Development of AUTOSAR Software Components with Model-Based Design

Dr. Joachim Schlosser
Application Engineering
The MathWorks GmbH

MathWorks
Automotive Conference ’08
3 things to remember about AUTOSAR, Model-Based Design with Simulink and Real-Time Workshop Embedded Coder

- Ease of use: No structural model changes. Switch between AUTOSAR and other target without modifying the model.
- Easy workflows for new and legacy models, keeping it simple for the user.
- AUTOSAR is seamlessly integrated in Simulink and Real-Time Workshop environment.
Motivation
- Management of E/E* complexity associated with growth in functional scope
- Flexibility for product modification, upgrade and update
- Scalability of solutions within and across product lines
- Improved quality and reliability of E/E systems

Goal
- The objective of the partnership is the establishment of an open standard for automotive E/E architecture

*Electric/Electronic
Agenda

- Introduction to Model-Based Design
- Introduction to Codegeneration
- AUTOSAR
  - Overview
- The MathWorks approach to AUTOSAR
  - Applying Simulink to AUTOSAR: What’s New in R2008a
- Common Workflows
Introduction to Model-Based Design
Traditional development processes prevent errors from being caught early in the program.

System Architecture

Design

Mechanical

Electrical

Control Design

Embedded Software

Integrate

Implement

System Testing

Requirements Specifications
Model-Based Design enforces continuous testing and verification throughout the design process.

- **Requirements Specifications**
  - Generate:
    - Assertions
    - Test results
    - Documentation
  - Perform:
    - Simulation
    - Verification
    - Design Analysis
    - Traceability Analysis

- **Generate**
  - Integration
  - Verification & Validation
  - Rapid Prototyping
    - HW-in-the-Loop
    - SW-in-the-Loop

- **Implement**
  - Hardware
    - MCU
    - DSP
    - FPGA
    - ASIC
  - Software
    - C, C++
    - HDL (VHDL, Verilog)

- **Design**
  - Subsystem A
  - Subsystem B
  - Component A
  - Component B

- **System Architecture**
  - Environment System
  - System Environment
  - Execution Harness

- **Real-Time Software Prototypes**

- **Continuous Verification & Validation**
  - Generate:
    - Assertions
    - Test results
    - Documentation
  - Perform:
    - Simulation
    - Verification
    - Design Analysis
    - Traceability Analysis

- **Rapid Prototyping**
  - HW-in-the-Loop
  - SW-in-the-Loop
Benefits of Model-Based Design

- **Cost**
  - Minimize prototypes and re-work
  - Facilitates design reuse

- **Schedule**
  - Shortens time-to-market
  - Enhances team communication

- **Performance**
  - Fosters innovation
  - Improves quality
Introduction to Code Generation
Core Code Generation Building Blocks

Real-Time Workshop®
- Generates code from Simulink that is easy to interact and experiment with

Real-Time Workshop® Embedded Coder™
- Generates extremely efficient code that can be customized to look and perform like hand code

Embedded IDE Link for <embedded toolchain>
- Communication with compiler IDE

Target Support Package™ <microprocessor>
- Makes it easy to deploy generated code on particular microprocessors including

You can deploy code on any microprocessor using Real-Time Workshop and Real-Time Workshop Embedded Coder because they generate standard C (ANSI/ISO-C).
Multiple Domain Code Generation

- Simulation Acceleration
  - Simulink Accelerator
  - S-Function
  - GRT/ERT/RSIM

- Modeling and Simulation
- Functional Rapid Prototyping and HIL
  - xPC Target
  - Real-Time Windows Target
  - VxWorks Example

- Embedded Deployment
  - On-Target Rapid Prototyping
  - Embedded Code Generation
    - ANSI/ISO-C
    - Embedded Targets
    - Links for IDEs
AUTOSAR Architecture

AUTOSAR Software

Application Software Component

AUTOSAR Interface

Actuator Software Component

AUTOSAR Interface

Sensor Software Component

AUTOSAR Interface

AUTOSAR Runtime Environment (RTE)

Basic Software

ECU-Hardware

AUTOSAR Software

Application Software Component

AUTOSAR Interface

ECU Firmware

Standard Software

API 2
VFB & RTE relevant

API 1
RTE relevant

API 0

API 3 Private Interfaces inside Basic Software possible

The MathWorks
MATLAB & SIMULINK

MATLAB Automotive Conference ’08
Key Features

- **Modularity and configurability**
  - Modular software architecture for automotive ECUs
  - Consideration of HW dependent and HW independent SW modules
  - Integration of SW modules provided by different suppliers
  - Transferability of functional SW-modules within a particular E/E-system
  - Scalability of the E/E-system across the entire range of vehicle product lines

from: www.autosar.org
Applying Simulink to AUTOSAR
New in R2008a

- Simulink to import and export
  - AUTOSAR Software Component (SW-C) Descriptions, in XML
- Software-in-the-Loop support
  - Automatic S-Function configuration and generation for routing simulation data using AUTOSAR RTE API calls
- Real-Time Workshop® Embedded Coder to generate
  - AUTOSAR SW-C Implementations (runnables) compliant with AUTOSAR Run Time Environment, in C code
- Supports:
  - AUTOSAR v2.0 and v2.1
Mapping between AUTOSAR SW-C Meta Model and Simulink

- AUTOSAR Software Components
Mapping between AUTOSAR SW-C Meta Model and Simulink

- Ports and Interfaces
Common Workflows

- Importing AUTOSAR SW Component Descriptions
Common Workflows

- Development of controller behavior
Common Workflows

- Implementing and publishing Software Components by generating AUTOSAR compliant code
Volkswagen use of Production Code Generation for AUTOSAR

3 things to remember about AUTOSAR, Model-Based Design with Simulink and Real-Time Workshop Embedded Coder

- Ease of use: No structural model changes. Switch between AUTOSAR and other target without modifying the model.
- Easy workflows for new and legacy models, keeping it simple for the user.
- AUTOSAR is seamlessly integrated in Simulink and Real-Time Workshop environment.
3 things to remember about AUTOSAR, Model-Based Design with Simulink and Real-Time Workshop Embedded Coder

- **Ease of use:** No structural model changes. Switch between AUTOSAR and other target without modifying the model.
- **Easy workflows for new and legacy models,** keeping it simple for the user.
3 things to remember about AUTOSAR, Model-Based Design with Simulink and Real-Time Workshop Embedded Coder

- Ease of use: No structural model changes. Switch between AUTOSAR and other target without modifying the model.
The one thing to remember about AUTOSAR, Model-Based Design with Simulink and Real-Time Workshop Embedded Coder

- Ease of use: No structural model changes. Switch between AUTOSAR and other target without modifying the model.