2017 NDIA GROUND VEHICLE SYSTEMS ENGINEERING AND TECHNOLOGY SYMPOSIUM
POWER & MOBILITY (P&M) TECHNICAL SESSION
AUGUST 8-10, 2017 – NOVI, MICHIGAN

DELIVERING THE PROMISE OF AUTOSAR

Janak Patel
Senior Consultant
ETAS, Inc.
Ann Arbor, MI

ABSTRACT

In this paper, I will describe what AUTOSAR is, and the benefits it can provide in the development of ECUs. AUTOSAR provides an industry standard framework for the development of modular software architectures, including multi-core, cyber-secure, safety critical applications in the automotive/ground vehicle systems.

INTRODUCTION: KEY AUTOSAR CONCEPTS

The AUTOSAR standard, through the specification releases, describes in detail the concepts, communication mechanisms and methodology used to develop modular and standardized embedded software. At high level, the key idea of AUTOSAR is to enable the creation of well-defined software units, Software Components (SWC), that can easily be reused across vehicles or ECU projects and enable new business models between OEMs and Tier1 suppliers. Additionally, AUTOSAR provides well-defined workflows and data exchange mechanisms to allow unambiguous transfer of knowledge throughout the ECU development process.

Reusing AUTOSAR software components requires:

- Clear definition of the functional architecture of the E/E systems, to ensure the overall functional consistency
- Use of clear interfaces for each Software Components, typically emerging from the functional architecture, allowing clear separation between Software Components.
- Using well-defined communication mechanisms, fully supported and described in all technical details, that hide the target of communication from a communicating object in order to allow the implementation of communication independently of knowing how the system will be composed
- Standard services to provide an embedded software platform, defined independently from the application itself, as a common basis to enable the reuse of Software Components on different HW architectures.

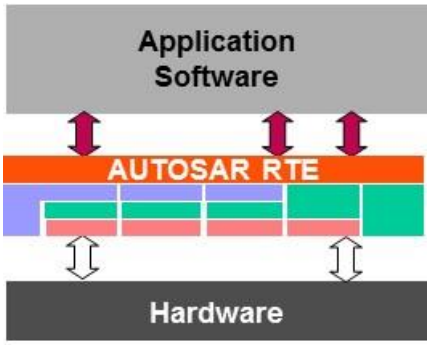

Fig 1: AUTOSAR Layered Software Architecture

The approach for developing AUTOSAR-compliant embedded software is very similar to standard System Engineering approaches applied in the automotive industry for decades: and this assimilation of legacy approaches is a key element of AUTOSAR's success. The AUTOSAR methodology is a software system specific implementation of the System Engineering Process and can easily be integrated into existing vehicle development projects.

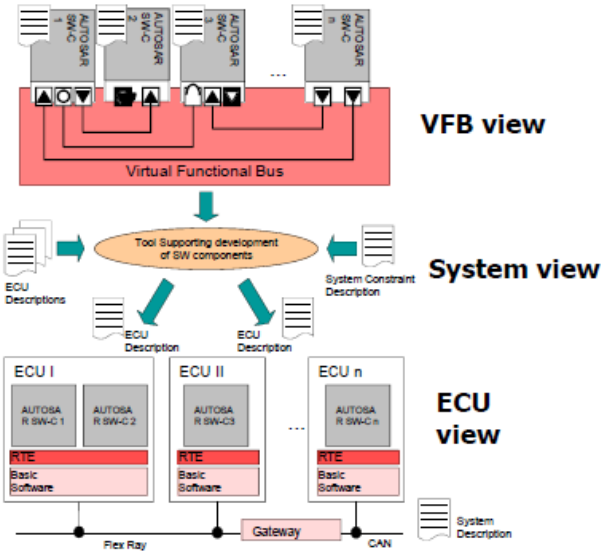

Fig 2: AUTOSAR Technical Process

OVERVIEW OF THE AUTOSAR METHODOLOGY

The scope of AUTOSAR is embedded software. AUTOSAR defines a software system from a Systems Engineering approach, i.e. consisting of sub-systems and then components that can be developed independently from the other components. In the AUTOSAR methodology description, and because software systems require a hardware platform for execution, the word "system" typically means "E/E system". There is no intention to standardize HW, but instead to integrate HW constraints (network, bus description, production costs, cabling, etc) in a systems approach while describing the functional architecture and with this, to enable different software architectures using the same software components.

The design process defined by the AUTOSAR methodology can be divided into 3 main steps:

- system design and description
- software design and development
- ECU integration

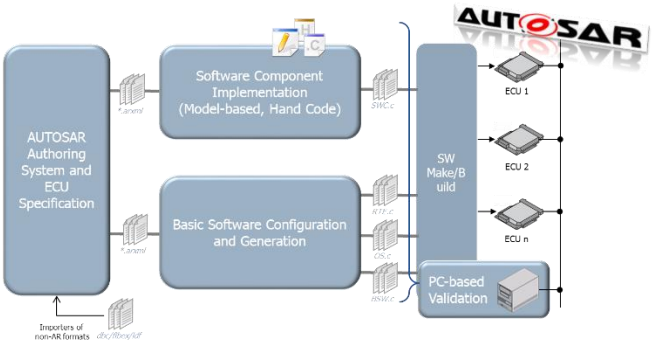

Fig 3: High level view on AUTOSAR methodology from (links to right) System description to ECU Integration

SYSTEM DESIGN AND DESCRIPTION

The methodology starts at E/E system level, with the system definition and description. At this level, the description is first purely functional, and completely independent from any technical implementation and hardware architecture. Functional components – intended to be Software Components later on in the process – are defined with their functional contribution to the E/E system and the interfaces they need in order to be integrated into the E/E system. This is a virtual description of the functions of the E/E system and the virtual communication between them. This is the "Virtual Function Bus" (VFB) view specifies what is communicated but not how (or where) it takes place.

Once the VFB view is available, system constraints, HW topology and descriptions are considered in order to map the created Software Components to HW architecture. Here

AUTOSAR defines and standardizes the description mechanisms, ensuring that all necessary HW related descriptions can be integrated and used to map the software.

From this phase the ECU description is available, meaning that the VFB is realized by a given HW configuration according to the mapping decision. Each ECU of the system can be then developed independently from the other, as dedicated sub-system with clearly defined interaction with the other sub-system. The process then to design and develop the ECU software is very similar to the one at system level.

SOFTWARE DESIGN AND DESCRIPTION

The elements of the functional architecture mapped into a given ECU are elements of the software architecture – which can typically be consisting of several elements as the technical view is not necessarily mapping the functional view (due cross domain software functions). An ECU specific functional architecture is then designed, which is the software architecture. Whereas the description is done for a single ECU, it is still done independently from the HW used. The VFB is technically implemented into a Run Time Environment (RTE) ensuring all ECU internal communication between application software and the services.

With this the ECU software architecture is available and the implementation of the defined software components is possible. There are basically 2 aspects to be considered:

- Development of Application software
- Configuration of the Basic Software modules

At this level, the Application Software Components can be developed independently from each other and the other part of the ECU software, thanks to the predefined architecture and interfaces.

The Basic Software Modules are standardized as well with AUTOSAR. They are typically available as libraries or drivers or services, and need to be configured according to the ECU and the software architecture. The application software itself is not necessary to perform this task as all requirements are included into the RTE configuration. Once configured, the Basic Software is generated and ready for the ECU integration. The major benefit of standardized Basic Software is OEMs are now no longer beholden to proprietary architectures.

ECU INTEGRATION

The last step described in the AUTOSAR methodology is the integration of Application Software and Basic Software together into the ECU. Then, the ECU with its AUTOSAR software is available with the functionalities designed at the beginning and ready for further system integration as any other ECUs.

Once the ECU software is integrated on the Hardware, the integration into the system can be performed the same way as before. ECUs with AUTOSAR-compliant software can be integrated with ECU without AUTOSAR-compliant software into the complete physical system.

The methodology defined with AUTOSAR obviously enables many different use cases compared to non-modular approach and allow implementing new processes and business models.

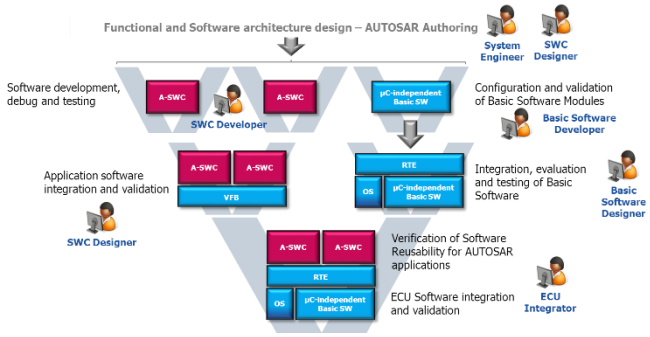

Fig 4: AUTOSAR modular architecture development and validation use cases

TECHNICAL IMPLEMENTATION OF AUTOSAR

THE VIRTUAL ECU CONCEPT

From process support perspective, and development and validation tools point of view, AUTOSAR is the enabling technology required to support virtualization of the development process.

In AUTOSAR, both Application Software and a significant part of the Basic Software are designed to be hardware independent. Only AUTOSAR's MCAL an OS modules are hardware dependant.

Consequently, most ECU software intended for production can be built and run on different hardware platforms: most notably on a standard PC with zero change.

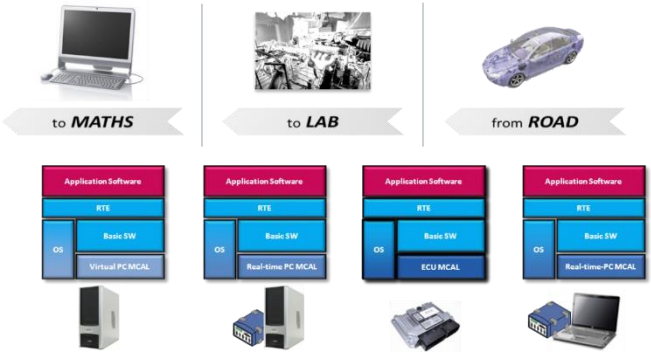

Fig 5: Virtual ECU concept: using AUTOSAR concept to support Road-to-Lab-to-Math initiatives

Providing a MCAL and OS implementation for the PC – through a "Virtual ECU" concept - makes it possible to port the complete integrated software to the PC environment and enable virtual testing of the ECU software

In order to benefit from AUTOSAR concepts it is key to ensure that the tools used for software debug and testing or on real ECU are easily connected to the Virtual ECU. It means the Virtual ECU itself shall provide the virtual interfaces to connect to the virtual system on the PC and to the tools. AUTOSAR allows this at several levels thanks to the modularity and enables tools to:

- Expose Application Software I/O for unit testing
- Expose Virtual electric signals for ECU software integration
- Expose time information to ensure the synchronization of the simulation

The time control aspect is especially the key point of the virtual validation. With a real ECU integrated into a real system, the time is given. With a virtual platform there is no "real" time anymore and components of the virtual platform typically have their own time and schedule. Without a common time for all parts of the platform and a corresponding control mechanism, the results of a simulation can be completely inconsistent.

With the timing and scheduling information from the Virtual ECU made available to the simulation tool chain the consistency is ensured. The time can then be synchronously simulated on the same basis for all components of the platform. This provides the flexibility to stop, slow down or speed up the simulation without jeopardizing the

consistency of the results, and ensuring the same behaviour than in real environment.

AUTOSAR AUTHORING

The AUTOSAR Authoring Tool is used to define the AUTOSAR architectures and describe the communication mechanism, both at functional and software level. It typically describes the AUTOSAR interfaces between the components in a standardised xml format.

In the AUTOSAR methodology, inputs and outputs from one design step to another are always defined through this arxml files. It has a significant impact on the technology to be used to take full advantage of this configuration-based process. Indeed, database should be avoided, and the Eclipse technology recommended, in order to natively supporting the methodology. Especially the AUTOSAR Authoring Tool, which is the starting point of the AUTOSAR system description, and used for system and software description, should fully support the AUTOSAR mechanism in order to ensure and facilitate the deployment of the configuration to all sub-systems impacted.

The AUTOSAR Tool Platform (Artop) based on Eclipse is an infrastructure platform implementation of common base functionality that are used to design and configure AUTOSAR compliant systems and ECUs.

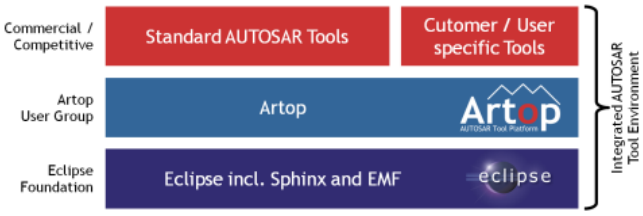

Fig 6:: Eclipse and Artop architecture

Using Eclipse and Artop together with the Virtual ECU concept, the engineers can benefit from the complete tool chain for all software development and validation tasks in one platform. Together with commercial tools such as the ETAS ISOLAR solutions, free Eclipse plug-ins are available for debug purpose or to automate the validation thanks to the test framework. The software engineer can write tests in Java or C# in the same environment and immediately apply them on the Software Component under test.

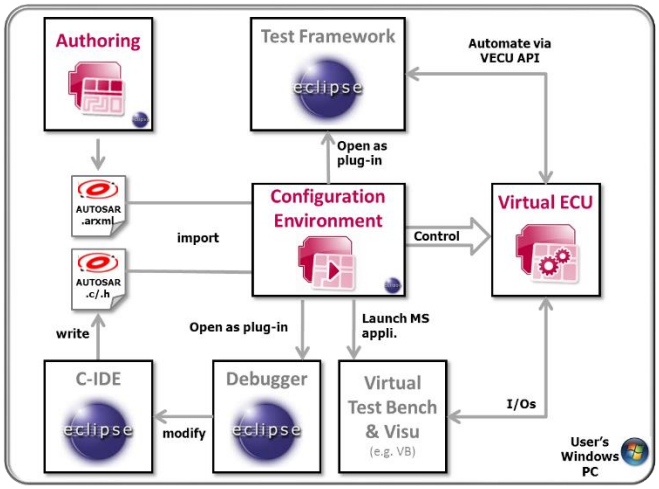

Fig 7: Eclipse-based AUTOSAR Tooling in one PC

CONCLUSION

AUTOSAR realizes many desirable software engineering concepts in an industry standard way. By both specifying a software architecture strategy and workflow methodology, AUTOSAR increases software reuse, and enables unambiguous transfer of information between the development process.