AUTOSAR -
A standardized automotive software architecture

Helmut Fennel
AUTOSAR
A standardized automotive software architecture

Helmut Fennel, AUTOSAR Spokesperson

OOP 2007
Munich, 24th January 2007

Contents

- Introduction
  - Challenges in automotive E/E development
  - What is AUTOSAR?
- Technical concept of AUTOSAR
  - Architecture concept
  - Methodology & metamodel
  - ECU layered software architecture
  - Configuration
  - Error handling
- AUTOSAR specification status at end of phase I
- AUTOSAR phase II
- Outlook
  - Benefits and changes
Challenges in Automotive E/E Development

- Extend product offering and increase product differentiation
- Stable or decreasing development costs
- Strengthen brand image in the market
- Propose specific features and functions across the product range
- Ensure long term competitiveness, as well as presence in emerging markets, through cost reduction
- Increase quality and reduce “non quality” costs

Increasing share of electronics in vehicle value

- Electronics share (in value): 2004: 20% → 2015: 40%
  (McKinsey, Automotive Electronics - Managing innovations on the road)
- Software share (in value): 2000: 4,5% → 2010: 13%
  (Mercer Consulting, Automobile technology 2010)

Automotive Electronics will increase due to improved Performance and Efficiency of Semiconductors

References:
1) Fischer (Uni MH) 2004
2) Moore (Intel) 2003, Moravec 2005
3) Isaac (IBM) 2001
4) Kurzweil, Moravec, nach 3)
AUTOSAR Managing Complexity by Exchangeability and Reuse of Software Components

AUTOSAR Project Objectives and Key Topics

**Project Objectives**

- Implementation and standardization of basic system functions as an OEM wide “Standard Core” solution
- Scalability to different vehicle and platform variants
- 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
- Consideration of availability and safety requirements
- Redundancy activation

**Topics**

- Basic Software
- Methods of Software Integration
- Functional APIs
AUTOSAR Key Topics

AUTOSAR provides three main areas of results:

**Basic Software:**
Software architecture including a complete basic (environmental) software stack for an ECU as an integration platform for hardware independent SW applications

**Methods of Software Integration:**
Exchange formats (templates) to enable a seamless configuration process of the basic software stack and the integration of application software in ECUs

**Functional API:**
Specification of functional interfaces as a standard for application software modules

---

AUTOSAR – Partnership Structure

**10 Core Partners (OEM & Tier 1 Supplier)**
- Organizational control
- Technical contributions
- Administrative control
- Definition of external Information (web-release, clearance, etc.)
- Leadership of Working Groups
- Involvement in Working Groups

**52 Premium Members**
- Leadership of Working Groups
- Involvement in Working Groups
- Technical contributions
- Access to current information

**44 Associate Members**
- Access to finalized documents
- Utilization of AUTOSAR standard

**Support Roles**
- 3 Development Members
- 17 Attendees

Membership status as of Dec 4th, 2006
AUTOSAR – Core Partners and Members

Status: 4th December 2006

10 Core Partners

44 Associate Members

52 Premium Members

General
OEM

Generic
Tier 1

Standard
Software

Tools and
Services

Semi-
conductors

AUTOSAR – Facts & Figures

- Complex project organization and management
  - Decision finding based on unanimity

- Established processes, e.g.:
  - Reporting and monitoring
  - Configuration management
  - Review process

- 132 Deliverables for Release 2.1
- approx. 175 Full Time Equivalents in Peak

- 90 Deliverables for Release 2.0

- 46 BSW-Modules for Release 2.1

- 26 active WP / WG in Phase I

- approx. 650 Experts involved
Use Case ‘Front Light Management’ mapped to AUTOSAR architecture

Exchange of type of front light and consequences to the Use Case mapping
Functional software is described formally in terms of “Software Components” (SW-C).

Using “Software Component Descriptions” as input, the „Virtual Functional Bus“ validates the interaction of all components and interfaces before software implementation.

Mapping of “Software Components” to ECUs and configuration of basic software.

The AUTOSAR Methodology supports the generation of an E/E architecture.
AUTOSAR Metamodel

Formal description of all methodology related information

- The metamodel is modeled in UML
- The structure of the information can be clearly visualized
- The consistency of the information is guaranteed
- Using XML, a data exchange format can be generated automatically out of the metamodel

AUTOSAR Metamodel

The AUTOSAR Metamodel
- is the backbone of the AUTOSAR architecture definition
- contains complete specification, how to model AUTOSAR systems

M0: Realized System in the car (Implements a real system)
M1: Model of the system (Defines a real system)
M2: Model of the model (Metamodel) (Defines AUTOSAR Modeling Elements)
M3: Model of the Metamodel (Meta-Metamodel) (Defines UML Modeling Elements)
AUTOSAR Metamodel and Methodology

Methodology
- defines activities and work-products
- is integrated in the metamodel

Metamodel defines the contents of the work-products
- Formal description of all the information that is produced or consumed in the AUTOSAR methodology
- Benefit of using the metamodel:
  - No inconsistencies
  - Easy maintenance
  - Consistent terminology

AUTOSAR Layered ECU Software Architecture
AUTOSAR Layered ECU Software Architecture

AUTOSAR Runtime Environment (RTE)

- System Services
  - Onboard Device Abstraction
  - Microcontroller Drivers

- Memory Services
  - Memory Hardware Abstraction
  - Memory Drivers

- Communication Services
  - Comm. Hardware Abstraction
  - Communication Drivers

- I/O Services
  - I/O Hardware Abstraction
  - I/O Drivers

- Complex Services
  - Complex Device Drivers

BSW-Layers

- Services
  - ECU Abstraction and Complex Drivers
  - Microcontroller Abstraction

Microcontroller

AUTOSAR ECU Software Architecture

AUTOSAR Runtime Environment (RTE)

- AUTOSAR Interface
  - AUTOSAR Interface
  - Standardized Interface
  - Operating System
  - Microcontroller Abstraction

AUTOSAR Software

- Application Component
  - AUTOSAR Interface
  - Standardized Interface
  - Services

- Actuator Software Component
  - AUTOSAR Interface
  - Standardized Interface
  - Communication

- Sensor Software Component
  - AUTOSAR Interface
  - Standardized Interface
  - Standardized Interface

AUTOSAR Interface

- Standardized Interface
  - Services
  - Communication

Basic Software

ECU-Hardware

- API 2
  - VFB & RTE relevant
  - Standard Interfaces inside Basic Software possible
- API 1
  - RTE relevant
- API 0
Abstraction inside the Infrastructure Architecture

Example: Memory stack

Equal mechanisms to access internal (on-chip) and external (on-board) memory devices and type of memory hardware (EEPROM, Flash)

Abstracts from the location of peripheral memory devices (on-chip or on-board) and the ECU hardware layout.
Abstraction inside the Infrastructure Architecture
Example: CAN communication stack

CAN Communication Services provide a uniform interface to the CAN network. Hide protocol and message properties from the application.

Extended Configuration Concept
The Generic Configuration Editor

- Schema - Configuration Parameter Template
- Schema - Configuration Parameter Template
- XML – ECU Configuration Description
Dedicated BSW Module

module configuration

module generation

- Schema - ECU Configuration Template

conforms to

conforms to

conforms to

Provided by AUTOSAR
Provided by Implementer
Automatically generated

AUTOSAR Integrator
Error Handling
Diagnosis during development and normal operation

Key innovations
- Consistent and unambiguous error handling
- Supports diagnostics during implementation and integration phase → DET
- Supports monitor and control needs during normal operation → DEM

DEM: Diagnostic Event Manager
DET: Development Error Tracer

Technical scope of AUTOSAR

New concepts

Industry-wide consolidation of 'existing' basic software designs
Content of AUTOSAR Release 1.0, 2.0, and 2.1

<table>
<thead>
<tr>
<th>AUTOSAR Runtime Environment (RTE)</th>
</tr>
</thead>
<tbody>
<tr>
<td>System Services</td>
</tr>
<tr>
<td>Memory Services</td>
</tr>
<tr>
<td>Communication Services</td>
</tr>
<tr>
<td>I/O Hardware Abstraction</td>
</tr>
<tr>
<td>Complex Drivers</td>
</tr>
<tr>
<td>Microcontroller</td>
</tr>
<tr>
<td>I/O Drivers</td>
</tr>
<tr>
<td>Memory Drivers</td>
</tr>
<tr>
<td>Communication Drivers</td>
</tr>
<tr>
<td>I/O Hardware Abstraction</td>
</tr>
<tr>
<td>Communication HW Abstraction</td>
</tr>
</tbody>
</table>

**Release 1.0**
- I/O communication
- CAN stack
- FlexRay stack
- Diagnostic routines (non OBD relevant)
- Memory stack
- Operating system

**Release 2.0 Additions**
- Runtime Environment
- LIN stack
- Gateway functionality
- ECU services
  - Watchdog manager
  - ECU state manager

**Release 2.1 Updates**
- BSW configuration
- Processes
- Conformance Testing
- Change & Release Management

AUTOSAR Status at end of Phase I

- **AUTOSAR Release 2.0 specifications**
  - Published in May 2006, for information only
  - Available for download – [www.autosar.org](http://www.autosar.org)

- **AUTOSAR Release 2.1**
  - Scheduled for end of 2006
  - Will also be published and available for download

- **Promote AUTOSAR as a de-facto standard**
- **Allow for open communication on specifications and AUTOSAR contents**
AUTOSAR Phase II 2007 – 2009

AUTOSAR Development Partnership will continue
- Identical Core Partners
- Completion of contracts with phase I members in progress
- Exploitation and maintenance
  - Already in 2008 the first cars with AUTOSAR technology inside
  - All Core Partners have planned the introduction of AUTOSAR products until 2010
  - Establish conformance test process
- Further development and amendment of the standard, e.g.
  - Safety features
  - Vehicle & application mode management
  - Debugging
  - Standardization of application interfaces

Phase II – Work package structure

Working Groups

- WPII-1 System Architecture
  - WPII-1.1 Software Architecture
  - WPII-1.2 Vehicle and Application Mode Management
  - WPII-1.3 Functional Safety
  - WPII-1.4 Debugging

- WPII-2 Software and Test Specification
  - WPII-2.1 Basic Software
    - WPII-2.1.1 COM Stack
    - WPII-2.1.2 FlexRay
    - WPII-2.1.3 MCAL
    - WPII-2.1.4 Diagnostics
    - WPII-2.2 Conformance Test Specification

- WPII-3 Validation
  - WPII-3.1 Basic Software Validation
  - WPII-3.2 Template Validation

- WPII-4 Enabling Exploitation
  - WPII-4.1 Follow-up Organization

- WPII-5 Maintenance of Releases
  - WPII-5.1 Change and Release Management
  - WPII-5.2 Maintenance of Specifications

- WPII-10 Application Interfaces
  - WPII-10.0 Coordination of Appl. Interfaces
  - WPII-10.1 Body and Comfort
  - WPII-10.2 Powertrain
  - WPII-10.3 Chassis Control
  - WPII-10.4 O&P Safety
  - WPII-10.5 MM / T / HMI
Standard Maintenance

- **Change Management**
  - Controlled handling of Request for Changes (RfC) involving expert evaluation where necessary
- **Release Management**
  - Synchronization of complete AUTOSAR releases (set of specifications)
  - Communication about upcoming and valid releases

Conformance Testing

**Objective:** Verify that a product under test adheres to corresponding AUTOSAR specifications.

**Conformance Test Agency (CTA)**
- Accreditation by accreditation body
- Third party CTA
- or first party CTA (product supplier)

**Conformance Test Suite (CTS)**
- Serves for execution of standardized test cases
- Implements the conformance test specifications
Benefits from AUTOSAR

- OEM overlapping reuse of software modules
- Maintaining ability to compete on innovative functions, heightened design flexibility
- Simplification of the integration task
- Reduction of total SW development costs
- Reduction of version proliferation
- Development partitioning among suppliers
- Increase of efficiency in functional development
- New business models possible
- Common interfaces with development processes
- Seamless, manageable, task optimized (time dependent) tool landscape
- Transparent and defined interfaces enable new business models

Automotive Software Development will change.

- Hardware- and software will be widely independent of each other.
- Development processes will be simplified. This reduces development time and costs.
- Reuse of software increases at OEM as well as at suppliers. This enhances also quality and efficiency.

Automotive Software will become a product.
AUTOSAR Outlook

- AUTOSAR is ready to start automotive product development
- Exploitation has already started
- AUTOSAR welcomes new members in Phase II

“Cooperate on standards, compete on implementation.”

Thank you for your attention!

http://www.autosar.org

request@autosar.org