nishikawa, Hiroyuki Hirano, Toyota Motor Corporation
Klaus Lange, Bernd Kunkel, Volkswagen

Copyright © 2008 CTEA

ABSTRACT

The AUTomotive Open System ARchitecture (AUTOSAR) Development Partnership has published early 2008 the specifications Release 3.0 [1], with a prime focus on the overall architecture, basic software, run time environment, communication stacks and methodology.

Heavy developments have taken place in the OEM and supplier community to deliver AUTOSAR loaded cars on the streets starting 2008 [2].

The 2008 achievements have been:

- Improving the specifications in order to secure the exploitation for body, chassis and powertrain applications
- Adding major features: safety related functionalities, OBD II and Telematics application interfaces.

INTRODUCTION

The objective of the AUTomotive Open System ARchitecture (AUTOSAR) Development Partnership is to establish an open industry standard for the automotive software architecture between suppliers and manufacturers [2].

The standard comprises a set of specifications describing software architecture components and defining their interfaces [3].

The principal aim of the standard is to master the growing complexity of automotive electronic architectures: with up to 70 ECUs in a car working together and controlling major functions of the vehicle. The need to build a common architecture became stringent for a variety of reasons, among which:

- Defining a common language between the ECUs cooperating on the same functions
- Splitting the hardware and the software in order to allow software reuse or smooth evolutions limiting redevelopments and validations. Finally AUTOSAR is enabling several different functions or software modules to be hosted on the same ECU, independently from the supplier of either part.

Since SAE Convergence 2006 [3], the AUTOSAR Development Partnership has been:

- Improving the existing specifications in order to enable and to safeguard the exploitation with respect to body, chassis, and power train applications by the members and the partners of AUTOSAR
- Adding major features, ranging from OBD II, Human Machine Interfaces to application interfaces.

The ongoing developments of AUTOSAR products by the member companies provide a unique feedback loop into the development of the standard itself, which allows fast and pragmatic improvements. The reusability of software has already been experienced in major developments and it has resulted in substantial savings in the overall development costs.

This paper will guide the reader through the main focus areas for the Phase II of the AUTOSAR Development Partnership and provide additional insight to the work planned for this phase. The last two parts of this paper are dedicated to a more detailed view on the standardization of application interfaces and conformance testing.

AUTOSAR BASICS

AUTOSAR as a standard is providing specifications on 3 main areas, software architecture, application interfaces and methodology. As a development project, AUTOSAR is following a stepwise approach in order to meet both quality and time schedule requirements.

TECHNICAL CONCEPTS [6], [8]

AUTOSAR software architecture

The AUTOSAR layered architecture [10] is offering all the mechanisms needed for software and hardware independence. The upper layer is dedicated to the applications; the lower part, the infrastructure, is containing the basic software layer and the Run Time Environment (RTE) [11].

The basic software layer containing the 53 Basic Software Modules [9], is organized in 3 layers providing the different levels of abstraction from the hardware (see Fig.1): the Electronic Control Unit (ECU) and the microcontroller ; the upper layer, hardware independent, is providing services to the applications software via the RTE.

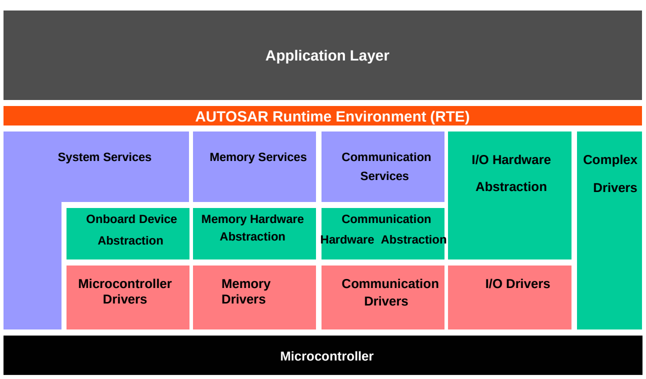

Fig.1: AUTOSAR layered software architecture

The RTE is the only interface to the software applications in order to fulfill this hardware/software independence. As a central concept of AUTOSAR, the RTE is the implementation on a real ECU of the Virtual Functional Bus [6] (VFB) (see Fig.2) where all the vehicles applications are connected regardless of their allocation to different ECUs.

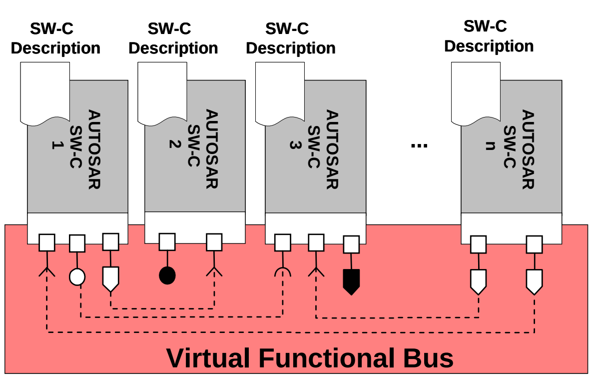

Fig.2: Deployment of application SW-Cs on the VFB

Application interfaces [20], [21], [22]

In order to facilitate the integration of application software components in a system, a clear semantic of the interfaces are being defined in function catalogues. Descriptions of the interfaces that conform to the standardized AUTOSAR format are used as an input to the development process without the need to disclose any internal design, often competition relevant.

AUTOSAR methodology [16]

In addition to defining architecture and interfaces, AUTOSAR includes a model based development methodology containing all the concepts needed for a complete generation of an AUTOSAR system. This methodology is including all templates and XML description formats necessary to describe, configure and generate automatically the infrastructure. The following templates are modeled into the UML AUTOSAR Metamodel:

- The Software Component Template [19] is dedicated to the description of application software components at the VFB level. It contains an architectural view of the software components with its ports, interfaces and data elements and an internal view with runnable entities.
- The System Template [18] describes the network topology (bus systems, connected ECUs and gateways…), the communication for each channel (K-matrix, gateway table), and the allocation of software components to the different ECUs.
- The ECU Resources Description Template [17] describes the general characteristics of each ECU used in this system: name, manufacturer, environment, available resources, available programming capabilities, available basic software modules and signal path from pins to the ECU abstraction.
- The Basic Software Configuration Template includes all the relevant parameters dedicated to the configuration of each Basic Sofware Module.

AUTOSAR DEVELOPMENT APPROACH

The AUTOSAR development partnership has selected a phased strategy in order to manage the challenges ahead along with the tight schedule.

The phase I (2004-2006) of the development has been designed in order to provide a baseline for the industrial deployment on the market. The phase II (2007-2009) is focusing on the introduction of enhanced concepts.

As shown in Fig.3, related to the phase 1 development approach, the specifications out of one development step are validated on harware platforms, providing further improvements for the next steps.

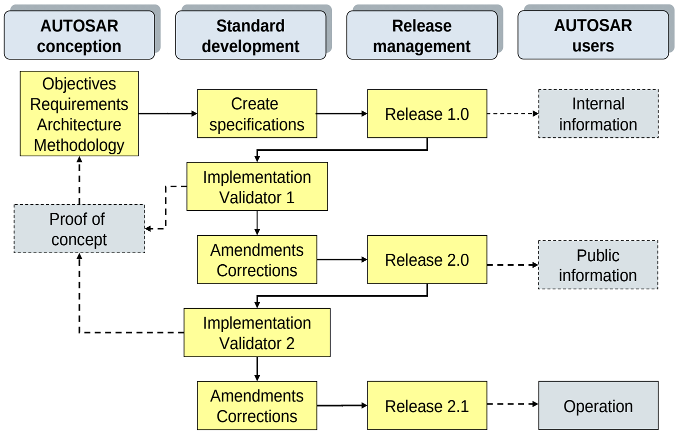

Fig.3: AUTOSAR development and validation approach

ACHIEVEMENTS OF THE AUTOSAR PROJECT

AUTOSAR PROJECT PHASE I OUTCOME [3], [4]

The main purpose of the phase I was to achieve a complete set of specifications of architecture, methodology and templates.

The Release 2.1, end of 2006, is a complete set of specifications including the configuration concept. It is an update of the Release 2.0 with the outcome from implementation and validation of BSW modules on hardware platforms (see Fig.3). The same process of specification-implementation-validation has been performed with the release 1.0, leading to the release 2.0.

Both releases 2.0 and 2.1 are in use by several AUTOSAR members for series productions. The release 2.0 has been published, for information, on the AUTOSAR web site [1].

VALIDATION APPROACH [2]

During the AUTOSAR phase I, the two validations steps have been performed on hardware platforms (16 bits, 32 bits) in order to improve the specifications of both releases 1.0 and 2.0 and to obtain the proof of the concept. On the Release 2.0 validation, the new configuration concept has been implemented with several tools.

The first feed back on specifications was given during the review phase and unit tests by the software houses in charge of the implementation of the BSW modules. Then, along with the integration of the modules, made by a third party company, interfaces, behaviour and configurability have been validated.

A first system test has been performed on a single ECU using a realistic application calculating the vehicle speed. Finally, this application was distributed on 4 ECUs interconnected via CAN and FlexRay.

This validation approach has led to big improvements of the AUTOSAR Basic Software specifications; it is applicable as a generic approach for further integration tasks within an automotive product development.

AUTOSAR PROJECT PHASE II (2007-2009)

As shown in Fig.4, three releases have been planned for the AUTOSAR phase II, providing a continuous improvement of the specifications and introducing new concepts.

The release 3.0, was published early 2008 on the AUTOSAR web site [1]. It has included a large number of improvements and corrections with respect to the previous releases.

The release 3.1 is dedicated to the incorporation of OBD regulations support mechanisms.

At the end of phase II, the Release 4.0 will integrate new features and conformance test specifications. The overall planning of Phase II is shown in Fig.4 with three main sections (Basic Software and RTE, Methodology and Templates, Application Interfaces). The Release 4.0 will be delivered end of 2009 after the validation phase.

-Basic Software and RTE:

On the first section of this picture, the 3 releases of phase II are shown on different lines. The Release 4.0 is including the sequence on concepts preparation.

The work on Conformance Test specification and validation is shown on separate lines.

-Methodology and Templates, Application Interfaces:

They are the two next sections of the picture: the 2 releases 3.0 and 4.0 are shown on the same line.

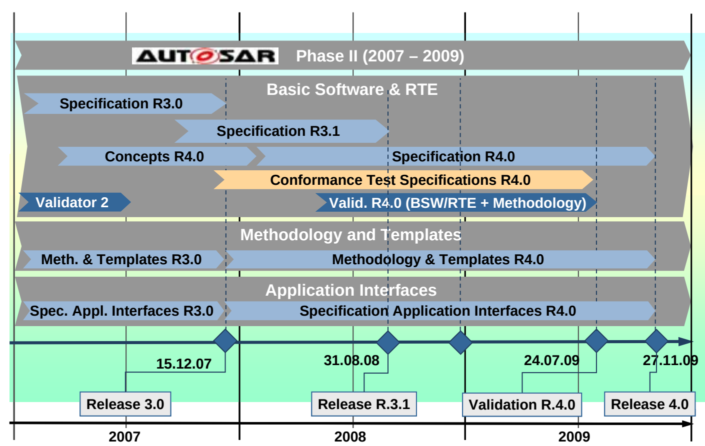

Fig.4: AUTOSAR planning of the phase II

Furthermore, AUTOSAR is defining the follow-up organization (phase III) starting from January 2010, for maintenance and further evolutions of the standard.

ACHIEVEMENTS OF RELEASE 3.0

158 documents are part of the release 3.0, a large part of them being quite stable, 30% with significant improvements, and 10% of those documents were

completely new. In total, more than 500 change requests were processed improving the quality of the standard.

Architecture (BSW and RTE) [10], [11]:

The basic software architecture has reached a high level of stability; commercial implementations of the basic software modules for the release 3.0 are available on the market. Major improvements were made on the wake up and start up of ECUs and networks providing both harmonization of features and reduction of complexity. Beside evolutions of existing modules, three new basic software modules were added to the communication stack, dedicated bus state managers for CAN, LIN and FlexRay (see Fig.5).

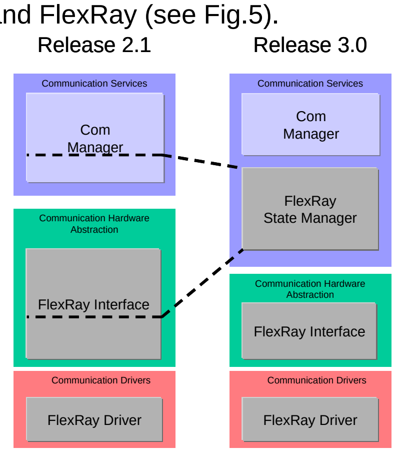

Fig.5: Evolution of the communication stack in Release 3.0

Methodology and Templates [16]:

The improvements made on the templates ensure the consistency of the standard. Interfaces, behaviour and configuration paramaters of the BSW are now included in AUTOSAR models allowing a better control of evolutions and the automatic generation of the relevant specifications chapters as shown in Fig.6.

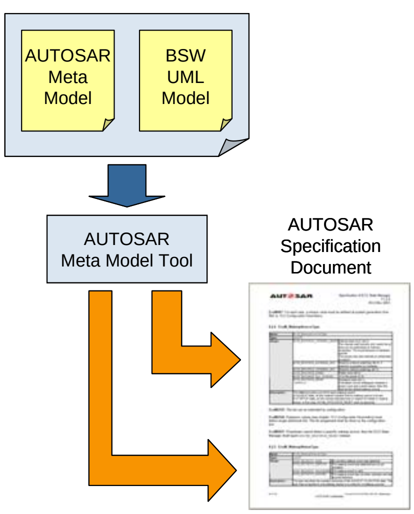

Fig.6: Model based generation on specifications

Furthermore, AUTOSAR has also worked out the harmonization between the ASAM FIBEX standard [23] and the AUTOSAR System Template: both metamodels are now matching up: FIBEX tools describing topologies, networks and communication can be integrated easily into AUTOSAR methodology and tooling.

Application Interfaces:

In order to cover the full vehicle functionalities, AUTOSAR has started, during the phase II, to work on two new car domains Application Interfaces: Telematics/Multimedia/HMI and Passive Safety. Moreover, Powertrain, Chassis, and Car body interfaces have performed their first steps of integration. The Release 3.0 is containing both the explanatory documents of each of the latest domains and the related integration master table available in XML format.

RELEASE 3.1

The release 3.1 is an improvement of the release 3.0, delivered mid 2008 and dedicated to the introduction of the OBD (On Board Diagnostics) features into AUTOSAR BSW modules. All the variants of different OBD regulations are covered (OBDII, EOBD, JOBD…) are covered in this release impacting mostly the three BSW modules of the diagnostic services.

AUTOSAR RELEASE 4.0 PLANNING

The Release 4.0 will introduce new concepts on both architecture and methodology. In addition, a full set of conformance test specifications on basic software specifications will be provided on module level. Technical readiness of the specifications is planned at the end of 2008. These specifications will be validated in order to reach the final approval state end of 2009.

CONCEPTS

The new concepts introduced in AUTOSAR Release 4.0 are adding technical and functional improvements and extensions to the following main areas: functional safety, architecture, communication stack and templates. Some of these concepts are summarized in this chapter.

Functional safety concepts

Functional Safety is one of the main objectives [5] as AUTOSAR is going to support safety related applications and therefore to support the ISO 26262 standard. Exemplarily some of the new safety features are mentioned below:

The memory partitioning concept will provide a fault containment technic to separate software applications from each other. This concept is allowing safety and non safety applications to be implemented on the same ECU.

Defensive behaviour is a solution preventing from data corruption and wrong service calls on microcontrollers having no hardware support for memory partitioning.

Support for multi microcontrollers architectures aims on detection of faults on the core microcontroller by a secondary unit.

Program flow monitoring is controlling the temporal and logical behaviour of applications by checking at specified points of code execution if the timing and logical order of execution requirements are met.

Architectural improvements

The multi-core concept will enable a high integration of independent ECUs and the migration of one cohesive application to a multi-core ECU. One approach will be a single operating system managing all other cores, new services will be added in order to activate tasks or send events across cores, and synchronize or protect shared objects.

Harmonizing and completing the local error handling mechanisms, the error handling concept has been designed for reusing the same strategy within different architectural areas (memory stack, communication stack, etc.). It will enable applications specific decisions allowing for example, specific SW-Cs to be stopped and restarted.

Evolutions of the communication stack

Currently the AUTOSAR LIN stack is supporting LIN 2.0 master functionality. The changes from LIN 2.0 to LIN2.1 do not affect the protocol but the transport layer, diagnostics and API. A specific configuration parameter will be introduced that allows switching on and off LIN2.0 specific functionalities.

The AUTOSAR FlexRay stack, currently relying on the version 2.1 of the protocol, will be updated with the version 3.0 providing new features like the Time Triggered Master. Some Flexray features will be implemented directly on hardware, offering ways for performant implementations.

The current Release 3.0 is restricting signals to 8 bytes. This is only due to the CAN and LIN frames format, the AUTOSAR architecture is able to overcome this restriction, therefore Release 4.0 will implement the support of large data types and dynamic length signals.

Methodology and templates

Improvements of methodology and templates are planned for the release 4.0: for example harmonisation of ECU configuration parameters, enhancements on measurements and calibration, rework of the ECU Resource Template, further alignment with the FIBEX standard.

Moreover, focussing the large variability found in vehicles and the scalability to different vehicle and platforms variants which is one of the main objectives of AUTOSAR, the variant handling concept will give the ability to support variants in different situations:

- Different usage of a given SWC with conditional or optional interfaces,
- Allocation or implementation of this SWC on different platforms or topologies,
- Adaptation of the system description/generation to different communication matrix.

Finally, the methodology and templates will be enriched with the ability to describe timing requirements.

STANDARDIZATION OF APPLICATION INTERFACES

Release 4.0 will contain a large set of application interfaces standardized by the AUTOSAR partnership.

Efforts have been focused on the non innovative applications as a loophole to emphasize software reuse and exchange considered as one of the main requirements. Reuse of applications will avail the use of AUTOSAR Standardized Application Interfaces. For instance some automotive applications are already known all over the world, like the management of doors and speed computation.

The application interfaces table contains a richness of data standardized by experts of all partners. These standardized interfaces allow software designers and implementers to use them in case of expanding or reusing SW-Cs independent of a specific HW/ECU.

AUTOSAR AND VEHICLE FUNCTIONAL DOMAINS

Application interfaces are dwelling across several vehicles functional domains. AUTOSAR has arbitrary chosen to split vehicle domains into five groups: Body, Powertrain, Chassis, Occupant & Pedestrian Safety and HMI, Telematics & Multimedia. AUTOSAR's pundits have had to face a tremendous effort, firstly by defining the level of description intended to be worked and standardized and secondly by ensuring the consistency among all domains. All vehicle functional work groups dedicated to cover definition of standardized application interfaces have a different mind set and habits to address description of application interfaces.

In general an application is referring to the intellectual property and the scope of AUTOSAR is not to standardize the functional internal behavior (e.g. algorithms, optimization) but the content exchanged between applications in order to clarify exchanges between the automotive community from OEM to suppliers such as supplier to supplier and so forth.

FROM VEHICLE FUNCTIONS TO APPLICATIONS AND SOFTWARE COMPOSITIONS

AUTOSAR has specified a layered software architecture also to achieve a clear disjunction between the applications and the rest of the ECU software. An application might be described through the composition of sets of software components, using the AUTOSAR Software Component Template [19]. A software composition could be seen as an elementary piece of software. AUTOSAR supports the engineer in mapping the components to ECUs. The Fig.8 is showing such a composition of Software components with their XML representation.

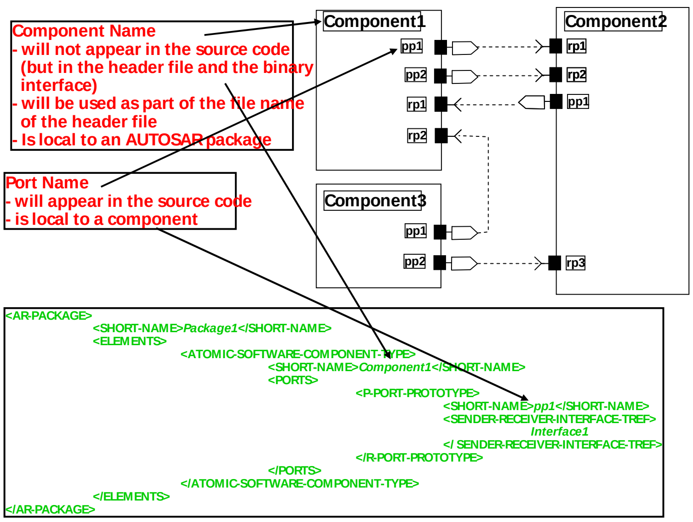

Fig. 7: Component and port descriptions from a flat view to the xml structure outcome

Each domain has its own view of details required to mark off boundaries and inner description of applications. An AUTOSAR internal process to allocate components to domains has been raised in order to identify exchanges between those domains and keep them consistent. AUTOSAR Phase I and II have been used persistently to identify and to consolidate components in order to select application interfaces candidates for the standardization.

The information exchange between components takes place through an interface. Following AUTOSAR Standardized Templates, an interface is depicted as a set of data elements. Indeed an interface might contain one to several data elements. These data elements are based on data types. AUTOSAR offers an abstracted way to define data types. Nevertheless brought back to the application interfaces standardization process these data types have been categorized to allow functional engineers to define their own requirements.

For instance some data types are dealing with continuous values and they are requiring upper, lower values, offset and units. While some others data types are dealing with a list of enumerations. At the end an interface is connected between components through the definition of ports. Standardization of ports is not part of the AUTOSAR standard but it is provided to alleviate the understanding of domain interactions.

AUTOSAR APPLICATION INTERFACES TABLE

Application interfaces are chasing one objective: establish an enhanced data dictionary, comprising not only data signals but also software components, ports and interfaces. This data dictionary will be used by a wide community of automotive engineers.

Application interfaces have been defined above the Virtual Functional Bus concept. It means that all application interfaces could be rearranged in different applications if necessary.

Numerous elements raised by the standardization of application interfaces have been collected within a table and delivered as a XML outcome that conforms to the AUTOSAR Standard R3.x and R4.x. The AUTOSAR application interfaces table contains all software components required to standardize interfaces and elements which belong to it.

Additionally with the application interfaces table the AUTOSAR partnership delivers an Explanation of Application Interfaces. It is a document which contains concepts and information related to the content of the application interfaces (e.g. explanatory information). This document also contents all information that cannot be integrated in the application interfaces.

Centralization and homogenization of results led AUTOSAR to work out a modeling guide including a naming convention.

A data integration process evolves and maintains the content trough several steps in Phase II. Integration Step1 has been used to apply the AUTOSAR Modeling Guide including the naming convention on results within the application interfaces table. This tremendous effort achieved on R3.0 results is reused and evolved by the AUTOSAR functional work groups for the Integration Step2 which will contain a new set of standardized interfaces. Step3 is expected as an optimization step to provide means to support enhancement of the AUTOSAR Modeling Guide, integration of signal qualifier features, integration of the AUTOSAR variant handling concept.

AUTOSAR CONFORMANCE TEST

OBJECTIVES AND SCOPE

AUTOSAR specifications are now in use by many companies to build automotive products and bring them on the market. The use of AUTOSAR trademark implies the conformance to the AUTOSAR specifications which is a basic condition for interoperability, reuse, portability and scalability of those products. They have therefore to demonstrate their conformance to the AUTOSAR standard.

In order to ensure that implementations of AUTOSAR basic software modules and run time environment are behaving according to the dedicated software specifications (SWS), conformance test specifications (CTSpecs) are being developed and established by AUTOSAR, starting with the release 4.0. As the corresponding SWSs they will be part of the AUTOSAR standard of Release 4.0. The CTSpecs will be used, in a further step, by conformance test agencies in order to check the conformance of basic software implementations against the standard and deliver the relevant attestation to the product supplier. They will have therefore to implement conformance test suites that can execute the CTSpecs on the test benches.

CONFORMANCE TEST PROCESS

CTA accreditation [12]

The conformance test process [15] is requiring on one hand the conformance test specifications provided by AUTOSAR, and on the other hand the conformance test suite (CTS) derived from CTSpecs and implemented by conformance test agencies (CTAs). The status of CTA, as an accredited party is based on an impartial assessment of the candidate. These assessments are conducted by an independent assessor and will be repeated in reasonable intervals. Depending on the needs of the customers, CTAs can either be independent bodies delivering a third party attestation of conformance or product suppliers accredited as CTAs and delivering self declaration of conformance for their products [13], [14].

CTSpecs design [15]

The testing scope for CTSpecs is including the interfaces, the valid configurations and the behavior observable through these interfaces.

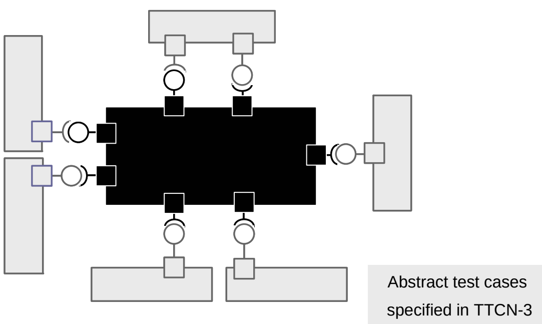

Fig 8: BSW black box testing

As shown in Fig. 8, the BSW module under conformance tests is a black box, accessible only through its interfaces by test components.

CTSpecs are designed to be accurate and stringent not allowing room for diverse interpretations, furthermore the execution of the CTSpecs on the different CTS by different CTAs should always give the same results. Therefore, the TTCN-3 language for test description has been chosen, enabling automatic execution of conformance test suites.

As shown in Fig.9, the CTAs will create the test environment necessary (test system adapter, test data converter) to perform, on their own testbenches, the conformance test of BSW module implemented for a specific target.

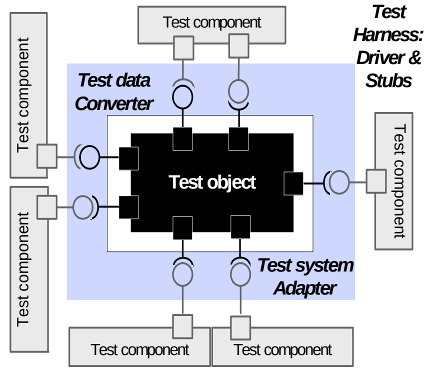

Fig. 9: Test environment designed by CTAs

CONFORMANCE TEST SPECIFICATIONS CREATION

Scope of conformance tests [15]

Currently 53 individual modules [9], [10] plus RTE are defined in the AUTOSAR BSW architecture (see Fig.10 showing a detailed view of the Communication stack).

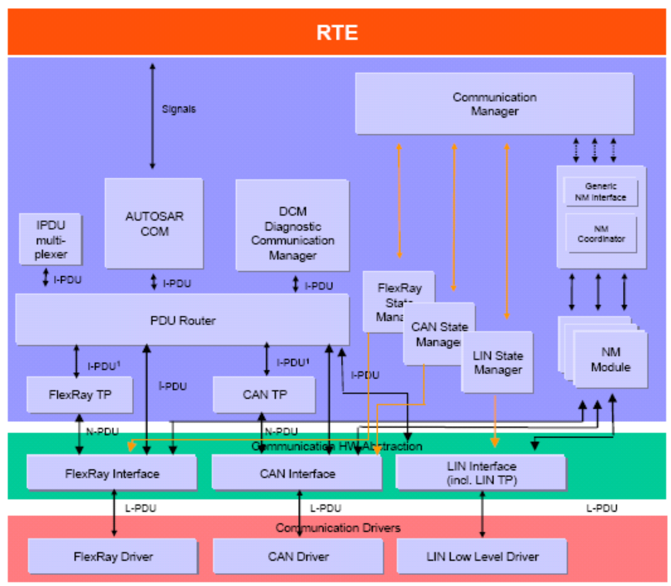

Fig. 10: AUTOSAR COM stack

AUTOSAR is developing, for the release 4.0, conformance tests at ICC3 level which is the set of those 53 BSW individual modules plus RTE. This is the level of detail given in the released basic software specifications.

Conformance test methodology

The methodology to develop conformance tests specifications was built and validated during the pilot phase conducted in 2007 on the memory stack specifications. One first result from this pilot phase was to improve the testability of specifications of the BSW modules at atomic requirement level. A second one was to make use of AUTOSAR UML models to extract automatically BSW interfaces and configuration parameters.

Finally, AUTOSAR came to the conclusion that conformance test specification creation is a very huge and challenging task, implying very different skills working in close coordination (test design and validation, basic software architecture, modeling, etc.). The complexity is comparable to the basic software specifications themselves due to the large number of modules, their wide interactions and configurability.

The CTSpec creation process [14] has been designed to fulfill the following key requirements: Division of work, division of know-how, quality assurance, work efficiency, traceability and scalability. The development has been structured in different phases that are performed for each SWS of a basic software module:

- SWS analysis, performed jointly with BSW and test experts, is mainly a review of input specifications improving testability and clarifying the specifications ambiguities,
- CTSpecs design is including extraction of testable specification items, test cases identification, configuration parameters, overall architecture of the test components, high level description of the test cases, and test coverage,
- Implementation will provide the test cases development in the TTCN-3 language,
- Test validation implementation is providing simulation of basic software modules,
- Test assessment is providing the final test case validation and ensures the test environment consistency and maintenance.

The Fig.11 shows the different roles in the CT Specification process, the workproducts are pointed by plain arrows and verification steps by dotted arrows.

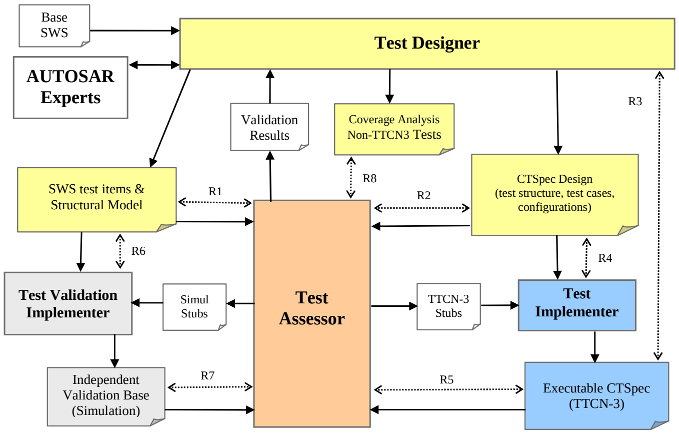

Fig. 11 CTSpecs creation process: roles and artifacts

<u>Conformance tests creation project</u>

Mid 2007, AUTOSAR launched the project for the creation of conformance test specifications. In order to meet the target of delivering the CTSpecs for the AUTOSAR release 4.0, the work has been anticipated on the current specifications starting with the most stable ones: 4 set of modules have been defined Set 1 and Set 2 from Release 3.0, those CTSpecs will stay AUTOSAR internal. Set 3 and 4 for Release 4.0. An update phase is planned for the modules of set 1 and 2.

The first CTSpecs will be given to pilot CTAs in order to start the early implementation of CT Suites, execute the tests and give feed-back to AUTOSAR.

CONCLUSION

The AUTOSAR development partnership, in which a major part of the automotive community is taking part, has brought to reality a unique and worldwide standard, which will allow accelerating developments and reducing significantly integration times. Currently, according to the defined conformance paths [13], [14], AUTOSAR products are available on the market, Tiers 1 and OEMs will bring AUTOSAR applications on the road very soon. The openness, the modularity of the standard, along with the elaboration of methodologies and processes up to conformance tests have all allowed a very rapid contribution of industrial actors at all levels – software modules, integration, tools, conformance tests – providing a further improvement of the base for the future.

DEFINITIONS, ACRONYMS, ABBREVIATIONS

ASAM: Association for Standardisation of Automation and Measuring Systems
BSW: Basic Software
CTA: Conformance Test Agency
CTSpecs: Conformance Tests Specification
CTS: Conformance Test Suite
ECU: Electronic Control Unit
E/E: Electric/Electronics
EOBD: European OBD
FIBEX: Field Bus Exchange Format
HMI: Human Machine Interface
HW: Hardware
ICC: Implermentation Conformance Class
OBD: On Board Diagnostics
OBDII: US ODB
OEM: Original Equipment Manufacturer
JOBD: Japanese OBD
RTE: Runtime Environment
SW-C: Software Component
SWS: Software specification
TTCN-3: Testing and Test Control Notation
VFB: Virtual Functional Bus
XML: eXtensible Mark up Language

REFERENCES

1. www.autosar.org Official website of the AUTOSAR Partnership

2. Heinecke, Harald et al.: AUTOSAR – Current results and preparations for exploitation. Euroforum conference May 3rd 2006

3. Fennel, Helmut et al.: Achievements and Exploitation of the AUTOSAR Development Partnership, SAE Convergence Congress, Detroit, 2006

4. Scharnhorst, Thomas et al.: AUTOSAR – Challenges and Achievements 2005. Electronic Systems for Vehicles 2005, VDI Congress, Baden-Baden, 2005

AUTOSAR SPECIFICATIONS (EXTRACT)

These specifications can be freely downloaded from the AUTOSAR Partnership Official website, for information only.
Main documents:

5. Main Requirements: AUTOSAR_MainRequirements.pdf

6. Technical Overview: AUTOSAR_TechnicalOverview.pdf

7. Specification of the Virtual Functional Bus:

AUTOSAR_SWS_VFB.pdf

8. Glossary: AUTOSAR_Glossary.pdf

Architecture documents:

9. List of Basic Software Modules: AUTOSAR_BasicSoftwareModules.pdf

10. BSW Layered Architecture:

AUTOSAR_LayeredSoftwareArchitecture.pdf

11. Specification of RTE Software

AUTOSAR_SWS_RTE.pdf

Conformance tests:

12. Conformance Test Agency Accreditation

AUTOSAR_DS_Accreditation.pdf

13. Conformance Test Process Definition Path A-C

AUTOSAR_DS_CTPathA-C.pdf

14. Conformance Test Process Definition Path D

AUTOSAR_DS_CTPathD.pdf

15. AUTOSAR Conformance Test Specification Process Overview

AUTOSAR_CTSpec_Process_Overview.pdf

Methodology and templates:

16. Methodology

AUTOSAR_Methodology.pdf

17. Specification of ECU Resource Template

AUTOSAR_ResourceTemplateECU.pdf

18. System Template

AUTOSAR_SystemTemplate.pdf

19. Software Component Template

AUTOSAR_SoftwareComponentTemplate.pdf

Application Interfaces explanatory documents:

20. ApplicationInterfaces_Explanation_BodyComfort.pdf

21. ApplicationInterfaces_Explanation_Powertrain.pdf

22. ApplicationInterfaces_Explanation_Chassis.pdf

Related standard: ASAM Fibex

23. http://www.asam.net/03_standards_06.php

CONTACT

request@autosar.org