

White Paper



Software as a Commodity Product

The AUTOSAR effect

Uday H. Prabhu, Praveen Bhatt

Abstract

The Automotive Open System Architecture (AUTOSAR) is a move by the automotive OEM conglomerate to bring about a common method of developing software for the automobile-common to the extent that it creates an open standard methodology for software architecture. Creation of standards drive companies to stick to rules and thereby ensures that the software remains the same when developed by anyone conforming to specifications. Now with all factors equal, the only differing factor will be quality and price. This is where software has become a purchasable commodity in a Bill of Materials (BOM) since it conforms to an industry specification. Just like a tyre with a specified wheel width, size and speed specifications can be bought from Bridgestone or Michelin or Goodyear, software can be bought from Bosch or Conti or Vector or from Infosys or even an unknown small new company making it a commodity.

Software programs in fundamental areas such as hardware abstraction layers, device drivers, signal processing, message routing, error handling and data acquisition are more or less standard and common across Electronic Control Unit (ECU) applications. These can be termed as the basic building blocks of software. These blocks do not see many changes over the lifecycle of a product or a platform. In the automotive industry every company at the Tier-1 level has its own version of how this layer should be written-this includes different coding styles, internal standards, varying software architectures, differing modular concepts, historical and legacy code inheritances and so on.

At an OEM level it was therefore seen that multiple variants and designs of software existed for similar applications. An application, therefore, had to be modified many times to cater to these variants. This resulted in time spent resolving issues in the interactions to the hardware dependent layers rather than concentrating on the application at hand. AUTOSAR has managed to create the commodity concept with regards to automotive software. This paper explores how software can be purchased as a commodity by automotive system manufacturers.

For more information, Contact askus@infosys.com.

AUTOSAR (AUTomotive Open System ARchitecture)

AUTOSAR is open automotive software architecture, standardized and jointly developed by automobile manufacturers, suppliers and tool developers. It paves the way for innovative electronic systems that further improves performance, safety and environmental friendliness. It stands for a strong global partnership that creates one common standard: "Co-operate on standards, compete on implementation." The concept of AUTOSAR is illustrated in the diagram below. The software parts are highlighted at the very end.

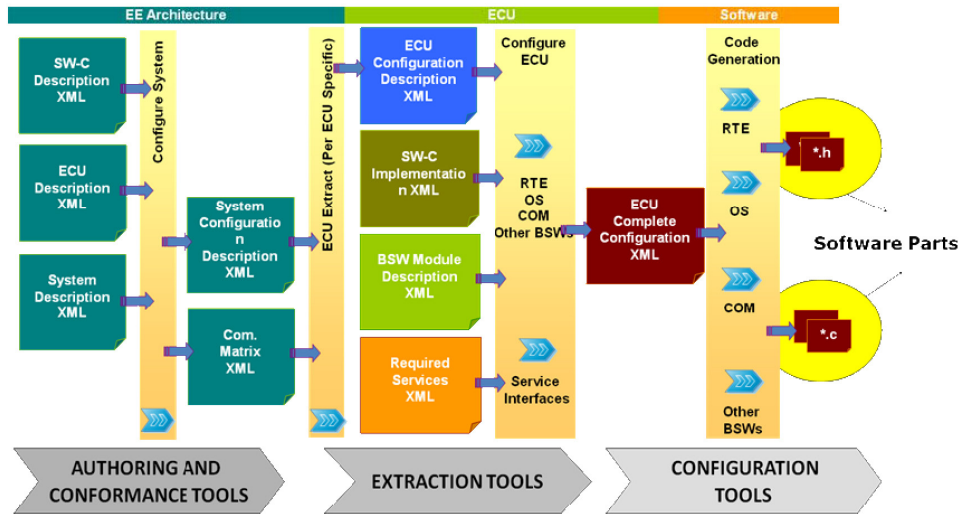


Figure 1: AUTOSAR Development Architecture

Commodity Product

Commodity is defined as a product or a set of goods, which has a constant demand and is supplied *“without any qualitative differentiation”* within a market. This product remains the same no matter who manufactures/develops/produces it. Some examples are steel, milk, paper, pencils etc. The price of this product is typically a universal constant but it fluctuates based on global supply and demand. There could be variants to the product but it is important to note that these are well-defined and bounded in terms of specifications.

One of the characteristics of a commodity product is that its price is determined as a function of its market. There are various kinds of commodities based on the manufacturing or development process. Basic commodities, Soft Commodities, Hard commodities etc. are some examples.

“A product or a service enters into the commodity segment because it loses the ability to differentiate itself.”

It also becomes an increasingly low or zero investment area because infusion of intellectual capital does not return adequate change in its properties or efficiency. Therefore these commodities tend to veer towards a price sensitive market place. Looking at the bell curve of innovation and capital spending, commoditization happens towards the end of the cycle.

Software as a Commodity

With the background of AUTOSAR and the concept of a commodity now clearly defined, let us look at how functional parts of automotive software have become commodity items.

Let us take the example of a basic Controller Area Network (CAN) communication protocol stack as illustrated below.

The software stack comprises three key building blocks

- The Communication Drivers Layer
- The Communication Hardware Abstraction Layer (HAL)
- The Communication Services Layer

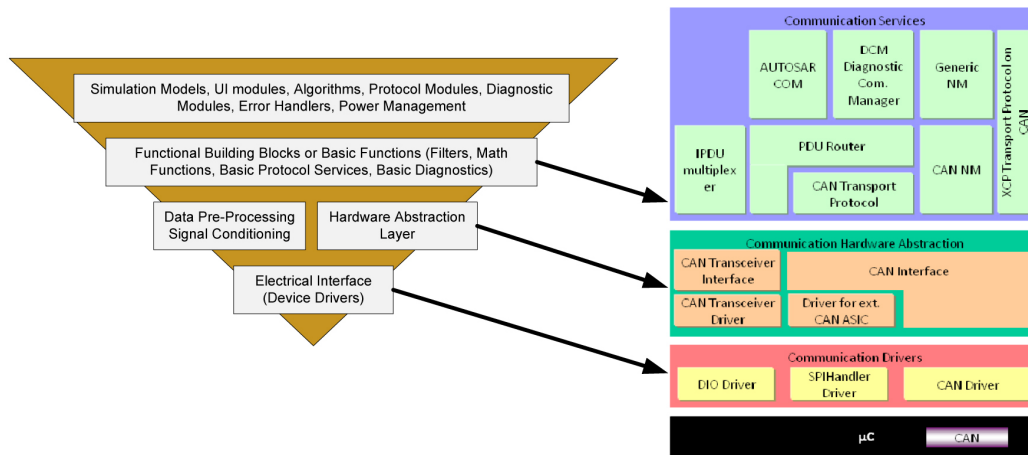


Figure 2: Typical Software Architecture and Basic Building Blocks

As per AUTOSAR, majority of the three blocks (highlighted above as Drivers, HAL and Functions) are extremely well-defined and specified. Therefore, it is realistic and practical to state that these software modules will be 99% identical, when developed by different companies, all of whom will adhere and refer to the AUTOSAR specifications.

Now with all things being common amongst the companies - AUTOSAR specifications, automotive guidelines, conformance testing frameworks, quality standards etc., the only differentiating factor will be price.

Software is now a commodity!

Overall Process Flow

Let us now look at how this can be realized in a real-life scenario. Purchasing departments have two ways of soliciting competitive bids from suppliers. They are as follows

- Request for Bid (RFB)
- Request for Proposal (RFP)

For either of the above two options, the following process work-flow explains how software can be defined, ordered, evaluated and finally procured.

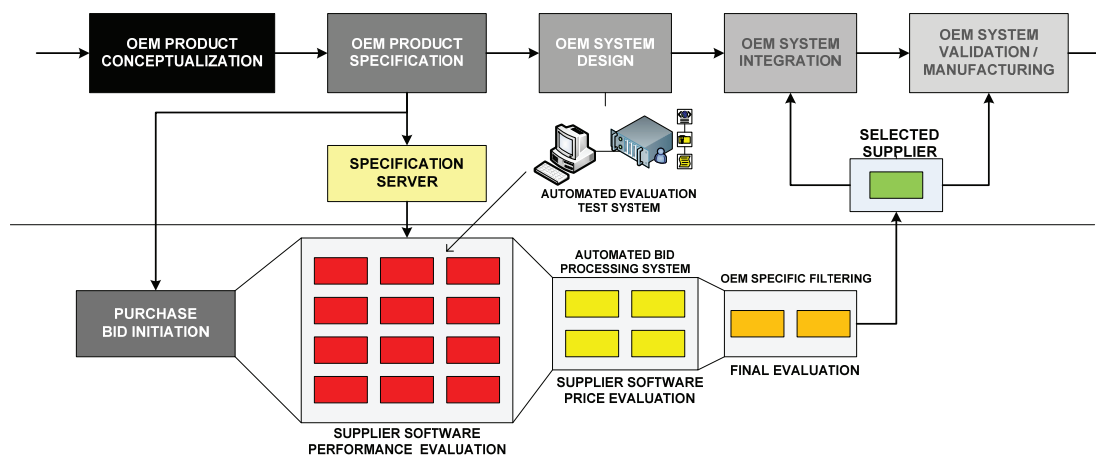


Figure 3: Software Purchase and Evaluation System

a) Product Conceptualization

- This is the typical R&D platform development phase, where interfaces and performance specifications are defined and documented.
- The AUTOSAR framework will be defined here along with the various components that are applicable for the platform.

b) Product Specification

- When AUTOSAR has been chosen as the framework, the software components will be defined here with reference to the AUTOSAR specification releases
- The specification thus defined will be sent to the Purchase department for “*software purchase*”
- Purchase shall initiate an RFP/RFB depending on the situation and release the specifications or reference to the AUTOSAR specification
- Criteria for evaluation and performance parameters are defined in relation to the specification.

c) System Design

- While the purchase process is in progress, the system design continues with the remainder of the system components.
- The test system and evaluation system are developed and tested to evaluate the incoming components.
- The other parameters of the system are developed and prototyped as applicable.

d) Evaluation of supplier responses

- Once the RFP/RFB closes, the bids/proposals received are evaluated through a three-step process:
 - Step 1: This is a performance evaluation using an automated test system that proofs the submitted software for functional, behavioral and performance requirements. The software is automatically integrated into an automated test system that runs a set of test scripts to evaluate functional, performance and non-functional requirements. This is the key part of the entire evaluation process.
 - This is evaluated using pre-defined test scripts that check for performance parameters, response times, critical sections, fault scenarios, error handling and other key functions. An automation reports generator provides a detailed evaluation report. Suppliers are selected or rejected based on these results. Data is shared with all.
 - Step 2: The shortlisted suppliers are then put through an online price bidding activity. The result will be the lowest cost supplier amongst the shortlisted ones.
 - Step 3: The final set of suppliers are then evaluated on a few key customer requirements that could be OEM/Tier 1 specific. The final supplier gets to supply the software and support the integration activity of the system.
- The important part to be noted in the above process is that evaluation criteria and system used is shared with all respondents, and the results are also shared post the evaluation. This ensures a very clear and transparent process of selection with absolutely no ambiguities.
- The evaluation results are available online to see through an evaluation summary document. This is created for each response showing points / scores awarded, and the reasons for the allocation / deduction of the said points.
- Based solely on the evaluation criteria, the successful supplier is selected.
- Purchasing notifies the successful supplier and issues the Contract/Agreement and/or Purchase Order.

e) System and Software Integration

- The software is automatically integrated into the system and validated.
- There is typically no requirement for a supplier to be part of the process as the software is defined as per precise requirements.

Advantages and benefits of the commodity concept

Quality

The quality of the supplied code is in conformance with specified standards and guidelines with no room for ambiguity or mis-interpretation. There is however a need for AUTOSAR to constantly seek granular reviews of its specifications for areas of improvement and increased clarity. Any ambiguity will lead to variants dictated by the customers to their advantage or convenience.

Regularization

Pre-specified code bases are extremely consistent. With the advent of code generators and configurators, bugs are fixed in the template itself and bug-fixes are propagated across all of the code consistently, so the output code is always better maintained than its hand-coded equivalent. By incorporating industry standards, the generated code will automatically lead to standardization.

Productivity

The productivity of the systems engineering group and the software department is unquestionably enhanced by code purchases. The engineers will be making intelligent use of their time to develop better systems and not worry about the software components that make up the architecture. Productivity will increase every time a software component is packaged and well-documented towards its commoditization.

The below table provides a few indicative improvement numbers, which could be realized with an efficient implementation of the proposed concept.

Performance Parameter	Improvement	Details of Performance Benefits
Reuse Management	20-25%	<ul style="list-style-type: none">• 100% reuse of common software components and functional blocks.• Complete framework reuse for variants development.
Productivity	25-30%	<ul style="list-style-type: none">• Core pre-tested components of the code will be used in a plug-and-play fashion during integration.• Prove out time will be drastically reduced as the core components to be integrated are pre-tested and simply integrated.
Delivered Defects Density per Program	>40%	<ul style="list-style-type: none">• Defect density reduced drastically, as pre-tested software components are re-used.• Defects are handled by the supplier with detailed specifications.
Schedule Improvements	>35%	<ul style="list-style-type: none">• On an average, schedule will shrink to approximately 65% to 70% of original schedule due to easy availability of a large number of software components that make up the functional building blocks.

Conclusion

It has been decades since AUTOSAR first realized the importance of coming up with a common open architecture for the automotive market. Since the first software application was developed by referring to a specification document of automotive companies, there have been misinterpretations, software bugs and hardware bugs resulting in millions of dollars of rework, re-fabrication and recalls. Eliminating the bugs from the basic software have resulted in delays and stretched development schedules. With the increasing advantage and relevance of AUTOSAR in the automotive market and the strides made in standardization of software practices, the standardization of software and commoditization of the same is the next big step towards reliable just-in-time software integration.

References

1. www.autosar.org The Official AUTOSAR Website

About the Authors

[Uday H. Prabhu](#) is a Group Project Manager at Infosys Limited and is based out of Mysore, India. He comes with more than 17 years of experience in the Engineering industry and automotive electronics development. He is part of the Infosys AUTOSAR group, which is a premium member of the AUTOSAR consortium.

[Praveen Bhatt](#) is a Project Manager at j Limited and is based out of Bangalore, India. He manages the Automotive and Electronics practice in the Infosys Engineering group and possesses more than 10 years of experience in developing embedded software. He can be contacted at praveen_bhatt@infosys.com.



For more information, contact askus@infosys.com

About Infosys

Many of the world's most successful organizations rely on Infosys to deliver measurable business value. Infosys provides business consulting, technology, engineering and outsourcing services to help clients in over 30 countries build tomorrow's enterprise.

For more information about Infosys (NASDAQ:INFY), visit www.infosys.com.