Processes and tools for the development of modular embedded automotive software
Agenda

- Company overview
- Automotive HMI development
- “Under the hood” software development
- Conclusions/Q&A
EB Automotive Software

Engineering competence close to our customers

USA: Novi (Detroit), Bothell (Seattle)

France: Paris

Austria: Vienna

China: Beijing, Shanghai

Germany: Erlangen, Munich, Gaimersheim (Ingolstadt), Böblingen (Stuttgart), Braunschweig (Wolfsburg), Konstanz

Japan: Tokyo
Success Story
OSEK and AUTOSAR compliant standard core for BMW

The Company
Headquartered in Munich, Germany, the BMW AG is one of the major manufacturers of performance and luxury cars. It owns and produces the MINI brand, and is the parent company of Rolls-Royce Motor Cars. BMW produces more than one million cars per year.

The Challenge
BMW started using standard software in series production as early as 2001. The Standard Core includes up to 56 basic software modules. In addition, BMW substituted selected standard modules with ones developed on their own. BMW searched for a company to take over the integration work.

The Solution
EB has become the main supplier of the BMW Standard Core in 1997. EB has developed most of basic software and has integrated the complete Standard Core versions SC3, SC4, SC5, SC6 and BAC2.1 on a great variety of hardware platforms.

The Benefit
EB’s expertise in OSEK and AUTOSAR software development has helped make efficient project specific integrations and portings of the BMW Standard Core.
Success Story

Modular infotainment system at AUDI

The Company
Headquartered in Ingolstadt, Germany, AUDI AG is one of the major manufacturers of luxury cars and is a 99%-owned subsidiary (as of 2004) of Volkswagen, the largest car manufacturer in Europe. Audi produces more than 960,000 cars per year with several renowned models in its line-up of A-series cars, which include the A4, A6, and A8.

The Challenge
Beginning of 2005 Audi started off their latest generation of the Multi-Media technology (MMI 3G) and sought a development partner who not only had a deep understanding of Automotive user interfaces but who would also manage the powerful project with many different country, feature and display variants and over 1400 different user dialogs.

The Solution
In successful co-operation, Audi Electronics Venture GmbH (AEV) and EB have developed jointly the truly modular infotainment system for the Audi A5, A6 and A8 as well as the Audi Q5, which was introduced in fall 2008 to the public.

The Benefit
Audi benefited from the customized system which at the same time features an extremely high reusability through modularity and usage of open standards, e.g. OSGi. Additionally, EB’s integrated HMI development tool EB GUIDE with automatic code generation helped to manage complexity.
Trends in Automotive Embedded Software Development

• Dealing with increased complexity
• Dealing with increased legislation
• How do we get better products, faster, at a lower cost?
EB Automotive Software

Configuration and Development Tools:
- ECU Configuration and Validation (EB tresos®)
- ADAS and Navi Configuration (EB Assist ADTF)
- Infotainment Configuration

Human Machine Interface:
- Application Module 1
- Application Module 2
- Application Module 3
- Driver Assistance
- Navigation (EB street director)
- Infotainment Suite

Function Software:
- AUTOSAR, OSEK (EB tresos AutoCore/OsekCore)
- Driver Assistance
- Navigation (EB street director)
- Infotainment Suite

Middleware, Real-time Operating Systems:
- FlexRay, CAN, LIN Config and Protocols (EB tresos®)
- Linux
- WinCE
- QNX

Networking:
- MOST, TCP/IP

Hardware:
- HW1
- HW2
- HWn
Solutions for the automotive world

**ECU Software**
- EB provides software modules for BMW standard core
- Leading AUTOSAR technical partner for JasPar in Japan
- Supplier of FlexRay solutions for BMW X5

**Infotainment Software**
- Audi A6 and Q7 Generic User Interface developed by EB
- Navigation and speech dialog software for the new Audi A1

**Driver Assistance**
- ADTF distribution partner for AEV (Audi Electronics Venture)
- Advance Driver Assistance development competenc

Software Engineering, Product Customization, System Integration, Consulting
Infotainment Software

EB GUIDE – One product line for HMI development

- **EB GUIDE Studio**
  Tool for graphical HMI development

- **EB GUIDE Speech Extension**
  Tool add-on for speech dialog design

- **EB GUIDE Speech Target Framework (STF)**
  Runtime for speech dialog, connecting to Automatic Speech recognition and Text-to-Speech engine

- **EB GUIDE Graphics Target Framework (GTF)**
  Runtime for graphical user interface

- **EB GUIDE Studio Plugins**
  Functional extensions of EB GUIDE standard tooling
Infotainment Software

EB street director – Versatile navigation software

- **EB street director Application**
  Complete navigation solution

- **EB street director Core SDK**
  Navigation development environment

- **EB street director Assistant**
  Connected content manager

- **EB street director ADAS Provider**
  Map based electronic horizon

- **Driver Assistance Applications**
  Extensions for EB street director
e.g. curve warning

- **Mobile Applications for EB street director**
e.g. picture viewer, MP3 and video player
Infotainment Software

- Support through all project phases up to overall project lead
- Product customization to customer requirements
- High reusability: EB uses its own products for engineering service projects; saves time and costs
- Success story: Audi MMI 3G Infotainment system
- Operating systems and technologies: Microsoft Auto, Windows Embedded, Linux, QNX, Bluetooth, MOST
Agenda

- **Process overview**
- Define AUTOSAR software application architecture, design and implementation using Rational Rhapsody
- Configure and generate AUTOSAR ECU BSW with EB tresos Studio/AutoCore
Industry solution: Automotive Open System Architecture (AUTOSAR)

An open and standardized automotive software architecture, jointly developed by automobile manufacturers, suppliers, and tool developers

**Goals**

- **Standardization of basic system functions**
  as an OEM wide “standard core” solution

- **Scalability**
  to different vehicle and platform variants

- **Transferability**
  of functions throughout the network

- **Integration**
  of functional modules from **multiple suppliers**

- Higher penetration of **commercial-off-the-shelf software and hardware** components across product lines

**Industry impact**

- Software will be largely independent of the microcontroller chosen

- Development processes will be simplified
  - Reduced development time and costs

- Reuse of software increases at OEM as well as at suppliers
  - Enhanced quality and efficiency

**Challenge:** Conventional proprietary interface

Expensive and repetitive integration effort for common non-differentiating functionalities

**Solution:** AUTOSAR standardized interface

Modular, scalable, transferable, and re-usable throughout the industry
Providing tools based on open standards

- Connect ECU RTE and BSW configurations to system requirements and application models
  - Avoiding costly manual linkages for demonstrating traceability

- Leverage integrations to change, software configuration, asset, and quality management
  - Establishing repeatable, automated and documented workflows
  - Improving collaboration amongst teams and increasing efficiency

- Adding IBM Rational Rhapsody visualization capabilities for EB tresos simulation
AUTOSAR’s layered architecture requires dedicated tools

IBM Rational software and Elektrobit provide best of breed tools
Agenda

- Process overview
- Define AUTOSAR software application architecture, design and implementation using Rational Rhapsody
- Configure and generate AUTOSAR ECU BSW with EB tresos Studio/AutoCore
Systems Development – Where AUTOSAR & SysML Fits

- SysML Provide Analysis Focus
- AUTOSAR Provide Design & Implementation Focus
AUTOSAR Methodology

**System Design, Software Architecture**
- Define Hardware Topology
- Define SWCs, Runnables, Data
- Mapping of SWCs to ECUs
- Communication Matrix
- Export as AUTOSAR Sys-D
- Generate ECU Extract

**Definition of ECU Application (SWC)**
- Model Application Behaviour
- Define ports and data types
- Create SWC Description
- Export SWC Description
- Generate application code

**Configuration of ECU Basic Software**
- EB tresos Studio
- EB tresos AutoCore
AUTOSAR Methodology

OEM Process

System

ECU

Exchange Files

AUTOSAR SYS-D
.arxml

AUTOSAR SWC-D
.arxml

AUTOSAR BSW-Config
.epc, .xdm

Tooling

System design, Software Architecture

- Define Hardware Topology
- Define SWCs, Runnables, Data
- Mapping of SWCs to ECUs
- Communication Matrix
- Export as AUTOSAR Sys-D
- Generate ECU Extract

Definition of ECU Application (SWC)

- Model Application Behaviour
- Define ports and data types
- Create SWC Description
- Export SWC Description
- Generate application code

Configuration of ECU Basic Software

EB tresos Studio
EB tresos AutoCore

TIER1

BSW

Application

OEM

System

ECU

AUTOSAR Methodology

EB tresos Studio
EB tresos AutoCore

AUTOSAR

System design,
Software Architecture

- Define Hardware Topology
- Define SWCs, Runnables, Data
- Mapping of SWCs to ECUs
- Communication Matrix
- Export as AUTOSAR Sys-D
- Generate ECU Extract

Definition of ECU Application (SWC)

- Model Application Behaviour
- Define ports and data types
- Create SWC Description
- Export SWC Description
- Generate application code

Configuration of ECU Basic Software

EB tresos Studio
EB tresos AutoCore

AUTOSAR

System design,
Software Architecture

- Define Hardware Topology
- Define SWCs, Runnables, Data
- Mapping of SWCs to ECUs
- Communication Matrix
- Export as AUTOSAR Sys-D
- Generate ECU Extract

Definition of ECU Application (SWC)

- Model Application Behaviour
- Define ports and data types
- Create SWC Description
- Export SWC Description
- Generate application code

Configuration of ECU Basic Software

EB tresos Studio
EB tresos AutoCore

AUTOSAR

System design,
Software Architecture

- Define Hardware Topology
- Define SWCs, Runnables, Data
- Mapping of SWCs to ECUs
- Communication Matrix
- Export as AUTOSAR Sys-D
- Generate ECU Extract

Definition of ECU Application (SWC)

- Model Application Behaviour
- Define ports and data types
- Create SWC Description
- Export SWC Description
- Generate application code

Configuration of ECU Basic Software

EB tresos Studio
EB tresos AutoCore

AUTOSAR

System design,
Software Architecture

- Define Hardware Topology
- Define SWCs, Runnables, Data
- Mapping of SWCs to ECUs
- Communication Matrix
- Export as AUTOSAR Sys-D
- Generate ECU Extract

Definition of ECU Application (SWC)

- Model Application Behaviour
- Define ports and data types
- Create SWC Description
- Export SWC Description
- Generate application code

Configuration of ECU Basic Software

EB tresos Studio
EB tresos AutoCore

AUTOSAR

System design,
Software Architecture

- Define Hardware Topology
- Define SWCs, Runnables, Data
- Mapping of SWCs to ECUs
- Communication Matrix
- Export as AUTOSAR Sys-D
- Generate ECU Extract

Definition of ECU Application (SWC)

- Model Application Behaviour
- Define ports and data types
- Create SWC Description
- Export SWC Description
- Generate application code

Configuration of ECU Basic Software

EB tresos Studio
EB tresos AutoCore

AUTOSAR

System design,
Software Architecture

- Define Hardware Topology
- Define SWCs, Runnables, Data
- Mapping of SWCs to ECUs
- Communication Matrix
- Export as AUTOSAR Sys-D
- Generate ECU Extract

Definition of ECU Application (SWC)

- Model Application Behaviour
- Define ports and data types
- Create SWC Description
- Export SWC Description
- Generate application code

Configuration of ECU Basic Software

EB tresos Studio
EB tresos AutoCore

AUTOSAR

System design,
Software Architecture

- Define Hardware Topology
- Define SWCs, Runnables, Data
- Mapping of SWCs to ECUs
- Communication Matrix
- Export as AUTOSAR Sys-D
- Generate ECU Extract

Definition of ECU Application (SWC)

- Model Application Behaviour
- Define ports and data types
- Create SWC Description
- Export SWC Description
- Generate application code

Configuration of ECU Basic Software

EB tresos Studio
EB tresos AutoCore

AUTOSAR
AUTOSAR Methodology

System design, Software Architecture

Definition of ECU Application (SWC)
- Model Application Behaviour
- Define ports and data types
- Create SWC Description
- Export SWC Description
- Generate application code

Configuration of ECU Basic Software

Exchange Files
AUTOSAR SYS-D
.arxml

AUTOSAR SWC-D
.arxml

AUTOSAR BSW-Config
.epc, .xdm

Tooling

System design, Software Architecture
- (Define Hardware Topology)
- Define SWCs, Runnables, Data
- Mapping of SWCs to ECUs
- (Communication Matrix)
- Export as AUTOSAR Sys-D
- Generate ECU Extract

Supplier Process
• (Define Hardware Topology)
• Define SWCs, Runnables, Data
• Mapping of SWCs to ECUs
• (Communication Matrix)
• Export as AUTOSAR Sys-D
• Generate ECU Extract

EB trezos Studio
EB trezos AutoCore

EB trezos Studio
EB trezos AutoCore
AUTOSAR Methodology

System Design, Software Architecture

- Definition of ECU
- Configuration of ECU Basic Software

Exchange Files

- AUTOSAR SYS-D .arxml
- AUTOSAR SWC-D .arxml
- AUTOSAR BSW-Config .epc, .xdm

Rational Coverage

Tooling

System Design, Software Architecture

- (Define Hardware Topology)
- Define SWCs, Runnables, Data
- Mapping of SWCs to ECUs
- (Communication Matrix)
- Export as AUTOSAR Sys-D
- Generate ECU Extract

Definition of ECU Application (SWC)

- Model Application Behaviour
- Define ports and data types
- Create SWC Description
- Export SWC Description
- Generate application code

EB tresos Studio
EB tresos AutoCore

OEM and TIER-1

BSW

Application

EB...
AUTOSAR ECU Architecture
AUTOSAR Modeling Tools

- AUTOSAR System Authoring
  - Defines the architecture and integrates the algorithms
- AUTOSAR Behavioral Modeling Tools (BMT Implementation)
  - Implement the algorithms that plug into the architectures

Rhapsody BMT

C Code BMT

Simulink BMT
Software Component Diagram

- Defines the software architecture of the functional system
Example 1: AUTOSAR SW-C Port Binding Table

<table>
<thead>
<tr>
<th>From Component Prototype</th>
<th>From Port</th>
<th>From Port Interface</th>
<th>To Component Prototype</th>
<th>To Port</th>
<th>To Port Interface</th>
<th>Connector Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>ECU1Application</td>
<td>SensLPFiltered</td>
<td>SensLPFiltered_3</td>
<td>ECU1SW</td>
<td>SensLPFiltered</td>
<td>SensLPFiltered_2</td>
<td>DelegationConnectorPrototype</td>
</tr>
<tr>
<td>ECU1ApplicationSens</td>
<td>SensDigital</td>
<td>SensDigital</td>
<td>ECU1SW</td>
<td>SensLPFiltered</td>
<td>SensLPFiltered_2</td>
<td>DelegationConnectorPrototype</td>
</tr>
<tr>
<td>ECU1SW</td>
<td>SensLPFiltered</td>
<td>SensLPFiltered_2</td>
<td>ECU2SW</td>
<td>SensLPFiltered</td>
<td>SensLPFiltered</td>
<td>AssemblyConnectorPrototype</td>
</tr>
<tr>
<td>ECU2Application</td>
<td>SensLPFiltered</td>
<td>SensLPFiltered_1</td>
<td>ECU2SW</td>
<td>SensLPFiltered</td>
<td>SensLPFiltered_2</td>
<td>DelegationConnectorPrototype</td>
</tr>
<tr>
<td>ECU2Application2Actu</td>
<td>Actu2OnOff</td>
<td>Actu2OnOff_1</td>
<td>ECU2SW</td>
<td>Actu2OnOff</td>
<td>Actu2OnOff_2</td>
<td>DelegationConnectorPrototype</td>
</tr>
<tr>
<td>ECU2Application2Actu</td>
<td>Actu2OnOff</td>
<td>Actu2OnOff</td>
<td>ECU2SW</td>
<td>Actu2OnOff</td>
<td>Actu2OnOff_2</td>
<td>DelegationConnectorPrototype</td>
</tr>
<tr>
<td>ECU2SW</td>
<td>Actu2OnOff</td>
<td>Actu2OnOff_2</td>
<td>ECU3W</td>
<td>Actu2OnOff</td>
<td>Actu2OnOff_4</td>
<td>AssemblyConnectorPrototype</td>
</tr>
<tr>
<td>ECU2SW</td>
<td>SensLPFiltered</td>
<td>SensLPFiltered_2</td>
<td>ECU3W</td>
<td>Actu2OnOff</td>
<td>Actu2OnOff_4</td>
<td>AssemblyConnectorPrototype</td>
</tr>
<tr>
<td>ECU3Application</td>
<td>Actu2OnOff</td>
<td>Actu2OnOff_3</td>
<td>ECU3W</td>
<td>Actu2OnOff</td>
<td>Actu2OnOff_4</td>
<td>DelegationConnectorPrototype</td>
</tr>
<tr>
<td>ECU3Application</td>
<td>Actu4Value</td>
<td>Actu4Value_1</td>
<td>ECU3W</td>
<td>Actu4Value</td>
<td>Actu4Value</td>
<td>DelegationConnectorPrototype</td>
</tr>
<tr>
<td>ECU3SW</td>
<td>Actu2OnOff</td>
<td>Actu2OnOff_4</td>
<td>ECU3SW</td>
<td>Actu4Value</td>
<td>Actu4Value</td>
<td>DelegationConnectorPrototype</td>
</tr>
<tr>
<td>ECU3SW</td>
<td>Actu2OnOff</td>
<td>Actu2OnOff_4</td>
<td>ECU3SW</td>
<td>Actu4Value</td>
<td>Actu4Value</td>
<td>DelegationConnectorPrototype</td>
</tr>
</tbody>
</table>
Example 2: AUTOSAR System Signal to DataElement Mapping Table

<table>
<thead>
<tr>
<th>Signal</th>
<th>DataElement</th>
</tr>
</thead>
<tbody>
<tr>
<td>RainSensorActivationHighFkt</td>
<td>VehicleProject.system_2.WasherWiperSubSystem Test_Type.WasherWiper.RainSensor.ActivationHighFkt</td>
</tr>
<tr>
<td>RainSensorActivationHighFkt</td>
<td>VehicleProject.system_2.WasherWiperSubSystem Test_Type.WasherWiper.RainSensor.ActivationHighFkt</td>
</tr>
<tr>
<td>RainSensorDeviceEventLIN</td>
<td>VehicleProject.system_2.WasherWiperSubSystem Test_Type.WasherWiper.RainSensor.DeviceEventLIN</td>
</tr>
<tr>
<td>RainSensorDeviceEventLIN</td>
<td>VehicleProject.system_2.WasherWiperSubSystem Test_Type.WasherWiper.RainSensor.DeviceEventLIN</td>
</tr>
<tr>
<td>RainSensorSensitivityHigh</td>
<td>VehicleProject.system_2.WasherWiperSubSystem Test_Type.WasherWiper.RainSensor.SensitivityHigh</td>
</tr>
<tr>
<td>RainSensorSensitivityHigh</td>
<td>VehicleProject.system_2.WasherWiperSubSystem Test_Type.WasherWiper.RainSensor.SensitivityHigh</td>
</tr>
<tr>
<td>RainSensorDeviceIndicationLIN</td>
<td>VehicleProject.system_2.WasherWiperSubSystem Test_Type.WasherWiper.RainSensor.DeviceIndicationLIN</td>
</tr>
<tr>
<td>RainSensorDeviceIndicationLIN</td>
<td>VehicleProject.system_2.WasherWiperSubSystem Test_Type.WasherWiper.RainSensor.DeviceIndicationLIN</td>
</tr>
<tr>
<td>ResponseError_StatusLIN</td>
<td>VehicleProject.system_2.WasherWiperSubSystem Test_Type.WasherWiper.RainSensor.ResponseError_StatusLIN</td>
</tr>
<tr>
<td>ResponseError_StatusLIN</td>
<td>VehicleProject.system_2.WasherWiperSubSystem Test_Type.WasherWiper.RainSensor.ResponseError_StatusLIN</td>
</tr>
<tr>
<td>RainSensorActivationLowFkt</td>
<td>VehicleProject.system_2.WasherWiperSubSystem Test_Type.WasherWiper.RainSensor.ActivationLowFkt</td>
</tr>
<tr>
<td>RainSensorActivationLowFkt</td>
<td>VehicleProject.system_2.WasherWiperSubSystem Test_Type.WasherWiper.RainSensor.ActivationLowFkt</td>
</tr>
<tr>
<td>RainSensorDeviceEventLIN</td>
<td>VehicleProject.system_2.WasherWiperSubSystem Test_Type.WasherWiper.RainSensor.DeviceEventLIN</td>
</tr>
<tr>
<td>RainSensorDeviceEventLIN</td>
<td>VehicleProject.system_2.WasherWiperSubSystem Test_Type.WasherWiper.RainSensor.DeviceEventLIN</td>
</tr>
<tr>
<td>RainSensorSensitivityHigh</td>
<td>VehicleProject.system_2.WasherWiperSubSystem Test_Type.WasherWiper.RainSensor.SensitivityHigh</td>
</tr>
<tr>
<td>RainSensorSensitivityHigh</td>
<td>VehicleProject.system_2.WasherWiperSubSystem Test_Type.WasherWiper.RainSensor.SensitivityHigh</td>
</tr>
<tr>
<td>RainSensorDeviceIndicationLIN</td>
<td>VehicleProject.system_2.WasherWiperSubSystem Test_Type.WasherWiper.RainSensor.DeviceIndicationLIN</td>
</tr>
<tr>
<td>RainSensorDeviceIndicationLIN</td>
<td>VehicleProject.system_2.WasherWiperSubSystem Test_Type.WasherWiper.RainSensor.DeviceIndicationLIN</td>
</tr>
<tr>
<td>RainSensorDeviceIndicationLIN</td>
<td>VehicleProject.system_2.WasherWiperSubSystem Test_Type.WasherWiper.RainSensor.DeviceIndicationLIN</td>
</tr>
<tr>
<td>RainSensorDeviceIndicationLIN</td>
<td>VehicleProject.system_2.WasherWiperSubSystem Test_Type.WasherWiper.RainSensor.DeviceIndicationLIN</td>
</tr>
<tr>
<td>RainSensorState</td>
<td>VehicleProject.system_2.WasherWiperSubSystem Test_Type.WasherWiper.RainSensor.State</td>
</tr>
<tr>
<td>RainSensorState</td>
<td>VehicleProject.system_2.WasherWiperSubSystem Test_Type.WasherWiper.RainSensor.State</td>
</tr>
<tr>
<td>RainSensorIntensityStatus</td>
<td>VehicleProject.system_2.WasherWiperSubSystem Test_Type.WasherWiper.RainSensor.IntensityStatus</td>
</tr>
<tr>
<td>RainSensorIntensityStatus</td>
<td>VehicleProject.system_2.WasherWiperSubSystem Test_Type.WasherWiper.RainSensor.IntensityStatus</td>
</tr>
<tr>
<td>RainSensorIntensity_Status</td>
<td>VehicleProject.system_2.WasherWiperSubSystem Test_Type.WasherWiper.RainSensor.Intensity_Status</td>
</tr>
<tr>
<td>RainSensorIntensity_Status</td>
<td>VehicleProject.system_2.WasherWiperSubSystem Test_Type.WasherWiper.RainSensor.Intensity_Status</td>
</tr>
<tr>
<td>RainSensorState</td>
<td>VehicleProject.system_2.WasherWiperSubSystem Test_Type.WasherWiper.RainSensor.State</td>
</tr>
<tr>
<td>RainSensorState</td>
<td>VehicleProject.system_2.WasherWiperSubSystem Test_Type.WasherWiper.RainSensor.State</td>
</tr>
<tr>
<td>RainSensorIntensityStatus</td>
<td>VehicleProject.system_2.WasherWiperSubSystem Test_Type.WasherWiper.RainSensor.IntensityStatus</td>
</tr>
<tr>
<td>RainSensorIntensityStatus</td>
<td>VehicleProject.system_2.WasherWiperSubSystem Test_Type.WasherWiper.RainSensor.IntensityStatus</td>
</tr>
<tr>
<td>RainSensorState</td>
<td>VehicleProject.system_2.WasherWiperSubSystem Test_Type.WasherWiper.RainSensor.State</td>
</tr>
<tr>
<td>RainSensorState</td>
<td>VehicleProject.system_2.WasherWiperSubSystem Test_Type.WasherWiper.RainSensor.State</td>
</tr>
<tr>
<td>RainSensorIntensityStatus</td>
<td>VehicleProject.system_2.WasherWiperSubSystem Test_Type.WasherWiper.RainSensor.IntensityStatus</td>
</tr>
<tr>
<td>RainSensorIntensityStatus</td>
<td>VehicleProject.system_2.WasherWiperSubSystem Test_Type.WasherWiper.RainSensor.IntensityStatus</td>
</tr>
<tr>
<td>RainSensorState</td>
<td>VehicleProject.system_2.WasherWiperSubSystem Test_Type.WasherWiper.RainSensor.State</td>
</tr>
<tr>
<td>RainSensorState</td>
<td>VehicleProject.system_2.WasherWiperSubSystem Test_Type.WasherWiper.RainSensor.State</td>
</tr>
<tr>
<td>RainSensorIntensityStatus</td>
<td>VehicleProject.system_2.WasherWiperSubSystem Test_Type.WasherWiper.RainSensor.IntensityStatus</td>
</tr>
<tr>
<td>RainSensorIntensityStatus</td>
<td>VehicleProject.system_2.WasherWiperSubSystem Test_Type.WasherWiper.RainSensor.IntensityStatus</td>
</tr>
<tr>
<td>RainSensorState</td>
<td>VehicleProject.system_2.WasherWiperSubSystem Test_Type.WasherWiper.RainSensor.State</td>
</tr>
<tr>
<td>RainSensorState</td>
<td>VehicleProject.system_2.WasherWiperSubSystem Test_Type.WasherWiper.RainSensor.State</td>
</tr>
<tr>
<td>RainSensorIntensityStatus</td>
<td>VehicleProject.system_2.WasherWiperSubSystem Test_Type.WasherWiper.RainSensor.IntensityStatus</td>
</tr>
<tr>
<td>RainSensorIntensityStatus</td>
<td>VehicleProject.system_2.WasherWiperSubSystem Test_Type.WasherWiper.RainSensor.IntensityStatus</td>
</tr>
<tr>
<td>RainSensorState</td>
<td>VehicleProject.system_2.WasherWiperSubSystem Test_Type.WasherWiper.RainSensor.State</td>
</tr>
<tr>
<td>RainSensorState</td>
<td>VehicleProject.system_2.WasherWiperSubSystem Test_Type.WasherWiper.RainSensor.State</td>
</tr>
<tr>
<td>RainSensorIntensityStatus</td>
<td>VehicleProject.system_2.WasherWiperSubSystem Test_Type.WasherWiper.RainSensor.IntensityStatus</td>
</tr>
<tr>
<td>RainSensorIntensityStatus</td>
<td>VehicleProject.system_2.WasherWiperSubSystem Test_Type.WasherWiper.RainSensor.IntensityStatus</td>
</tr>
<tr>
<td>RainSensorState</td>
<td>VehicleProject.system_2.WasherWiperSubSystem Test_Type.WasherWiper.RainSensor.State</td>
</tr>
<tr>
<td>RainSensorState</td>
<td>VehicleProject.system_2.WasherWiperSubSystem Test_Type.WasherWiper.RainSensor.State</td>
</tr>
<tr>
<td>RainSensorIntensityStatus</td>
<td>VehicleProject.system_2.WasherWiperSubSystem Test_Type.WasherWiper.RainSensor.IntensityStatus</td>
</tr>
<tr>
<td>RainSensorIntensityStatus</td>
<td>VehicleProject.system_2.WasherWiperSubSystem Test_Type.WasherWiper.RainSensor.IntensityStatus</td>
</tr>
</tbody>
</table>
Example 2: AUTOSAR System Signal to DataElement Mapping Table

<table>
<thead>
<tr>
<th>signal</th>
<th>dataElement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Signal Rain Sensor Activation HMI Rqst</td>
<td>VehicleProject system...</td>
</tr>
<tr>
<td>Signal Rain Sensor Activation HMI Rqst</td>
<td>VehicleProject system...</td>
</tr>
<tr>
<td>Signal Rain Sensor Device Event LIN</td>
<td>VehicleProject system...</td>
</tr>
<tr>
<td>Signal Rain Sensor Sensitivity HMI Rqst</td>
<td>VehicleProject system...</td>
</tr>
<tr>
<td>Signal Rain Sensor Device Indicator LIN</td>
<td>VehicleProject system...</td>
</tr>
<tr>
<td>Signal Rain Sensor Device Indicator LIN</td>
<td>VehicleProject system...</td>
</tr>
<tr>
<td>Signal Response Error Stalls LIN</td>
<td>VehicleProject system...</td>
</tr>
<tr>
<td>Signal Rs Activation Rqst</td>
<td>VehicleProject system...</td>
</tr>
<tr>
<td>Signal Rs Activation Rqst</td>
<td>VehicleProject system...</td>
</tr>
<tr>
<td>Signal Rs Activation Status</td>
<td>VehicleProject system...</td>
</tr>
<tr>
<td>Signal Rs Activation Status</td>
<td>VehicleProject system...</td>
</tr>
<tr>
<td>Signal Rs Ambient Temp</td>
<td>VehicleProject system...</td>
</tr>
<tr>
<td>Signal Rs Ambient Temp</td>
<td>VehicleProject system...</td>
</tr>
<tr>
<td>Signal Rs Failure</td>
<td>VehicleProject system...</td>
</tr>
<tr>
<td>Signal Rs Failure</td>
<td>VehicleProject system...</td>
</tr>
<tr>
<td>Signal Rs Failure_UB</td>
<td>VehicleProject system...</td>
</tr>
<tr>
<td>Signal Rs Rain Intensity Status</td>
<td>VehicleProject system...</td>
</tr>
<tr>
<td>Signal Rs Rain Intensity Status</td>
<td>VehicleProject system...</td>
</tr>
<tr>
<td>Signal Rs Rain Intensity Status</td>
<td>VehicleProject system...</td>
</tr>
<tr>
<td>Signal Rs Rain Intensity Status</td>
<td>VehicleProject system...</td>
</tr>
<tr>
<td>Signal Rs Signal Rqst</td>
<td>VehicleProject system...</td>
</tr>
<tr>
<td>Signal Rs Signal Rqst</td>
<td>VehicleProject system...</td>
</tr>
<tr>
<td>Signal Rs Vehicle Speed</td>
<td>VehicleProject system...</td>
</tr>
<tr>
<td>Signal Rs Vehicle Speed</td>
<td>VehicleProject system...</td>
</tr>
<tr>
<td>Signal Rs Vehicle Speed</td>
<td>VehicleProject system...</td>
</tr>
<tr>
<td>Signal Rs Vehicle Type</td>
<td>VehicleProject system...</td>
</tr>
<tr>
<td>Signal Rs Vehicle Type</td>
<td>VehicleProject system...</td>
</tr>
<tr>
<td>Signal Rs Washer Status</td>
<td>VehicleProject system...</td>
</tr>
<tr>
<td>Signal Rs Washer Status</td>
<td>VehicleProject system...</td>
</tr>
<tr>
<td>Signal Rs Washer Status</td>
<td>VehicleProject system...</td>
</tr>
<tr>
<td>Signal Rs Wiper Screen Type</td>
<td>VehicleProject system...</td>
</tr>
<tr>
<td>Signal Rs Wiper Screen Type</td>
<td>VehicleProject system...</td>
</tr>
<tr>
<td>Signal Rs Wiper Screen Type</td>
<td>VehicleProject system...</td>
</tr>
<tr>
<td>Signal Rs Wiper Park Status</td>
<td>VehicleProject system...</td>
</tr>
<tr>
<td>Signal Rs Wiper Park Status</td>
<td>VehicleProject system...</td>
</tr>
<tr>
<td>Signal Rs Wiper Rqst</td>
<td>VehicleProject system...</td>
</tr>
<tr>
<td>Signal Rs Wiper Rqst</td>
<td>VehicleProject system...</td>
</tr>
</tbody>
</table>
Explicit AUTOSAR Implementation

Function: DigitalDriverPeriodicRoutine in file_5

```c
void DigitalDriverPeriodicRoutine()
{
    unsigned int lightLevel = 0;
    unsigned char enabled = 0;

    /* Read the control */
    enabled = Rte_Read_DigitalDriverPeriodicRoutine_in_control();

    if(enabled){
        lightLevel = Rte_Read_DigitalDriverPeriodicRoutine_in_lightLevel();

        Rte_Write_digitalOut_setLight(lightLevel);
    }

    Rte_Call_serialOut_setLight(lightLevel);
}
```
Rhapsody AUTOSAR BMT Implementation

- Modeling the behavior of the AUTOSAR Software Component using UML/SysML Graphical Models
  - Statecharts
  - Activity Diagrams (Flowcharts)
  - C code
- Generate code for the behavior of an AUTOSAR Software Component
  - Regarding the related AR definitions, targeting the RTE
DEMO Rational Rhapsody:

- Define AUTOSAR software application architecture, design and implementation using Rational Rhapsody
  - Define an AUTOSAR Software Architecture
    - Creating Application SWC
    - Defining SW Composition
  - Define an AUTOSAR Software Design & Implementation
    - Defining SWC Internal-Behavior
    - Defining SWC Implementation
  - Define some aspects of an AUTOSAR System
    - Mapping of SWC to ECU Instance
    - Mapping of DataElementPrototype / ClientServerOperation to SystemSignal
    - Defining the System TopLevelSWComposition
Agenda

- Process overview
- Define AUTOSAR software application architecture, design and implementation using Rational Rhapsody
- Configure and generate AUTOSAR ECU BSW with EB tresos Studio/AutoCore
Configure & generate AUTOSAR ECU BSW
tresos Studio/AutoCore
Configure & generate AUTOSAR ECU BSW

tresos Studio/AutoCore

AUTOSAR compliant ECU Basic SW

The software component template describes these components completely.

Of these software components only the AUTOSAR Interface side can be fully described in the software component template.
Configure & generate AUTOSAR ECU BSW

tresos Studio/AutoCore

Configuration &
generation
of ECU Basic SW

tresos AutoCore

AUTOSAR compliant ECU Basic SW
Configure & generate AUTOSAR ECU BSW

tresos Studio/AutoCore

Configuration & generation of ECU Basic SW

AUTOSAR compliant ECU Basic SW
Workflow: Exchange Files

System Design, Topology

ECU Application Software

APPLICATION

EB tresos Studio
EB tresos AutoCore

Target code

Appl. Code
\texttt{.c, .h}

BSW Code
\texttt{.c, .h}
Thank you!