Automotive Ethernet and SOA (Service Oriented Architecture)

Vector Live Webinar
Agenda

- **E/E Architecture Trends**
  - AUTOSAR Adaptive
  - AUTOSAR Methodology
  - PREEvision Overview
  - AUTOSAR Adaptive in PREEvision
  - Combined Systems (Classic + Adaptive)
  - Vector AUTOSAR Adaptive Toolchain
E/E Architecture Trends

Technology Trends and Impact on E/E Architecture Evolution

**E/E Architecture Trends**

- **Electrification**
  - Software
  - Complexity
  - Over-the-air Updates

- **Connectivity**
  - Security
  - IT
  - Bandwidth
  - Ethernet

- **Autonomous Driving**
  - Safety

---

**E/E architecture developments impacted by super trends**

**Yesterday**

- **Distributed E/E Architectures**
  - Mix of basic and advanced functionality in a network for ECUs

**Today**

- **Domain E/E Architectures**
  - Domain ECU to handle increasing cross domain functions
  - ECUs for basic functionality getting simpler

**Tomorrow**

- **Centralized E/E Architectures**
  - Powerful, central ECU for advanced functionality
  - Simple ECUs for basic functionality

---

**Increasing functionality**

- Functionality in the cloud
- Complexity in high-end “vehicle brain” ECU
- Basic functions in simple ECUs
- Complexity and cross domain functions concentrated in domain ECU
- ECUs get simpler and standardized
E/E Architecture Trends

Example: OEM Technical Roadmap

PREEvision at Porsche (Update 2018)

Dr. Matthias Görber, Markus Kühl
EEY2
2. Technical Architecture (Mapping of Logical Functions -> Topology)
Agenda

E/E Architecture Trends

- **AUTOSAR Adaptive**
  - AUTOSAR Methodology
  - PREEvision Overview
  - AUTOSAR Adaptive in PREEvision
  - Combined Systems (Classic + Adaptive)
  - Vector AUTOSAR Adaptive Toolchain
As a partner of the Automotive industry AUTOSAR saw the necessity to define a new platform

It was clear that future cars will have heterogenous architectures

The existing architectures had to be complemented by another one

Adaptive AUTOSAR
Agenda

E/E Architecture Trends
AUTOSAR Adaptive

- **AUTOSAR Methodology**
  - PREEvision Overview
  - AUTOSAR Adaptive in PREEvision
  - Combined Systems (Classic + Adaptive)
  - Vector AUTOSAR Adaptive Toolchain
AUTOSAR Methodology

AUTOSAR Adaptive workflow

**Design / Development**

1. Create System Design
2. Define SWCL Design
3. Define Global DEXT Design
4. Define SWCL Internal Services
5. Define Business Logic

**Deployment / Integration / Verification**

1. SWCL Design
2. Adapt platform software to hardware
3. Configure ARA platform services
4. Configure business logic
5. Define test machine
6. Redeploy for test target
7. Provide vehicle package

**Key Steps**

1. **E/E Architect**
   - Map platform SW into E/E architecture
2. **Machine Platform Architect**
   - Define high level UCM, watchdog, diagnostics
3. **SWCL Architect**
   - Platform SW mapping to OEM needs
     - Transformation of service interfaces
     - New apps / functionalities
   - Create diagnostic mapping
4. **Application Developer**
   - Write business logic

**Other Roles**

- **Vehicle Integrator**
- **ECU Integrator**
- **SWCL Integrator**
Agenda

E/E Architecture Trends
AUTOSAR Adaptive
AUTOSAR Methodology

**PREEvision Overview**

AUTOSAR Adaptive in PREEvision
Combined Systems (Classic + Adaptive)
Vector AUTOSAR Adaptive Toolchain
PREEvision is Vector’s model-based tool for automotive E/E Systems Engineering:

ONE tool for the design, management and documentation of complete E/E systems

Support of AUTOSAR (Classic and Adaptive) methodology

Design of safety-relevant systems
PREEvision Overview

Systems Engineering with PREEvision

**Plan**
- Tickets
- Resources
- Projects
- Work packages
- Milestones

**Design**
- Product Lines

**Analyze**
- HARA
- FMEA
- FTA
- Functional Safety Concept

**Test**
- Test Cases
- Test Implementations
- Test Runs

---

Integrated Model-based E/E Systems Engineering with PREEvision

**SysML Modeling with Automotive Profile**
- UML / SysML Modeling

**Automotive Meta Data Model**
- Automotive E/E
  - 4400 Classes
  - 2200 Attribute Types
  - 2500 Relation Types

**Central Data Storage**
- Conflict-free Working
- Variant Management
- Role and Rights Management

**Integrated File Management**
- Life Cycle Management
E/E Systems Engineering Overview

- Domain specific language and data model
- Single source model across all Automotive E/E development Use Cases:
  - The model is the Single Point of Truth
  - Mappings ensure full traceability
  - The model can be analyzed by metrics
- All data objects have a semantic meaning
  - Base for various model checks e.g. for Correctness, Completeness, Consistency
- Automated algorithms for synthetization, scheduling, signal routing, etc.
Agenda

E/E Architecture Trends
AUTOSAR Adaptive
AUTOSAR Methodology
PREEvision Overview

AUTOSAR Adaptive in PREEvision
Combined Systems (Classic + Adaptive)
Vector AUTOSAR Adaptive Toolchain
AUTOSAR Adaptive in PREEvision

Design Workflow for AUTOSAR Adaptive

- Service Design
- Service Interface Design
- Service Interface Binding
- Software Design
- Software Hardware Mapping
- Service Instantiation
- Communication Design

Hardware Design

Network Design

Software

Hardware

HPC

Hypervisor

Executive

Microcontroller

Switch

VC

Core

Switch

VLAN

ECU

Provided Service Instance

Consumed Service Instance

ARXML

SOME/IP Service Instance VP/IP Editor
The AUTOSAR Adaptive Explorer helps at all phases of the workflow with creating the Adaptive AUTOSAR System.

Combined with diagrams, Signal Router and further data synthesis it’s possible to create an Adaptive System from scratch or edit an existing one.
Service

- Service is an abstract description for a set of software functionality with the purpose to be available and be reused by several different clients.
- The Service Definition allows describing capabilities of the services as well as their dependencies.
- Additionally the Service ID and the preferred VLAN for a Service can be defined. These settings are considered by the following phases of the top-down design.
- The Service Definition can also be realized using the SOA Architecture Diagram.
The Service Interface describes the capabilities of a Service.

- It allows the aggregation of Properties, Methods and Events.
- Service Interfaces can be created and modified using the Service Interface Editor.
- Additionally, Service Interfaces can be created and edited graphically on a Class Diagram.
The sub-elements of a Service Interface are characterized by a data type.
- Data types can be assigned using the Service Interface Data Type Editor.
- Designing complex data types can also be realized graphically using the Class Diagram.
PREEvision supports the SOME/IP protocol for the transport of service data over the network.

SOME/IP identifiers shall be defined for every sub-element transmitting data into the bus.

A synthesis for the SOME/IP interface deployment is available.

The definition of the SOME/IP identifiers can be realized using the SOME/IP Interface Deployment Editor.
The serialization of the Service Interfaces is defined by means of Transformers.

Transformation rules can be defined for each sub-element of the Service Interface.

This can be done within the Service Interface Transformation Properties Editor.
AUTOSAR Adaptive in PREEvision

Service Interface (5/5)

- E2EProfileConfiguration is defined by End To End Transformers, End2EndEventProtection-Props by End To End Transformation Properties using dedicated editors.
- End To End Protection can be assigned to every Event and Property Notifier of a Service Interface using the End To End Transformation Properties Assignment.
The Software Architecture in AUTOSAR Adaptive is described by means of adaptive software components which can be grouped in compositions.

Software Architectures can be created using the known software diagrams: SW Component Type or Instance Diagram and Interface Assignment Diagram.

Alternatively, is also possible to work with the Provided Port Connection Editor and Required Port Connection Editor available in the AUTOSAR Adaptive Explorer.
The AUTOSAR Adaptive Application describes a collection of executables that form an adaptive application. It realizes the transition for the support of Software Cluster.

Adaptive applications can contain several executables.

The creation and edition of AUTOSAR Adaptive Applications can be realized with the Adaptive Application Editor.
For deploying an Adaptive Application, startup dependencies to other applications or to states of the machine or function groups need to be defined.

It is also necessary to configure the startup parameters of the process for the operating system.

This activity is supported by the Process Startup Dependency Editor.

For designing the state machine of an application, the State Chart can be used.
The design of complex ECUs and high performance computers (HPCs) is supported in diagrams.

Hypervisor, Microprocessor, Microcontroller, Core and Execution Context are available.

The combination of different operating systems (Linux, MICROSAR, Android) and platforms (AR-Classic, AR-Adaptive) is possible.
AUTOSAR Adaptive Platform supports Ethernet as communication bus technology.
The design of the network topology can be realized using the Network Topology Diagram.
The topology can be viewed and edited with the Network Topology Editor.
As in the classic platform, the software components are distributed to the available machines.

That can be realized graphically, working with diagrams and drag&drop, or using the Software To Machine Mapping Editor.
Service Instances (1/2)

- A Service Instance makes the functionality of a Service Interface available on the AUTOSAR adaptive platform.
- Service Instances are automatically synthesized based on the information of the SW-HW mapping and the Service Interfaces assigned to the SW Ports.
- With the SOME/IP Service Instance Editor the service instances can be edited.
Communication Design (Service Instances)

- Service Instances contain the information needed for the communication into the Ethernet bus.
- Tp/IP configuration as IP addresses and ports and the Service Discovery configuration can be realized using dedicated table editors.
A further aspect to be configured for the machine deployment are the Service Discovery parameters.

IP addresses as well as ports can be configured using the Service Discovery Configuration Editor.
Support of AUTOSAR Adaptive 19-03

- Support of AUTOSAR 19-03 for import and export

<table>
<thead>
<tr>
<th>Format</th>
<th>Use Case</th>
<th>AUTOSAR version</th>
</tr>
</thead>
<tbody>
<tr>
<td>Service Interface Description</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Execution Manifest*</td>
<td>Import/Export</td>
<td>19-03</td>
</tr>
<tr>
<td>Machine Manifest*</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Service Instance Manifest*</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Application Description</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Custom</td>
<td>Export</td>
<td></td>
</tr>
</tbody>
</table>

* Preliminary manifests
Agenda

E/E Architecture Trends
AUTOSAR Adaptive
AUTOSAR Methodology
PREEvision Overview
AUTOSAR Adaptive in PREEvision

▶ Combined Systems (Classic + Adaptive)
Vector AUTOSAR Adaptive Toolchain
Modern Vehicle Architectures

- Functionalities are grouped in different domains or zones.
- Each domain or zone has a main controller – also called Integration ECU.
- Integration ECUs are connected to the computing layer via Ethernet.
- The computing layer consists of several High Performance computers including the connectivity unit.
- The connectivity communicates with the external world, i.e. the Backend Servers and Mobile Devices.
- Such architectures can be modeled with a combination of AUTOSAR Classic ECUs and AUTOSAR Adaptive Machines.
Combined Systems (Classic + Adaptive)

Workflow for combined systems

Service Design
Service Interface Design
Service Interface Binding
Adaptive Software Design
Network Design
Classic Software Design
Software Connection
Adaptive Communication Design
Software Instantiation
Signal Mapping and Serialization
Classic Communication Design
Mapping
Connection Task
Common Task
AR Classic specific
AR Adaptive specific
Application Description
Manifest
ECU Extract
Service Interfaces of AUTOSAR Adaptive can be linked to AUTOSAR Classic Interfaces (Sender Receiver Interfaces, Client Server Interfaces and Trigger Interfaces)
Combined Systems (Classic + Adaptive)

Classic–Adaptive interface linking
Connecting Adaptive and Classic SWCs (1/2)

- The Port Adapter
  - allows connecting Adaptive SW Components with Classic SW Components
  - is the model representation of the links between the sub-elements of one Service Interface and several Classic Interfaces
Several clients and servers can be connected
On the Software Layer, Adaptive and Classic software components can be connected with Port Adapters.

The software components are mapped to machines and ECUs.

The **Signal Router creates a SOME/IP representation** of the Service Interface on the bus (Ethernet).

The SOME/IP representation consists of **PDUs with serialized Signals** containing the data of methods or events.

For **Classic** ECUs, this information is taken into the ECU Extract.

For **Adaptive** Machines, this information is used to check the consistency of the model and will be ignored on export.
Agenda

E/E Architecture Trends
AUTOSAR Adaptive
AUTOSAR Methodology
PREEvision Overview
AUTOSAR Adaptive in PREEvision
Combined Systems (Classic + Adaptive)

- Vector AUTOSAR Adaptive Toolchain
AUTOSAR Adaptive Toolchain

Adaptive System Design
- PREEvision

Diagnostics Design
- CANdela Studio

System Design
- System Architecture
- Sys Diag Design
- Service Instance

Application Design
- App Design
- App Diag Design
- App Manifest

Deployment Configuration
- Platform Service Deployment
- Machine Manifest

Simulation/Test
- CANoe

Application Code
- DaVinci Adaptive Tool Suite

SW Platform
- AMSR Generators

Manifest
- Executable SW

* Preliminary manifests
For more information about Vector and our products please visit

www.vector.com

Author:
Marcelino Varas / Ji-Hwan Chang / Euiyeul Kim
Vector Germany