From Simulink to AUTOSAR
Enabling Innovation with Model-Based Design

MathWorks
AUTOMOTIVE CONFERENCES 2016

Shwetha B M – Product Marketing @ MathWorks US (sshwetha@mathworks.com)
Michael Fröstl – Pilot Engineering @ MathWorks DE (mfroestl@mathworks.de)
Agenda

- **Introduction to AUTOSAR**
  - Simulink approach to AUTOSAR
  - Overview of Modeling SWC’s & Modeling Styles

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

- **Advanced Topics – Top 5**
  - Startup, Reset, and Shutdown Modeling
  - Basic Software (BSW) Access
  - Lookup Tables (STD_AXIS, COM_AXIS, …)
  - Mode Management (ModeSenderPorts, ModeSwitchPoints, …)
  - Variability inside a Software Component

- **Getting Started**
Complexity of 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...

AUTOSAR Releases:
- 2.0: 90 files
- 2.1: 120 files
- 3.0: 125 files
- 3.1: 129 files
- 3.2: 136 files
- 4.0: 183 files
- 4.1: 198 files
- 4.2: 227 files

www.autosar.org
AUTOSAR Adoption
What is AUTOSAR?

AUTOSAR® (AUTomotive Open System ARchitecture) is an open and standardized automotive software architecture
AUTOSAR Members

9 Core Partners

BMW Group
BOSCH
DAIMLER
PSA PEUGEOT CITROEN
TOYOTA
VOLKSWAGEN AG

57 Premium Members

General
OEM

Generic Tier 1

Standard Software

Tools and Services

Semi-conductors

88 Associate Members

17 Attendees

11 Development Members

www.autosar.org
AUTOSAR Support from Embedded Coder and Simulink
What is the latest AUTOSAR Schema Version?

- 5.0
- 6.1
- 4.2
- 4.1
- 4.3
Agenda

- Introduction to AUTOSAR
  - Simulink approach to AUTOSAR
  - Overview of Modeling SWC’s & Modeling Styles
- AUTOSAR Design Workflows
  - Bottom Up, Top Down & Round Trip
- Advanced Topics – Top 5
  - Startup, Reset, and Shutdown Modeling
  - Basic Software (BSW) Access
  - Lookup Tables (STD_AXIS, COM_AXIS, …)
  - Mode Management (ModeSenderPorts, ModeSwitchPoints, …)
  - Variability inside a Software Component
- Getting Started
Simulink Approach to AUTOSAR

Available via web download

Simulink and Embedded Coder
+ AUTOSAR Support package for Embedded Coder

No separate AUTOSAR Blockset needed

Export

C Code and ARXML

Import

Code-generation through Mapping

Generate XML file for schema version

Maximum SHORT-NAME length: 32

- Use AUTOSAR compiler abstraction
- Support root-level matrix I/O using

4.2

Available via web download
Modeling AUTOSAR Communication

- Ports in an 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
Mapping Simulink to AUTOSAR

AR: Port

AR: Interface

AR: Data Element 1

AR: Data Element 2

AR: Data Element 3
Example Mapping to a Receiver Port

Add a New Simulink Port

Add a New Data Element to the Interface

Validate AUTOSAR to Simulink Mapping

Map Simulink Port to AUTOSAR Port
How AUTOSAR ports are categorized?

- Depending on the acceleration
- Depending on the velocity
- Depending on the direction
- Depending on the force
- Depending on the speed
Agenda

- **Introduction to AUTOSAR**
  - Simulink approach to AUTOSAR
  - Overview of Modeling SWC’s & Modeling Styles
- **AUTOSAR Design Workflows**
  - Bottom Up, Top Down & Round Trip
- **Advanced Topics – Top 5**
  - Startup, Reset, and Shutdown Modeling
  - Basic Software (BSW) Access
  - Lookup Tables (STD_AXIS, COM_AXIS, …)
  - Mode Management (ModeSenderPorts, ModeSwitchPoints, …)
  - Variability inside a Software Component
- **Getting Started**
Model AUTOSAR Components

Application Layer

- AUTOSAR Software Component 1
- ... ...
- AUTOSAR Software Component n

Virtual Functional Bus (VFB)

Periodic rate-based

Periodic and Asynchronous

Multi-rate and Asynchronous
Agenda

- Introduction to AUTOSAR
  - Simulink approach to AUTOSAR
  - Overview of Modeling SWC’s & Modeling Styles

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

- Advanced Topics – Top 5
  - Startup, Reset, and Shutdown Modeling
  - Basic Software (BSW) Access
  - Lookup Tables (STD_AXIS, COM_AXIS, …)
  - Mode Management (ModeSenderPorts, ModeSwitchPoints, …)
  - Variability inside a Software Component

- Getting Started
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

```matlab
%% 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');
```
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</td>
<td>2.1, 3.0, 3.1, 3.2 (Rev 3.2.2), 4.0, 4.1, 4.2 (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, 4.1 (Rev 4.1.1)</td>
</tr>
<tr>
<td>R2012a/b, R2013a/b, R2014a</td>
<td>2.1, 3.0, 3.1, 3.2, 4.0 (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</td>
</tr>
<tr>
<td>R2009a/b</td>
<td>2.0, 2.1</td>
</tr>
<tr>
<td>R2008a/b</td>
<td>2.0, 2.1</td>
</tr>
</tbody>
</table>
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
Trivia

Identify the Non-AUTOSAR Interface?

- Client-Server Interface
- NV Interface
- Sender-Receiver Interface
- Implicit Interface
- Mode-Switch Interface
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

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

%Create new model with interfaces
model = importerObj.createComponentAsModel('pkg/swc/ASWC');
Import with Internal Behavior

Top Down Workflow
Commands to create a skeleton model with internal behavior setup in the skeleton model (create with runnables)
### Updating Existing Models from ARXML

**V1.arxml**

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

% 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('rtwino_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

Merge SWC Description
Agenda

- **Introduction to AUTOSAR**
  - Simulink approach to AUTOSAR
  - Overview of Modeling SWC’s & Modeling Styles

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

- **Advanced Topics – Top 5**
  - Startup, Reset, and Shutdown Modeling
  - Basic Software (BSW) Access
  - Lookup Tables (STD_AXIS, COM_AXIS, …)
  - Mode Management (ModeSenderPorts, ModeSwitchPoints, …)
  - Variability inside a Software Component

- **Getting Started**
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

```matlab
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`
Lookup Tables (STD_AXIS, COM_AXIS, …)
Lookup Table Import Enhancements

Support Round-Trip Workflows by importing Lookup Tables.

- Import arxml files that contain CURVE, MAP, lookup tables in STD_AXIS and COM_AXIS configurations.
- Importer configures Lookup Table Blocks with new classes
  - Simulink.LookupTable
  - Simulink.Breakpoint
- Supports Integrated (STD_AXIS) and Distributed(COM_AXIS) lookups
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
Symbols in AUTOSAR component modeling

Create symbolic dimensions for array sizes

- Modify array size in system constants between simulations, without regenerating code
- Generate C and arxml code with symbols corresponding to variant array size

```c
/* SignalConversion: 'ConcatenateIn1' */
for (i = 0; i < Rte.SysCon_SymDimA; i++) {
    VectorConcatenate[i] = tmp[i];
}
```

```xml
<MAX-NUMBER-OF-ELEMENTS BINDING-TIME="PRE-COMPILE-TIME">
    <SYSC-REF DEST="SW-SYSTEMCONST"/>
</MAX-NUMBER-OF-ELEMENTS>
```
Agenda

- **Introduction to AUTOSAR**
  - Simulink approach to AUTOSAR
  - Overview of Modeling SWC’s & Modeling Styles

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

- **Advanced Topics – Top 5**
  - Startup, Reset, and Shutdown Modeling
  - Basic Software (BSW) Access
  - Lookup Tables (STD_AXIS, COM_AXIS, …)
  - Mode Management (ModeSenderPorts, ModeSwitchPoints, …)
  - Variability inside a Software Component

- **Getting Started**
AUTOSAR Support from Embedded Coder
Author and develop AUTOSAR software components for automotive systems

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

by MathWorks Embedded Coder Team
03 Oct 2014 (Updated 02 May 2018)

Develop AUTOSAR software components for automotive

Watch this File

Description
Installer file for Embedded Coder® Support Package for AUTOSAR Standard. For more details on capabilities for this support package, please visit:

www.mathworks.com/hardware-support/autosar

Opening the .mlpkginstall file from your operating system or from within MATLAB will initiate the installation process for this Hardware Support Package and acquire the latest support package available for the release you have.

This .mlpkginstall file is functional for R2014b and beyond.

**Feature highlights for new releases are shown in 'Updates' section at bottom of this page**

Required Products
- Simulink Coder
- Embedded Coder
- Simulink
- MATLAB Coder
- MATLAB

MATLAB release
MATLAB 9.0 (R2016a)

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

See detailed course outline.
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 business goals.

Meet Our Team

Kirsty van Ryneveld is a consultant engineer who focuses on data analysis, software development, and application deployment.

Questions?

Contact Consulting
And one last thing …
AUTOSAR – Antagonizing the „German Coast Guard“ Effect

Source: https://youtu.be/zkalf0odHs8 German Coast Guard Commercial ‘We are Sinking’ [HD]