AutoSAR Overview

FESA Workshop at KTH 2010-04-12

Prof. Jakob Axelsson
Volvo Cars and
Mälardalen University

This presentation is based on a tutorial prepared by the AutoSAR Consortium
AUTOSAR – Members

Status June 2007

10 Core Partners

57 Associate Members

10 Core Partners

57 Associate Members

49 Premium Members
Exchangeability and Reuse of SW Components

Exchangeability between supplier's solutions

Supplier A
- Chassis
- Safety
- Body/Comfort
- Multimedia

Supplier B
- Chassis
- Safety
- Telematics
- Multimedia

Supplier C
- Body/Comfort
- Powertrain
- Telematics
- Multimedia

Exchangeability between manufacturer's applications

OEM a
- Platform a.1
- Platform a.2
- Platform a.n

OEM b
- Platform b.1
- Platform b.2
- Platform b.n

OEM c
- Platform c.1
- Platform c.2
- Platform c.n

OEM d
- Platform d.1
- Platform d.2
- Platform d.n

OEM e
- Platform e.1
- Platform e.2
- Platform e.n

OEM f
- Platform f.1
- Platform f.2
- Platform f.n
Changing Automotive SW Development

- **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 Main Working Topics

- **Architecture:**
  Software architecture including a complete basic or environmental software stack for ECUs – the so-called AUTOSAR Basic Software – as an integration platform for hardware independent software applications.

- **Methodology:**
  Exchange formats or description templates to enable a seamless configuration process of the basic software stack and the integration of application software in ECUs and it includes even the methodology how to use this framework.

- **Application Interfaces:**
  Specification of interfaces of typical automotive applications from all domains in terms of syntax and semantics, which should serve as a standard for application software.
Intra- and Inter-ECU Communication

- Ports implement the interface according to the communication paradigm (here client-server based).
- Ports are the interaction points of a component.
- The communication is channeled via the RTE.
- The communication layer in the basic software is encapsulated and not visible at the application layer.

![Diagram of Intra- and Inter-ECU Communication](image)
AUTOSAR Methodology

VFB view

Standardized description templates for application software components (interfaces and BSW requirements)

Standardized exchange formats and methodology for component, ECU, and system level

Tools for
- support of component mapping
- generation of RTE, i.e. inter- and intra ECU communication

Standardized Basic Software (BSW) architecture, detailed specifications for implementation and configuration of BSW
AutoSAR Descriptions

**ECUs**
- ECU Resource Description
- ECU Resource Description
- ECU Resource Description

**Supported protocols:**
- CAN, LIN, FlexRay

**System Description**
- CAN
- LIN
- LM-L
- LM-R
- BC-V
- BC-H
- SMLS

**Software Components**
- SwitchEval
  - SW-Component Description
- BlinkInputModule
  - SW-Component Description
- BlinkMaster
  - SW-Component Description
- LightActuatorsControl
  - SW-Component Description
- LightSourceSetting
  - SW-Component Description
- LightSourceSetting
AUTOSAR System Design Process

Input: Requirements & Vehicle Info

1a SW Component Description

1c System Description

1b ECU Resource Description

2 Configure System & generate extracts of ECU descriptions

3 Configure each ECU

4 Generate SW executables for each ECU

Iterative corrections and/or optimizations (if required)
AUTOSAR System Design Process

Input: Requirements & Vehicle Info

1a SW Component Description

1c System Description

1b ECU Resource Description

2 Configure System & generate extracts of ECU descriptions

- SW Component Description
  - General characteristics (name, manufacturer, etc.)
  - Communication properties:
    - p_ports
    - r_ports
    - interfaces
  - inner structure (composition)
    - sub-components
    - connections
  - required HW resources:
    - processing time
    - scheduling
    - memory (size, type, etc.)

Iterative corrections and/or optimizations (if required)

Configure each ECU

Generate SW executables for each ECU

SW executables for each ECU
AUTOSAR System Design Process

Input: Requirements & Vehicle Info

1a
SW Component Description

1c
System Description

1b
ECU Resource Description

2
Configure System & generate extracts

3
ECU Resource Description
- General characteristics (name, manufacturer, etc.)
- Temperature (own, environment, cooling/heating)
- Available signal processing methods
- Available programming capabilities
- Available HW: - µC, architecture (e.g. multiprocessor)
- memory
- interfaces (CAN, LIN, MOST, FlexRay)
- periphery (sensor / actuator)
- connectors (i.e. number of pins)

4
SW executables for each ECU

- SW below RTE for micro controller
- Signal path from Pin to ECU-abstraction
AUTOSAR System Design Process

1. Input: Requirements & Vehicle Info
   - 1a: SW Component Description
   - 1b: ECU Resource Description
   - 1c: System Description

2. Configure System
   - Configure System & generate extracts of ECU descriptions
   - Iterative corrections and/or optimizations (if required)

System Description:
- Network topology
  - bus systems: CAN, LIN, FlexRay
  - connected ECUs, Gateways
  - power supply, system activation
- Communication (for each channel)
  - K-matrix
  - gateway table
- Mapping / Clustering of SW components

3. System Description:
   - 3: SW Component
   - 4: SW executables for each ECU
   - SW executables for each ECU
AUTOSAR Metamodel

• 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
## Application Interfaces to Ease Reuse

### Data Type Name: LongAccBase

**Description:**
- Instanitate a redundant sensor interface.
- Range might have to be extended for future applications (passive safety).

<table>
<thead>
<tr>
<th>Data Type Name</th>
<th>Description</th>
<th>Integer Range</th>
<th>Physical Range</th>
<th>Physical Offset</th>
<th>Unit</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>LongAccBase</td>
<td></td>
<td>-32768..+32767</td>
<td>-2,8595..+2,8594</td>
<td>0</td>
<td>rad/sec</td>
<td>This data element can also be used to instantiate a redundant sensor interface. Range might have to be extended for future applications (passive safety).</td>
</tr>
</tbody>
</table>

### Data Type Name: YawRateBase

**Description:**
- Yaw rate measured along vehicle z-axis (i.e. compensated for orientation).
- Coordinate system according to ISO 8855

<table>
<thead>
<tr>
<th>Data Type Name</th>
<th>Description</th>
<th>Integer Range</th>
<th>Physical Range</th>
<th>Physical Offset</th>
<th>Unit</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>YawRateBase</td>
<td></td>
<td>-32768..+32767</td>
<td>-2,8595..+2,8594</td>
<td>0</td>
<td>rad/sec</td>
<td>This data element can also be used to instantiate a redundant sensor interface. Range might have to be extended for future applications (passive safety).</td>
</tr>
</tbody>
</table>

### Data Type Name: RollRateBase

**Description:**

<table>
<thead>
<tr>
<th>Data Type Name</th>
<th>Description</th>
<th>Integer Range</th>
<th>Physical Range</th>
<th>Physical Offset</th>
<th>Unit</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>RollRateBase</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>