MATLAB EXPO 2017
From Simulink to AUTOSAR: Enabling AUTOSAR Code Generation with Model-Based Design
Durvesh Kulkarni
Introduction to AUTOSAR
- Simulink approach to AUTOSAR
- Overview of Modeling SWCs & Modeling Styles

AUTOSAR Design Workflows
- Bottom Up, Top Down & Round Trip

Advanced Topics – Top 5
- Startup, Reset, and Shutdown Modeling
- Basic Software (BSW) Access
- J-MAAB Type B Architecture
- Mode Management (ModeSenderPorts, ModeSwitchPoints, …)
- Variability inside a Software Component

Getting Started Resources

MATLAB EXPO 2017
What is AUTOSAR?

AUTOSAR® (AUTomotive Open System ARchitecture) is an open and standardized automotive software architecture.
Complexity of Classic AUTOSAR is growing

Components specification
Body and Interior Electronics APIs...
APIs for Powertrain, Chassis...
DCM, DEM Revisions...
Added safety concepts...
RTE Enhancements...
Variant Handling for Application Interfaces...
Efficient NV Data Handling via RTE...
Crypto Interface, V2X support...

AUTOSAR Releases

2.0          2.1            3.0              3.1                 3.2                        4.0                     4.1
90 files      120 files       125 files        129 files                  136 files                       183 files        198 files
4.2           4.3                         198 files        227 files                            238 files
AUTOSAR Adoption
AUTOSAR Members

9 Core Partners
- BMW Group
- BOSCH
- DAIMLER
- PSA PEUGEOT CITROEN
- TOYOTA
- VOLKSWAGEN AG
- Continental
- OPEL
- Ford

11 Development Members
- itemis
- SYMTA VISION
- SGS
- C & S
- Kenera
- TUV NORD
- OFFIS
- VALIDAS

57 Premium Members
- HONDA
- KIA
- HYUNDIA-KIA MOTORS
- Mazda
- PORSCHE
- RENAULT
- Denso
- HITACHI
- Johnson Controls
- MBtech
- MAN
- TRW
- Valeo
- Magna
- ZF
- Lear
- Continental
- RENESAS
- Infineon
- Freescale
- ARM
- NEC
- Fujitsu

11 Tier 1
- Autoliv
- Ak motif
- AEC
- TRW
- Magna
- ZF
- Lear
- Continental
- RENESAS
- Infineon

57 Tier 2
- ALTRAN
- Infineon
- Freescale
- ARM
- NEC
- Fujitsu
- Panasonic
- Hitachi
- Johnson Controls
- MBtech
- MAN
- TRW
- Valeo
- Magna
- ZF
- Lear
- Continental
- RENESAS
- Infineon

30 Tier 3
- ALTRAN
- Infineon
- Freescale
- ARM
- NEC
- Fujitsu
- Panasonic
- Hitachi
- Johnson Controls
- MBtech
- MAN
- TRW
- Valeo
- Magna
- ZF
- Lear
- Continental
- RENESAS
- Infineon

11 Tier 4
- ALTRAN
- Infineon
- Freescale
- ARM
- NEC
- Fujitsu
- Panasonic
- Hitachi
- Johnson Controls
- MBtech
- MAN
- TRW
- Valeo
- Magna
- ZF
- Lear
- Continental
- RENESAS
- Infineon

11 Tier 5
- ALTRAN
- Infineon
- Freescale
- ARM
- NEC
- Fujitsu
- Panasonic
- Hitachi
- Johnson Controls
- MBtech
- MAN
- TRW
- Valeo
- Magna
- ZF
- Lear
- Continental
- RENESAS
- Infineon

88 Associate Members
17 Attendees

www.autosar.org
AUTOSAR Support from Embedded Coder and Simulink

- **Application Layer**
  - Services Layer
  - ECU Abstraction Layer
  - Microcontroller Abstraction Layer
  - Basic Software

- **Run Time Environment (RTE)**
  - ECU Hardware

- **Software Architecture Definition**

- **Behavior Modeling & Code Generation**

- **Modeling and Simulation**

- **BSW Configuration & RTE Generation**

- **Basic SW Providers**

- **Authorsing Tools**
  - ETAS
  - Vector

MATLAB EXPO 2017
Agenda

- Introduction to AUTOSAR
  - Simulink approach to AUTOSAR
  - Overview of Modeling SWCs & Modeling Styles
- AUTOSAR Design Workflows
  - Bottom Up, Top Down & Round Trip
- Advanced Topics – Top 5
  - Startup, Reset, and Shutdown Modeling
  - Basic Software (BSW) Access
  - J-MAAB Type B Architecture
  - Mode Management (ModeSenderPorts, ModeSwitchPoints, …)
  - Variability inside a Software Component
- Getting Started Resources
Simulink Approach to AUTOSAR

Simulink and Embedded Coder
+ AUTOSAR Support package for Embedded Coder

Available via web download

No separate AUTOSAR Blockset needed

Import

Export

C Code and ARXML

Code-generation through Mapping

MATLAB EXPO 2017
AUTOSAR Schema Versions

Seamless support for AUTOSAR Releases

- Import detects AUTOSAR 2.x – 4.x release from ARXML file
- User selects AUTOSAR release from configuration set options for code generation and ARXML export

<table>
<thead>
<tr>
<th>MATLAB Release</th>
<th>AUTOSAR Release</th>
</tr>
</thead>
<tbody>
<tr>
<td>R2015b, R2016a/b, R2017a</td>
<td>2.1, 3.0, 3.1, <strong>3.2</strong> (Rev 3.2.2), 4.0, 4.1, <strong>4.2</strong> (Rev 4.2.1, 4.2.2)</td>
</tr>
<tr>
<td>R2014b, R2015a</td>
<td>2.1, 3.0, 3.1, 3.2, 4.0, <strong>4.1</strong> (Rev 4.1.1)</td>
</tr>
<tr>
<td>R2012a/b, R2013a/b, R2014a</td>
<td>2.1, 3.0, 3.1, 3.2, <strong>4.0</strong> (Rev 4.0.2)</td>
</tr>
<tr>
<td>R2011b</td>
<td>2.0, 2.1, 3.0, 3.1, 3.2</td>
</tr>
<tr>
<td>R2010a/b, R2011a</td>
<td>2.0, 2.1, 3.0, 3.1</td>
</tr>
<tr>
<td>R2009a/b</td>
<td>2.0, 2.1, 3.0</td>
</tr>
<tr>
<td>R2008a/b</td>
<td>2.0, 2.1</td>
</tr>
</tbody>
</table>
Introduction to AUTOSAR
  - Simulink approach to AUTOSAR
  - Overview of Modeling SWCs & Modeling Styles

AUTOSAR Design Workflows
  - Bottom Up, Top Down & Round Trip

Advanced Topics – Top 5
  - Startup, Reset, and Shutdown Modeling
  - Basic Software (BSW) Access
  - J-MAAB Type B Architecture
  - Mode Management (ModeSenderPorts, ModeSwitchPoints, …)
  - Variability inside a Software Component

Getting Started Resources
Model AUTOSAR Components

Application Layer

AUTOSAR Software Component 1

* * *

AUTOSAR Software Component n

Virtual Functional Bus (VFB)

Periodic Rate-Based

Periodic & Asynchronous

Multi-Rate & Asynchronous

MATLAB EXPO 2017
Modeling AUTOSAR Communication

- Ports in a AUTOSAR software component allow for communication

- Categories of ports based on direction
  - Require port
  - Provide port

- Each port can have either of the following Interfaces
Supported Events for a Runnable

Each Runnable should have at least one event attached
Agenda

- Introduction to AUTOSAR
  - Simulink approach to AUTOSAR
  - Overview of Modeling SWCs & Modeling Styles

- AUTOSAR Design Workflows
  - Bottom Up, Top Down & Round Trip

- Advanced Topics – Top 5
  - Startup, Reset, and Shutdown Modeling
  - Basic Software (BSW) Access
  - J-MAAB Type B Architecture
  - Mode Management (ModeSenderPorts, ModeSwitchPoints, …)
  - Variability inside a Software Component

- Getting Started Resources
Bottom-Up Workflow (Starting from Simulink)

AUTOSAR Authoring Tool

Import SWC Description

Export SWC Description/Generate SWC C code

AUTOSAR design (Meta-model)
Using MATLAB for automating common tasks

%% Setup AUTOSAR Configuration programmatically

model = 'Average_VehicleSpeed_Calculation';

% Modify AUTOSAR Properties
autosarProps = autosar.api.getAUTOSARProperties(model);
set(autosarProps, 'Input', 'IsService', true);
set(autosarProps, 'XmlOptions', 'ArxmlFilePackaging','SingleFile');

% Modify Simulink Mapping to AUTOSAR
slMap = autosar.api.getSimulinkMapping(model);
mapInport(slMap, 'Input', 'Input', 'Input', 'ExplicitReceive');
mapOutport(slMap, 'Output', 'Output', 'Output', 'ExplicitSend');
Top-Down Workflow (Starting from SWC Description)

AUTOSAR Authoring Tool

Export SWC Description

Import SWC Description

Model Based Design

Export SWC Description/Generate SWC C code

Merge SWC Description
Top Down Workflow

AUTOSAR Authoring Tool

ARXML Files

Import as new Simulink model

OR

Update existing Simulink model

Top Down Workflow

Starts with Authoring Tool, then user exports ARXML files from Authoring tool.

User can then either import the ARXML files into a new Simulink Skeleton model or Update an existing Simulink Model.
Updating Existing Models from ARXML

V1.arxml Updated to V2.arxml
Update Existing Models from ARXML

```matlab
%cleanup
bdclose('all');
clear;

open_system('ASWC'); % Model needs to be open in order to perform update Model Command

%Import ARXML Files
importerObj = arxml.importer('rtwdeo_autosar_multirunnables_v2.arxml')

%Update existing model
importerObj.updateModel('ASWC')
```
Round-Trip Workflow

AUTOSAR Authoring Tool

Export SWC Description

Merge SWC Description

Export SWC Description

Generate SWC C code
ARXML Import using Vector DaVinci

AUTOSAR Authoring Tool

Export SWC Description

Merge SWC Description

Export SWC Description/Generate SWC C code

MATLAB EXPO 2017
Verification with Software- and Processor-In-The-Loop (PIL)

- Support for SIL/PIL with AUTOSAR target
- Profile code and measure execution time on target
- Develop a custom PIL target for AUTOSAR using the toolchain build approach
MISRA C:2012 for AUTOSAR target

100% Compliance with MISRA C:2012 Mandatory and Required rules
Agenda

- **Introduction to AUTOSAR**
  - Simulink approach to AUTOSAR
  - Overview of Modeling SWCs & Modeling Styles

- **AUTOSAR Design Workflows**
  - Bottom Up, Top Down & Round Trip

- **Advanced Topics – Top 5**
  - Startup, Reset, and Shutdown Modeling
  - Basic Software (BSW) Access
  - J-MAAB Type B Architecture
  - Mode Management (ModeSenderPorts, ModeSwitchPoints, …)
  - Variability inside a Software Component

- **Getting Started Resources**
Startup, Reset, and Shutdown Modeling
AUTOSAR Startup, Reset, and Shutdown Modeling

New Simulink blocks for Initialize Function and Terminate Function

- You can map each Simulink initialize, reset, or terminate entry-point function to an AUTOSAR runnable
- All modeling styles are supported
  - Flexibility to use either Rate-Based or Export function modeling style
- Less wiring is required
- Can perform SIL

rtwdemo_autosar_swc
Basic Software (BSW) Access
AUTOSAR Basic Software (BSW) block library

Simulate BSW including Diagnostic Event Manager (DEM) and NVRAM Manager (NvM)

- Out of the box solution for calls to AUTOSAR BSW services
  - Drag and drop DEM/NvM blocks for Basic Software simulation
  - Everything is preconfigured

» rtwdemo_autosar_nvm_emulation
Power Up Power Down AUTOSAR NvM Emulation

- Initialize, Reset & Terminate Blocks can be effectively used to model Start Up and Shut Down functionalities.

- System Level Modelling of AUTOSAR Components & Services Basic Software blocks can be used.
J-MAAB Type B Support
This model shows the implementation leveraging periodic and asynchronous rates (sample times).

Asynchronous function-call runnable at the top level of the model interacts with a periodic rate-based runnable.

Model type B (β) — Places function layers above scheduling layers.

rtwdemo_autosar_swc_fcnCalls
Mode Management (ModeSenderPorts, ModeSwitchPoints, …)
AUTOSAR ModeSenderPorts and ModeSwitchPoints

Modeling of AUTOSAR Mode-Switch (M-S) communication

- Ability to model application mode manager components, including AUTOSAR mode sender ports.
- Mode sender ports output a mode switch to connected mode user components.
Variability inside a Software Component
Variants in AUTOSAR component modeling

Create variants for Ports and Runnables

- Import Variation Points on Ports and Runnables into Simulink
- Model using Variant Source and Variant Sink blocks
- Validate variant conditions on blocks match designed behavior from imported ARXML files
Agenda

- Introduction to AUTOSAR
  - Simulink approach to AUTOSAR
  - Overview of Modeling SWCs & Modeling Styles

- AUTOSAR Design Workflows
  - Bottom Up, Top Down & Round Trip

- Advanced Topics – Top 5
  - Startup, Reset, and Shutdown Modeling
  - Basic Software (BSW) Access
  - J-MAAB Type B Architecture
  - Mode Management (ModeSenderPorts, ModeSwitchPoints, …)
  - Variability inside a Software Component

- Getting Started Resources
AUTOSAR Support from Embedded Coder

AUTOSAR (AUTOMotive Open System ARchitecture) is an open and standardized automotive software architecture jointly developed by automobile manufacturers, suppliers, and tool developers.

Embedded Coder® Support Package for AUTOSAR Standard lets engineers model and simulate AUTOSAR software components, generate AUTOSAR production code, and verify AUTOSAR generated code using software- and processor-in-the-loop simulations. The support package also enables import and export of AUTOSAR Software Component descriptions that support top-down, bottom-up, and round-trip workflows involving third-party AUTOSAR authoring tools such as DaVinci Developer.

Platform and Release Support

See the hardware support package system requirements table for current and prior version, release, and platform availability.

View new features in the release notes.

http://www.mathworks.com/hardware-support/autosar.html
Embedded Coder Support Package for AUTOSAR Standard

version 1.9 (15.1 KB) by MathWorks Embedded Coder Team

Develop AUTOSAR software components for automotive systems.

Embedded Coder® Support Package for AUTOSAR Standard provides additional support to Embedded Coder that includes modeling AUTOSAR elements and generating arxml and AUTOSAR-compatible C code from a Simulink® model. Verify AUTOSAR generated code using software- and processor-in-the-loop simulations.

This support package is functional for R2014b and beyond.
Code Generation for AUTOSAR Software Components

This one-day course discusses AUTOSAR-compliant modeling and code generation using the Embedded Coder Support Package for AUTOSAR Standard. Workflows for top-down and bottom-up software development approaches are discussed in the context of Model-Based Design. This course is intended for automotive industry software developers and systems engineers who use Embedded Coder for automatic C/C++ code generation. Topics include:

- Generating Simulink models from existing ARXML system descriptions
- Configuring Simulink models for AUTOSAR compliant code generation
- Configuring AUTOSAR communication elements in a Simulink model
- Modeling AUTOSAR events in Simulink
- Creating calibration parameters
MathWorks Consulting Services
Get up and running fast. MathWorks Consulting Services - industry experience and MATLAB and Simulink expertise.

Why Choose MathWorks Consulting?
Working with MathWorks Consulting gives you the advantage of their years of project work, industry backgrounds, and deep MATLAB and Simulink knowledge.

Proven Solutions
MathWorks Consulting Services delivers reliable and effective solutions to solve your engineering challenges. Explore how MathWorks consultants work with you on a strategic level, understanding your unique business needs and developing innovative solutions.

Questions?
» Contact Consulting

Meet Our Team
Kirsty van Rynveld is a consultant engineer who focuses on data analysis, software development, and application deployment.

"MathWorks Consultants were well-qualified, professional, and fast. They understood not only the technical issues but also the business goals, which is essential when working on a core business system. We got more than we expected from MathWorks."

MATLAB EXPO 2017
And one last thing …
AUTOSAR – Antagonizing the „German Coast Guard“ Effect

Source: https://youtu.be/zkalf0odHs8 German Coast Guard Commercial 'We are Sinking' [HD]
Your feedback is valued.
Please complete the feedback form provided to you.