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.
Automotive Open System Architecture

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

- **Simulink Stateflow®**
  - Real-Time Workshop® / Stateflow Coder
  - Real-Time Workshop Embedded Coder

- **Modeling and Simulation**
  - Simulation Acceleration
    - Simulink Accelerator
  - Functional Rapid Prototyping and HIL

- **Rapid Prototyping**
  - 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
AUTOSAR Membership

Core Members

Premium Members

Associate Members

+ 50 other companies

MathWorks 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

<table>
<thead>
<tr>
<th>Embedding of MATLAB® and Simulink® in the AUTOSAR process</th>
</tr>
</thead>
<tbody>
<tr>
<td>Architectural modelling tool</td>
</tr>
<tr>
<td>Architecture redefinition</td>
</tr>
<tr>
<td>ECU/System Integration</td>
</tr>
<tr>
<td>Generation of SWC API</td>
</tr>
<tr>
<td>SW Integration</td>
</tr>
<tr>
<td>import</td>
</tr>
<tr>
<td>Architectural description of SWCs + interfaces</td>
</tr>
<tr>
<td>export</td>
</tr>
</tbody>
</table>

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.