A Survey of Tools for Automotive Software Development

D.M. Ajay, Aparna.V.
School of Computer Science and Engineering, VIT University Chennai Campus, Chennai, India.

ABSTRACT

Cooperation and Business models in automotive domain are becoming more flexible and complex. In future, today’s component-driven and hardware development processes shall be hugely replaced by function-driven and a requirement process. This will require an open architecture as well as exchangeable and scalable software modules. These issues are very hard to handle by individual companies. So it a challenge to the whole automobile industry, which resulted in the establishment of an open standard for automotive E/E architecture by the decision of leading OEM’s and Tier 1 suppliers jointly called the AUTOSAR development cooperation. In this paper, we discuss in detail the necessity of AUTOSAR, the architecture of AUTOSAR and study the working and comparison of different AUTOSAR implementation tools.

Keywords: ASE- Automotive Software Engineering, AUTOSAR, ECU- Electronic Control Unit, OEM- Original Automobile Equipment Manufacturers, E/E- Electrics /Electronics Architectures, ASC – Application Software Components, RTE – Run Time Environment.

I. INTRODUCTION

The field of Automotive Software Engineering is one the highly developing field in recent times. As already mentioned the cooperation and business models in automotive domain are becoming more flexible and complex. In future, today’s component-driven and hardware development processes shall be hugely replaced by function-driven and a requirement process. Also engineering will aim at system level optimization rather than single component optimization. Mostly vehicle functions should be tested in intensive road trials, that is, a given function should be available as a real time experience in the shortest time frame possible. Due to the advent of innovation in vehicle applications, contemporary automotive E/E architecture has reached a stage of complexity that will require a breakthrough in the technology which can manage it satisfactorily and fulfil the legal and heightened passenger requirements. On top of all, the automotive field is the place where two opposing factors collide, namely mass production and safety-critical applications output resulting in an enormous cost pressure. So it is huge challenge to the whole automobile industry, which resulted in the establishment of an open standard for automotive E/E architecture by the decision of leading OEM’s and Tier 1 suppliers jointly called the AUTOSAR development cooperation.

II. AUTOSAR

1. INTRODUCTION

Automotive Open System architecture has been formed by leading OEM’s and Tier 1 suppliers in the automobile industry with a common objective to create a basis for industry collaboration on basic functions by providing a platform which encourages competition on innovative functions. The fundamental idea of AUTOSAR is the re-use of software components to handle the future increase in complexity. This standard in the future will serve as a platform upon which vehicle
applications will be implemented. It also will serve the current barriers between functional domains by helping in minimizing them. It will be possible to independently map functions and functional networks to various different control nodes from the hardware. A joint technical team was formed in November 2002, for establishing the technical implementation strategy. The core partners are Ford Motor Company, Peugeot Citroen Automobiles S.A, Toyota Motor Corporation, General Motors, VDO, and Siemens.

2. AUTOSAR ARCHITECTURE

![Figure 1. Architecture of AUTOSAR](image)

1. AUTOSAR is a layered architecture which decouples functionality from the hardware and software services.
2. This architecture consists of totally 3 layers.
   a. BSL – Basic Software Layer: This layer provides hardware dependent and independent services to RTE using API. It makes upper most layers hardware independent.
   b. RTE - Run Time Environment: Functionality is to connect the Application Software Components to the right hardware.
   c. Application Layer: This layer is where the actual functionality is situated. It is composed of ASC that interact with the RTE.

3. METHODOLOGY USED IN AUTOSAR

3.1 Input description

1. Software Components: Independent of implementation of the components. Data specified are interfaces and the hardware requirements.
2. System: Interconnections between ECU’s (System Topology) need to be specified.
3. Hardware: Specifications of available hardware with programming capabilities and signal processing methods.

3.2 System configuration

Distributing the description of software components to the different ECU’s in a iterative process is the main task of this step.
3.3 **ECU – Configuration**

The Basic Software and RTE of each ECU are configured. This is done based on the allocation of the ASC to each ECU.

3.4 **Generation of software executables**

1. Software executables are generated based on the previous steps, implementation specification of each software component is necessary.
2. Tool chains are used to automate this methodology. Each and every subsequent methodology up to generation of executables supported by defining exchange formats using XML.
3. AUTOSAR methodology is supported by developing a meta-model.

3.5 **Motivation of AUTOSAR**

1. The effective management of E/E complexity in association with the growth in functional scope.
2. Flexibility in product upgrading, updating and modifying.
3. Scalability within and across product line solutions.
4. Improving the quality and reliability of E/E systems.

3.6 **Goals of AUTOSAR**

1. Scalable systems cost optimization.
2. Improved mechanisms for the containment of product and process complexity and risk.
3. Fulfilment of future vehicle requirements namely safety and availability, Software update/upgrade and maintainability.
4. Increased flexibility and scalability to transfer and integrate functions.

3.7 **Advantages of AUTOSAR**

1. Modularity, Scalability, Transferability, Re-usability.
2. Reduce number of ECU’s.
3. Reduce increasing number of mechanical systems.
4. Getting grip of the distributed development process.

Integration, update, upgrade of different ECU’s by different Tier-1 suppliers made easy for the OEM’s.

4. **IMPLEMENTORS OF AUTOSAR AND TOOLS**

Based on the AUTOSAR – paradigm several software suppliers offer implementations of AUTOSAR standards. Some of them are

4.1 **ARCCORE**

ArcCore is an implementer of AUTOSAR. It provides tools like Arctic Core for BSW implementation, BSW Builder tool which is used as BSW configurator, RTE Builder tool for RTE Generation, SWC Builder tool and Extract Builder tool for System Tooling. Arctic Core and base version of Arctic studio is an open source and all the other tools are commercial.
4.2 BOSCH

Bosch is a pioneer in AUTOSAR implementation. It provides tools like CUBAS, iSolar for BSW implementation, the same tools work as BSW configurator and RTE generator also. Bosch doesn’t provide any tool for System Tooling. All the tools provided by Bosch are commercial.

4.3 VECTOR INFORMATIK GMBH

Vector Informatik Gmbh provides tools for AUTOSAR implementation like MICROSAR tool for BSW implementation, DaVinci Configurator tool Pro for Configuring the BSW, MICROSAR Rte Generator tool for RTE Generation, DaVinci Developer/ DaVinci System Architect tool for System Tooling. All the tools provided by this implementer are commercial tools.

The detailed description is given below in Table. I

<table>
<thead>
<tr>
<th>Implementer</th>
<th>BSW Implementer</th>
<th>BSW Configurator</th>
<th>RTE Generator</th>
<th>System Tooling</th>
<th>License</th>
</tr>
</thead>
<tbody>
<tr>
<td>ArcCore</td>
<td>Arctic Core</td>
<td>BSW Builder</td>
<td>RTE Builder</td>
<td>SWC Builder, Extract Builder</td>
<td>GPL (Arctic Studio) / Commercial licenses available for all products</td>
</tr>
<tr>
<td>Bosch</td>
<td>CUBAS, iSolar</td>
<td>CUBAS, iSolar</td>
<td>CUBAS, iSolar</td>
<td>Unknown</td>
<td>Commercial</td>
</tr>
<tr>
<td>Vector Informatik GmbH</td>
<td>MICROSAR</td>
<td>DaVinci Configurator Pro</td>
<td>MICROSAR Rte Generator</td>
<td>DaVinci Developer / DaVinci System Architect</td>
<td>Commercial</td>
</tr>
</tbody>
</table>

5. SURVEY OF TOOLS IMPLEMENTING BSW (BASIC SOFTWARE)

5.1 Arctic Core Tool

This tool supports in building AUTOSAR 3.1 System. Multiple architectures are available Freescale MPC551x, MPC5554, MPC5567, MPC560x, MPC5568, MPC563x, HCS12, Arm Cortex M3 and Arm Cortex R4. It is licensed under GPLv2. We do have commercial license. Mercurical Version control to maintain the versions. Misra C is used which aims are to facilitate code safety, portability and reliability in the context of embedded systems.

5.2 iSolar Tool

This tool supports in building AUTOSAR 2.0 System upwards. It has an open extensible architecture. It is used mainly for commercial purpose. The Version maintenance is tool specific and Misra C is not used in this tool.

5.3 Microsar

This tool supports in building AUTOSAR 3.1, 4 System. It supports the following architectures FRIF, FRNM, FRSM and a choice between FRTP and FRISOTP. MICROSAR FR can be extended with XCP as an option. It is used mainly for commercial purpose. The Version maintenance is tool specific and Misra C is not used in this tool.
The detailed description is given in the Table. II.

<table>
<thead>
<tr>
<th>Tool</th>
<th>Support in Building AUTOSAR</th>
<th>Supported Architectures</th>
<th>License</th>
<th>Version Control</th>
<th>MISRA C</th>
</tr>
</thead>
<tbody>
<tr>
<td>Arctic Core</td>
<td>3.1 System</td>
<td>MPC551x, MPC5554, MPC5567, MPC560x, MPC5568, MPC563x, HCS12, Arm Cortex M3 and Arm Cortex R4</td>
<td>GPLv2 license</td>
<td>Mercurial version control</td>
<td>YES</td>
</tr>
<tr>
<td>iSolar</td>
<td>2.0 System upwards</td>
<td>Open, extensible architecture</td>
<td>Commercial</td>
<td>Tool Specific</td>
<td>NO</td>
</tr>
<tr>
<td>Microsar</td>
<td>3.1, 4 System</td>
<td>FRIF, FRNM, FRSM and a choice between FRTP and FRISOTP. MICROSAR FR can be extended with XCP as an option.</td>
<td>Commercial</td>
<td>Tool Specific</td>
<td>NO</td>
</tr>
</tbody>
</table>

REFERENCES


