VOLKSWAGEN AG, Carmeq GmbH

AUTOSAR-Compliant Functional Modeling with MATLAB®, Simulink®, Stateflow® and Real-Time Workshop® Embedded Coder of a Serial Comfort Body Controller

MathWorks Automotive Conference 2007

Andreas Köhler, Tillman Reck
Wolfsburg, Berlin, 20th June 2007
Agenda

- AUTOSAR - a short introduction
- Motivation
- AUTOSAR Body/Comfort ECU
- AUTOSAR tool chain setup
- Conclusion
Current OEM ECU configuration

Control units contain more and more functional software of different subcontractors:

- **Supplier Specific Function:**
  - Covers the current situation

- **OEM Specific Function:**
  - Corporately develop with the supplier:
    - Corporately IP, Software only available within OEM ECUs

- **Exclusive OEM Function:**
  - Individually from OEM or by a OEM ordered SW-supplier developed software

- **Carry-Over-Part-Funktion:**
  - Could be software of each above categories
AUTOSAR defines the ECU software architecture.
AUTOSAR supports the complete function driven development process

Function development

Architecture decision

System development

ECU development

EEXA/2 Fahrzeugarchitekturen
ELEKTRIK / ELEKTRONIK ENTWICKLUNG

Folie 5
20.06.2007
Andreas Köhler (Volkswagen AG), Tillman Reck (Carmeq GmbH), (V1.0.1)
Motivation

• Volkswagen AG and HELLA are jointly developing a fully functional Body/Comfort ECU for a Volkswagen series-production vehicle which is furnished with AUTOSAR compatible software.

• The aim of the project is to check, over a period of twelve months on a day-to-day basis, how the demands of the automotive industry with regard to introduction in serial development can be met.

• The main focus depends on migration scenarios, preparation of serial production and the influence on development process.
Integration into an existing vehicle

- **AUTOSAR Body/Comfort ECU:**
  - ECU, application SW & integration from Hella
  - Application & Volkswagen SSC module from VW
  - AUTOSAR basic software release 1.0 implemented by ELEKTROBIT (formerly 3Soft)
  - Processor & development tool chain from NEC automotive
  - The MathWorks delivers the AUTOSAR Development Kit (ADK)
Possible migration scenarios for basic SW & RTE

- **AUTOSAR basic software (RTE and lower modules)**
  1. Single sided RTE within OEM SSC
  2. OEM SSC with AUTOSAR modules
  3. AUTOSAR BSW with OEM SSC modules

KSG on AUTOSAR project has chosen variant 3!
Migrations scenarios for the application layer

- **Application modules**
  1. Refactoring of existing software (APP1)
  2. Development of new software (APP2)
  3. Integration of legacy code (object code)
  4. Integration of existing application MATLAB® models

---

KSG on AUTOSAR project realized ALL variants!
Agenda

- AUTOSAR - a short introduction
- Motivation
- AUTOSAR Body/Comfort ECU
- AUTOSAR tool chain setup
- Conclusion
• Create architectural model with interfaces (results from the AUTOSAR work package 10.1 were incorporated)
• Create SWC Description and import architecture definition to Simulink®
• Implementation of the SWC behavior with MATLAB® and Simulink®
• Simulation and Test with Simulink
• Generating AUTOSAR code + SWC Description with Real-Time Workshop® Embedded Coder + ADK
• Software and ECU Integration using Elektrobit Tresos
• ECU tests with CANoe and final in-vehicle tests
Embedding of MATLAB® and Simulink® in the AUTOSAR process

Architectural modelling tool

MATLAB Simulink and Stateflow

Modeling, refinement, code generation

Architectural description of SWCs + interfaces

Real-Time Workshop Embedded Coder + ADK Code Generator

Architectural description of SWCs + interfaces. Implementation

Generation of SWC API

ECU/System Integration

SWC.h
rteTypes.h

SW Integration

Architecture redefinition

Export

Import

XML

SWC description

*.c

*.h

*.o

Implementation
AUTOSAR application architecture

- Architecture model of the application
- Composition `CentralLockingMaster` (CLMaster) within Simulink implemented Atomic SWCs
AUTOSAR application in MATLAB and Simulink
Relationship between SWC, Runnable and Simulink model

- **SWC Comfort**
- **Intermediate layer Comfort_Parts specify init Runnables**
- **Runnable Entity (Comfort_Periodic) and its interfaces**
## Specifying Runnable Entities

Implementation of SWC generally by 4: Runnables: Cyclic, Init, StoreBefore, RestoreAfter:

<table>
<thead>
<tr>
<th>Runnables</th>
<th>Event</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rte_Comfort_Cyclic</td>
<td>TimingEvent</td>
<td>Simulink function</td>
</tr>
<tr>
<td>Rte_Comfort_Init</td>
<td>ModeSwitchEvent</td>
<td>First initialization</td>
</tr>
<tr>
<td>Rte_Comfort_StoreBefore</td>
<td>ModeSwitchEvent</td>
<td>Securing application system states</td>
</tr>
<tr>
<td>Rte_Comfort_RestoreAfter</td>
<td>ModeSwitchEvent</td>
<td>Reestablishing system states</td>
</tr>
</tbody>
</table>

**) Configured by Workspace database using Simulink Signal objects and StorageClass attributes to control code generation of structured data. Runnables are in legacy code.

Configure RTE-Events within subsystem *Comfort_Parts* (SWC) by Simulink Subsystem-Dialog (AUTOSAR)
Specifying Data Access Methods of Runnable Entities

Configuration by Port-Dialog (AUTOSAR)
- Runnable Access methods
  - Implicit: cyclic data flow
  - Explicit: non-cyclic data flow (optimizable)
- Name of the AUTOSAR interface
- Name of the AUTOSAR DataElement
Coding and personalization aspects

- Using explicit Data Access (versus Simulink data flow generally can be implemented the best via implicit data flow)
- Coding
  - SWC partially implemented by legacy code
- Personalization + Coding
  - Data distributed via physical busses == AUTOSAR record type
Integration of the AUTOSAR Application by the AUTOSAR Process

*third-party SWCs*

- **MATLAB Simulink Stateflow**
  - SW-Component Description
  - ECU Resource Description (HW only)
  - System-Constraint Description

**AUTOSAR System Configuration Generator**
- Modelling and Complex generation step:
  - Using MATLAB, Simulink, and Stateflow

**System Configuration Description**
- ECU extract of System Configuration
- ECU extract of System Configuration

**Complex generation step:**
- complex algorithm or engineering work

**Compiler/Linker**
- Software Components
- AUTOSAR RTE
- OS
- Basic Software
- MCAL

**Software Componenten**
- Software Components *.c
- Software Components *.h

**API**
- Information / Database (no files)
- Implementations *.c *.h

**Software Components**
- third-party SWCs
Agenda

- AUTOSAR - a short introduction
- Motivation
- AUTOSAR Body/Comfort ECU
- AUTOSAR tool chain setup
- Conclusion
Conclusion 1/2

Application Migration Scenarios

- Integration of Legacy Code ✓
- Refactoring ✓
- Reuse of Applications with AUTOSAR interfaces ✓
- Integration of automatic generated Applications with AUTOSAR interfaces ✓

⇒ Each method is realizable with AUTOSAR.
⇒ Case-by-case decision on method to apply.

Development Process

- Development according to the AUTOSAR Methodology gets another focus: configuration.

⇒ Intelligent tools supporting **efficient configuration** and **optimized code generation** are essential!
Conclusion 2/2

Proprietary OEM solutions for Basic Software during an interim period

- Proprietary solutions require various different integration concepts.
  ➔ Goal is: Cooperate on standards – compete on implementations!

Software Integration – Functional Integration

- Dynamic features of an application are covered by RTE and are thus taken into account at a later stage of integration only.
  ➔ For a broad functional integration of a software component, additional information is required in the software component description.

AUTOSAR is ready to be used now!

- Improvement potential: The project lacked variant management.
  ➔ AUTOSAR phase 2 still leaves enough to do.
  ➔ Basic concepts, specifications and methodology are ready to be used now!
Conclusion

The AUTOSAR control unit is integrated in a Passat Sedan!
Thanks for your attention!