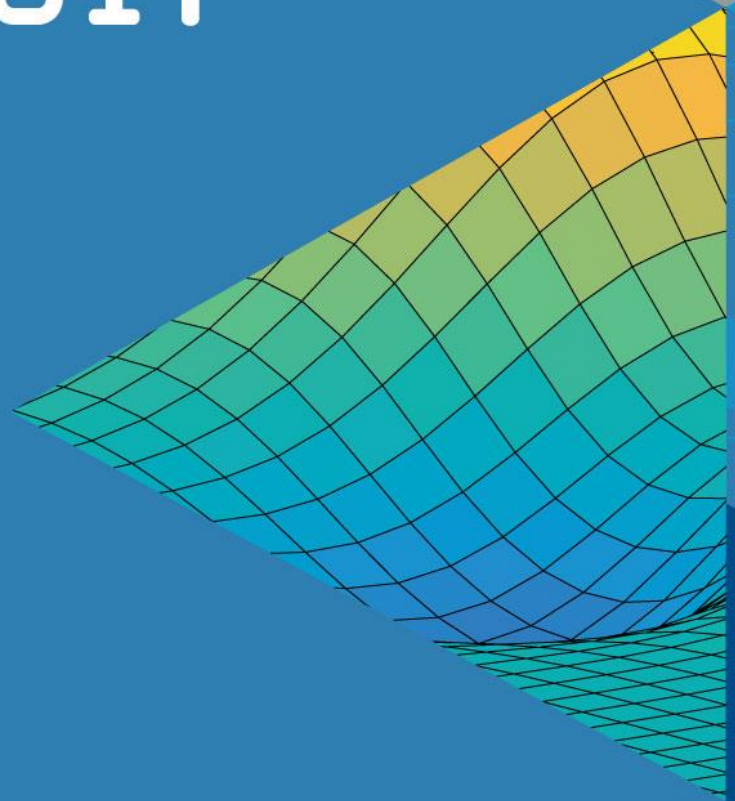
MATLAB EXPO 2017 KOREA

4월 27일, 서울

등록 하기 matlabexpo.co.kr



From Simulink to AUTOSAR Code

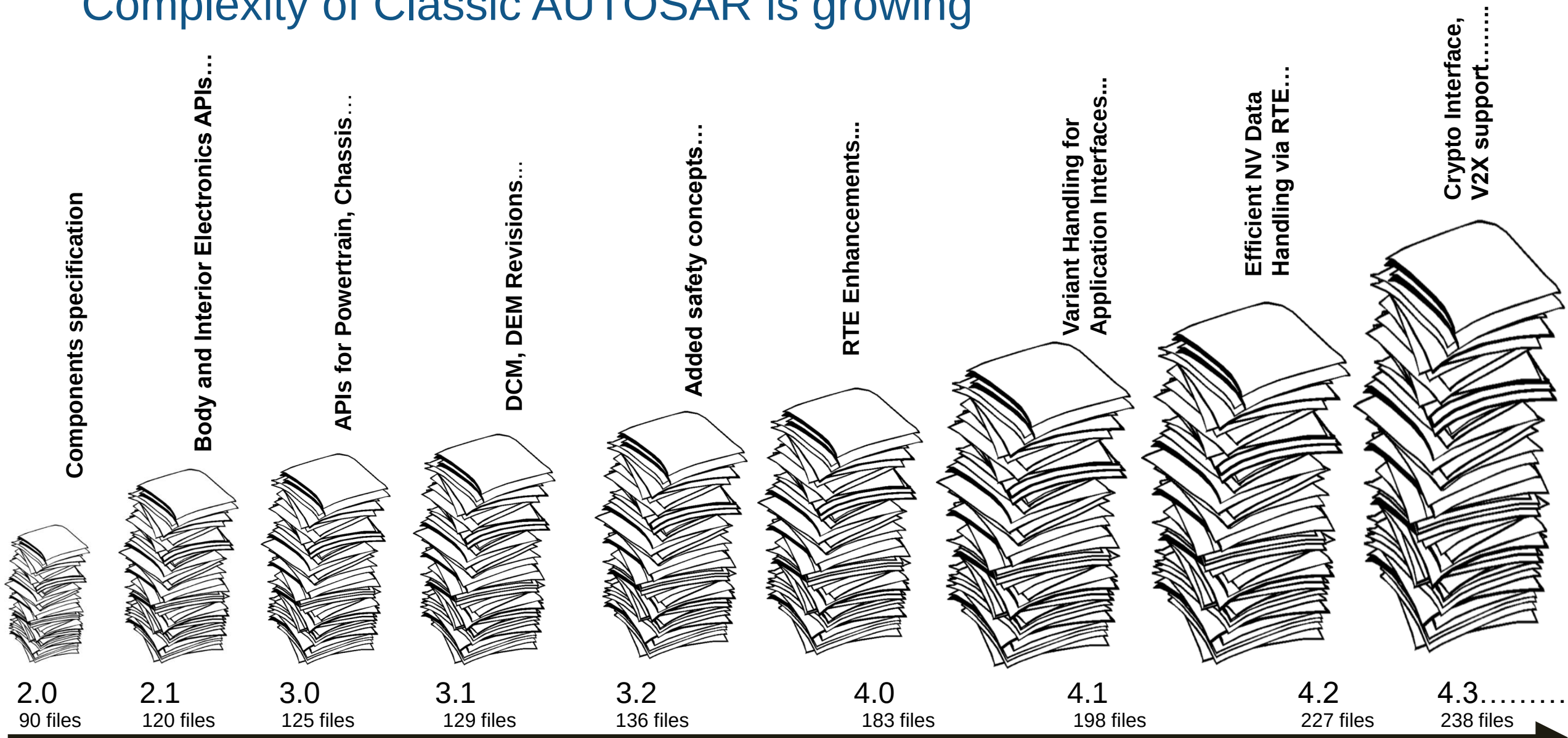
김종헌 부장
Senior Application Engineer
MathWorks Korea

Agenda

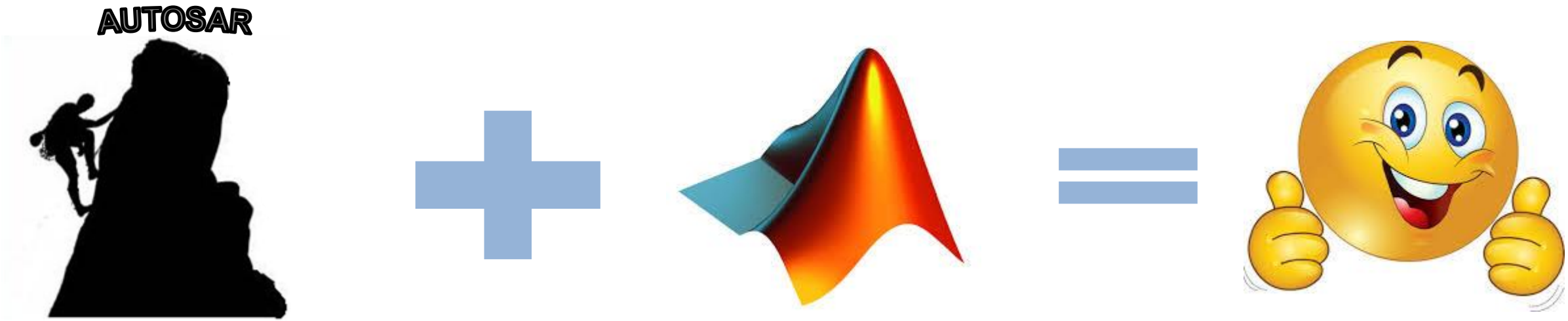
- Introduction to AUTOSAR Standards
  - Simulink approach to AUTOSAR

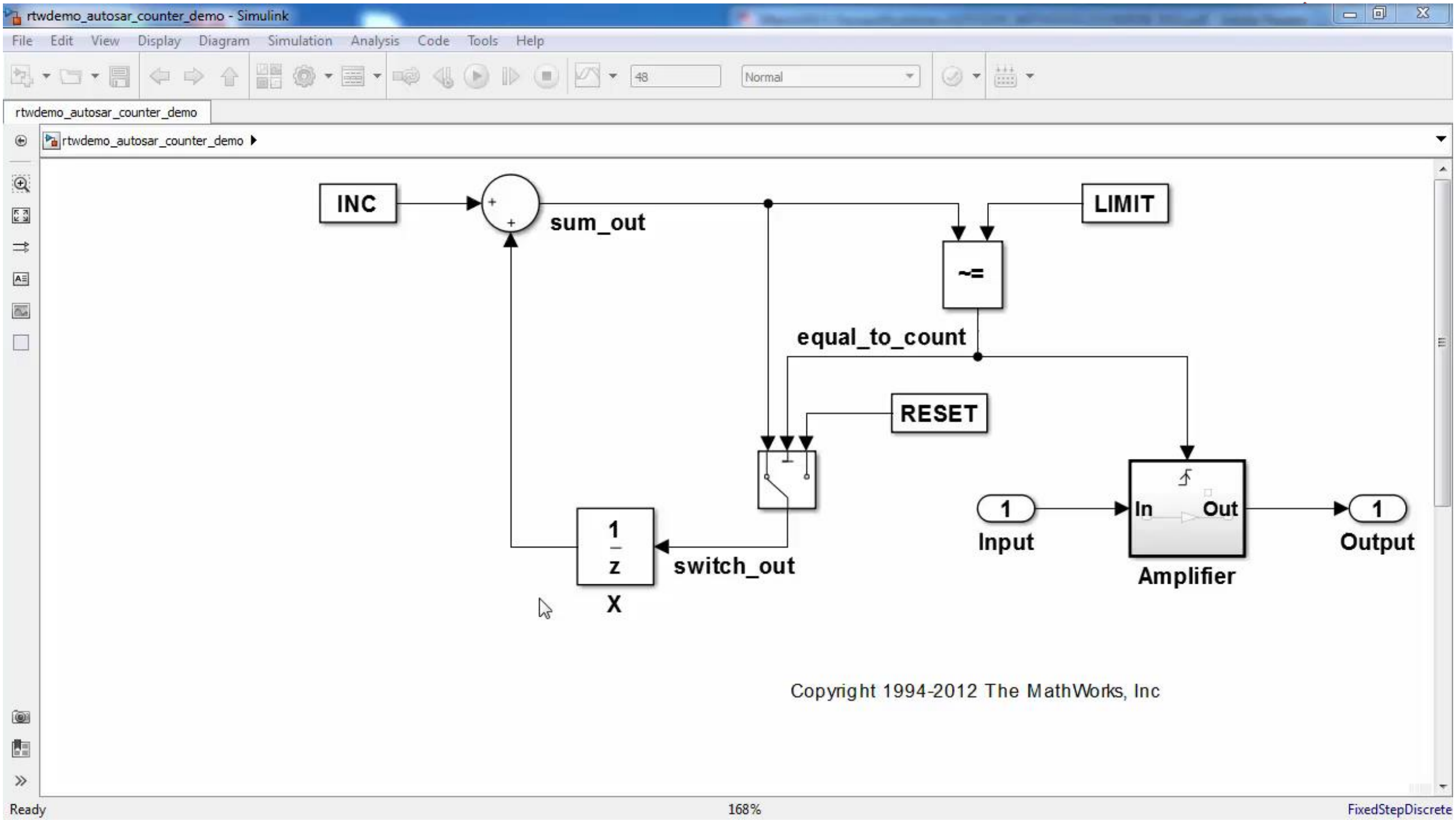
- AUTOSAR design workflows
    - Bottom Up
    - Top Down
    - Round trip
  - Overview:
    - Modeling AUTOSAR Components / Attributes
    - Components, Runnables and Events
    - Modeling styles

Complexity of Classic AUTOSAR is growing



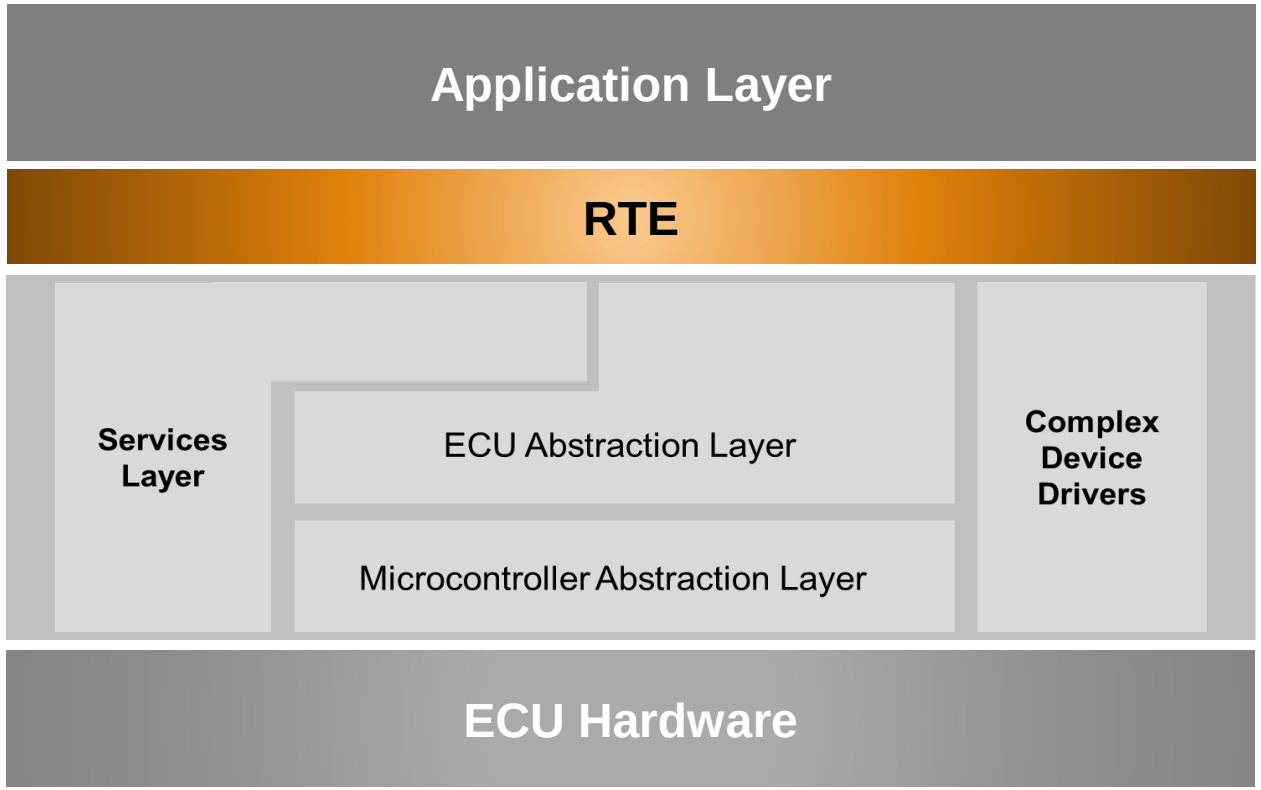
AUTOSAR Adoption




rtwdemo_autosar_counter_demo - Simulink
File Edit View Display Diagram Simulation Analysis Code Tools Help
48
Normal
rtwdemo_autosar_counter_demo
rtwdemo_autosar_counter_demo
INC
sum_out
LIMIT
~=
equal_to_count
RESET
1
z
X
switch_out
1
Input
In
Out
Amplifier
1
Output
Copyright 1994-2012 The MathWorks, Inc
Ready
168%
FixedStepDiscrete

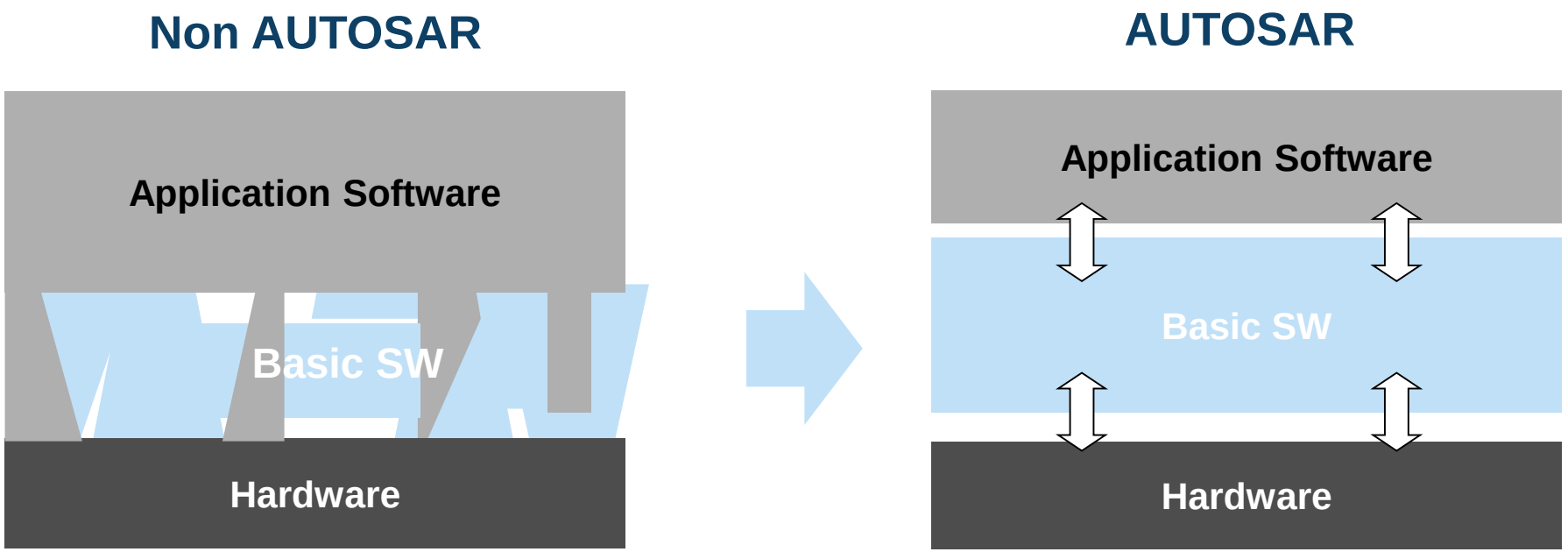
What is AUTOSAR?

AUTOSAR® (**AUT**omotive **O**pen **S**ystem **AR**chitecture) is an open and standardized automotive software architecture



www.autosar.org

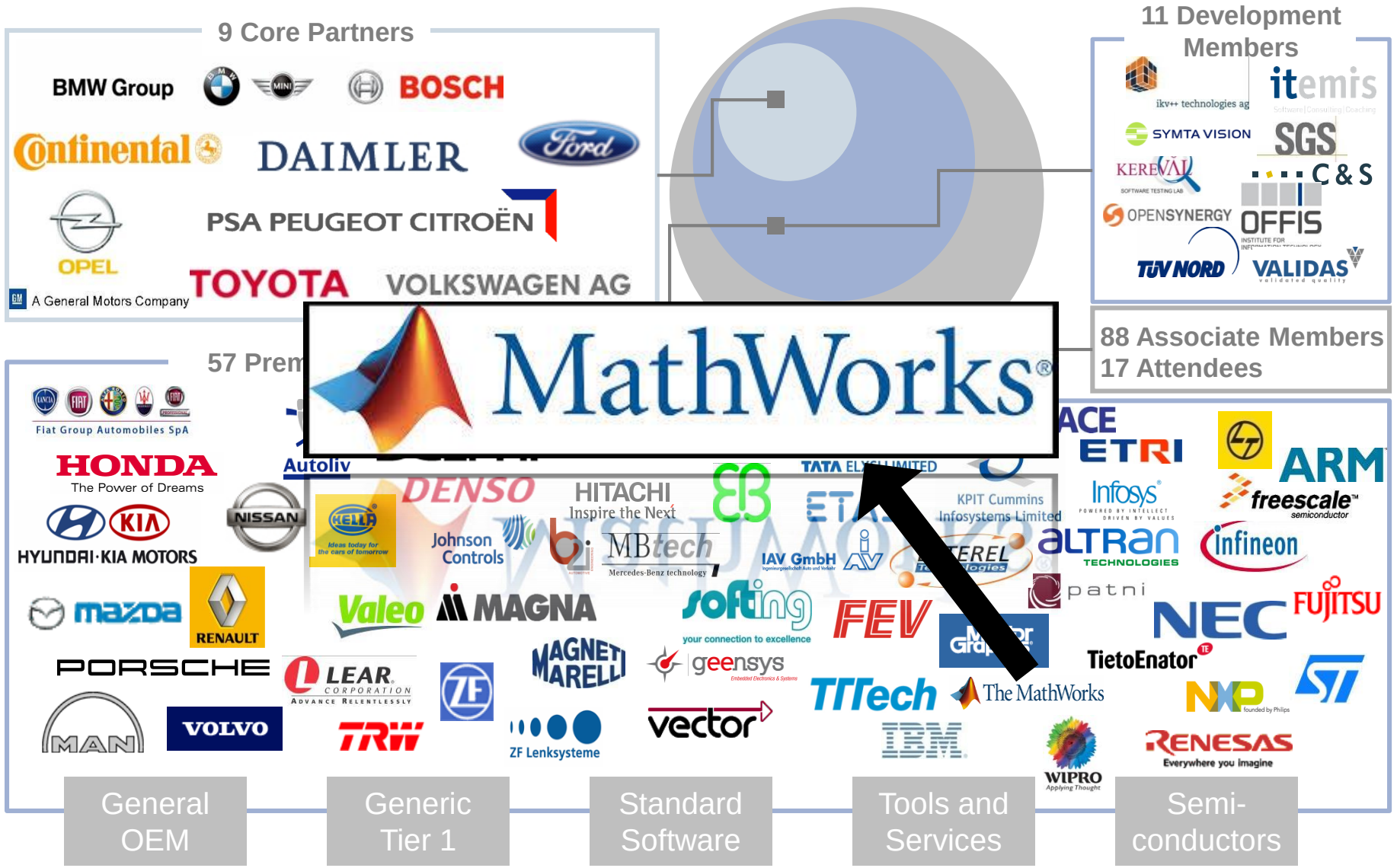
AUTOSAR Vision



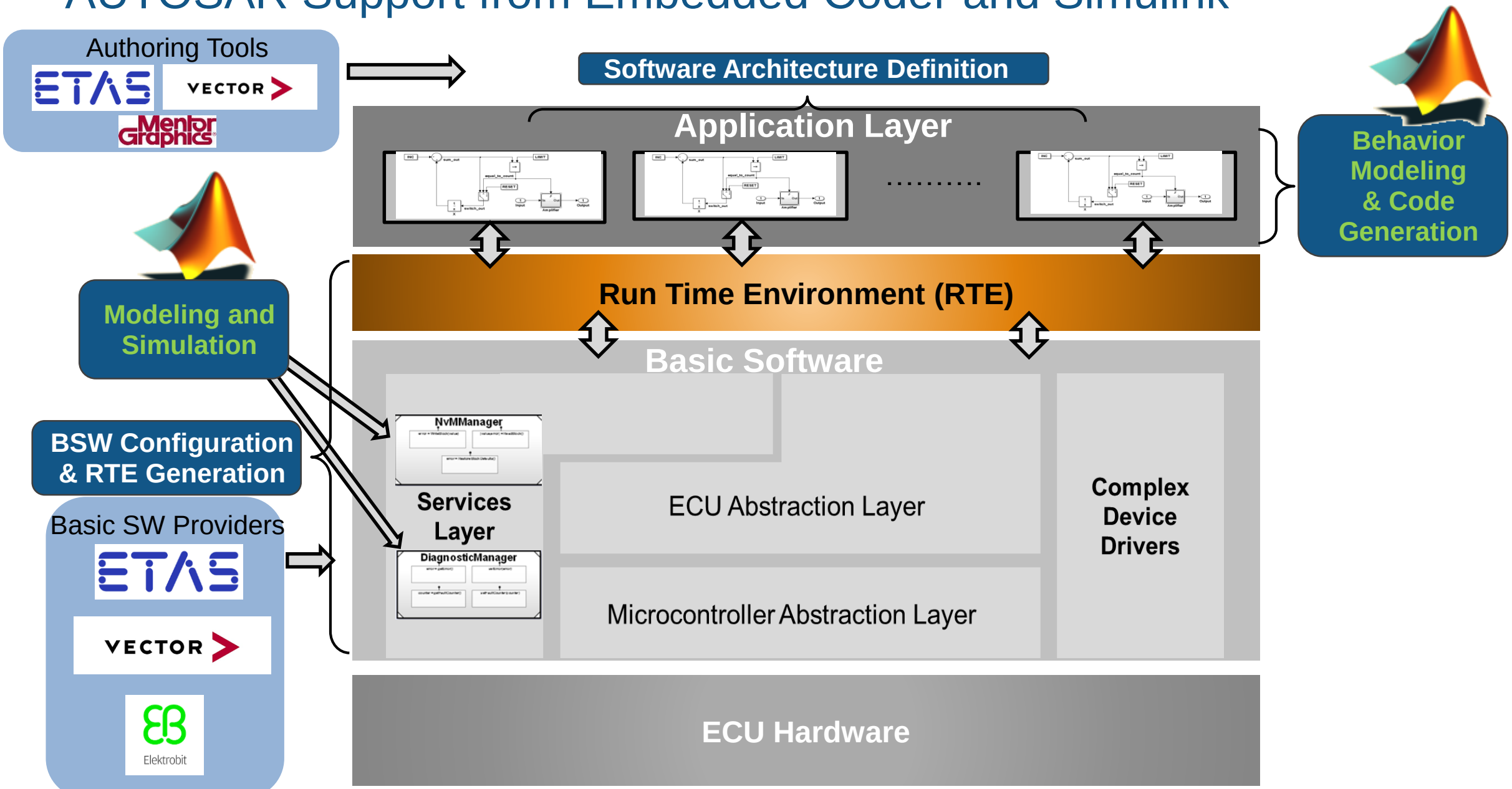
AUTOSAR Slogan-

“Cooperate on Standards – compete on implementation”

AUTOSAR Members



AUTOSAR Support from Embedded Coder and Simulink

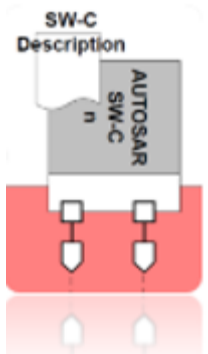


Simulink Approach to AUTOSAR

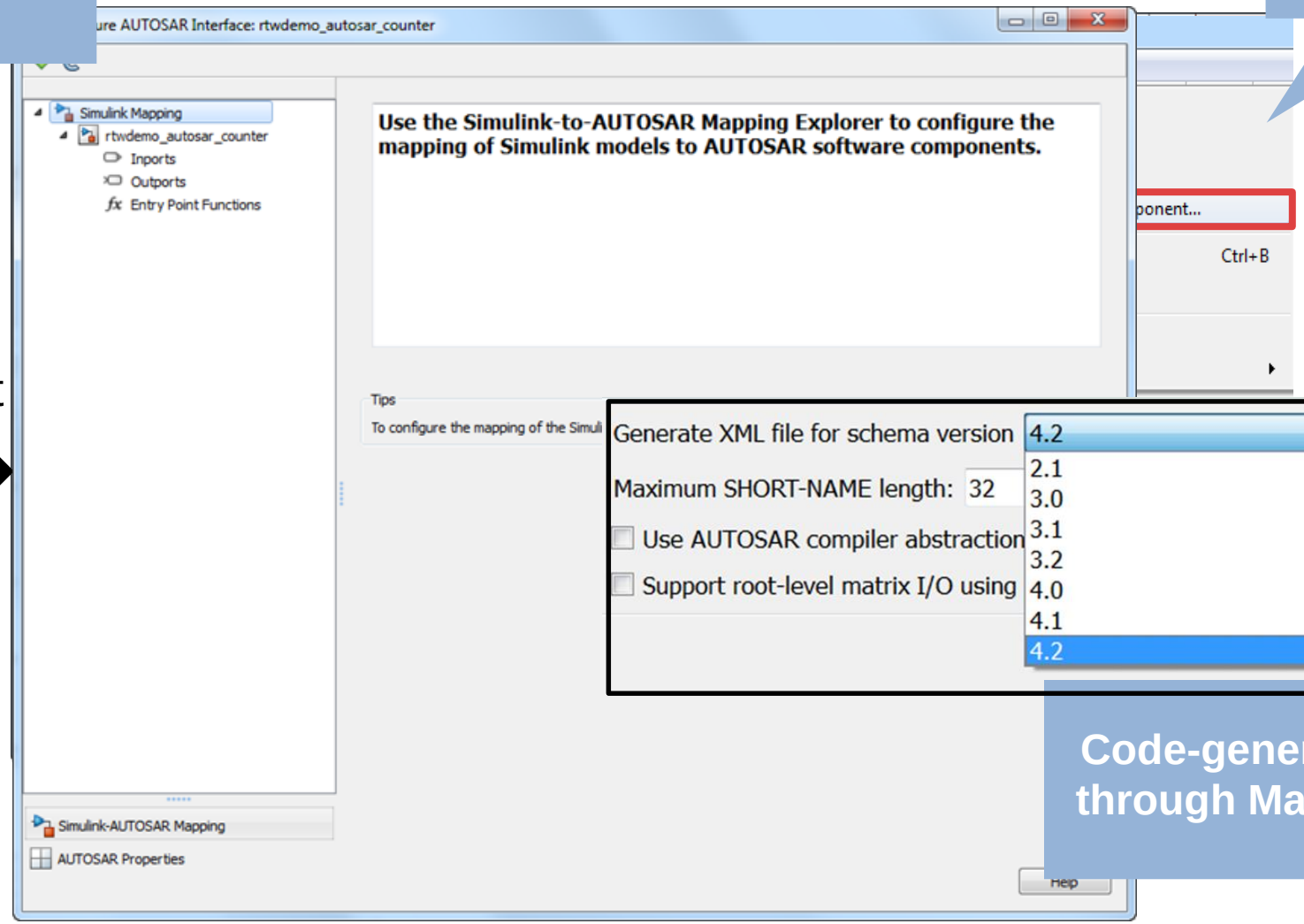
Available via web download

Simulink and Embedded Coder
+ **AUTOSAR Support package for Embedded Coder**

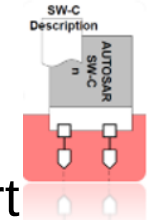
No separate AUTOSAR Blockset needed



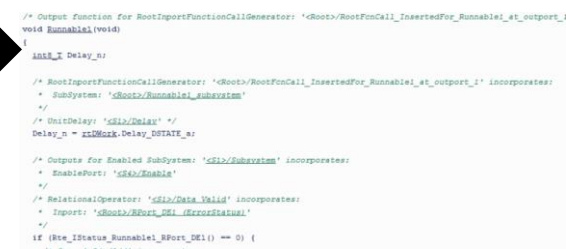
Import



Export

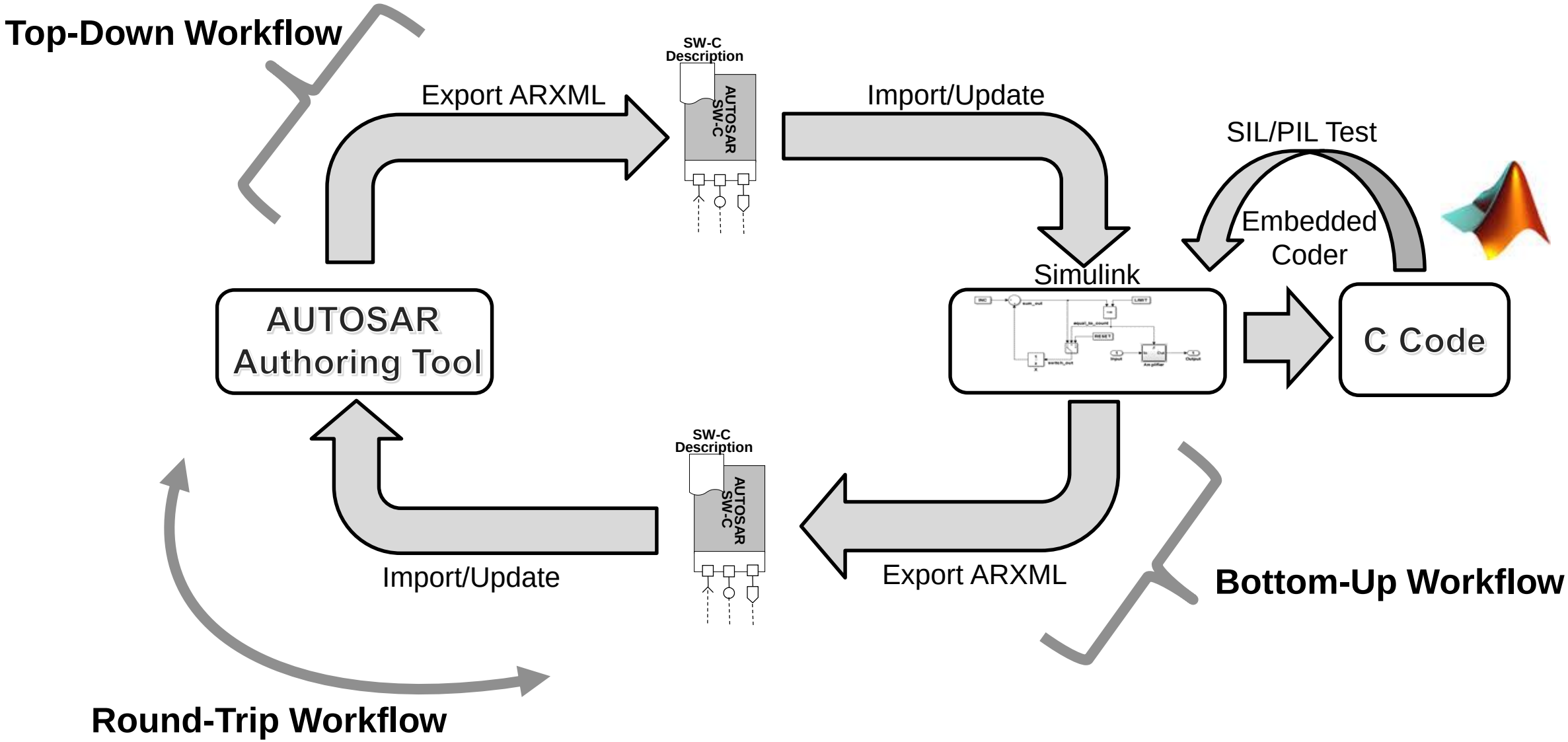


C Code and ARXML



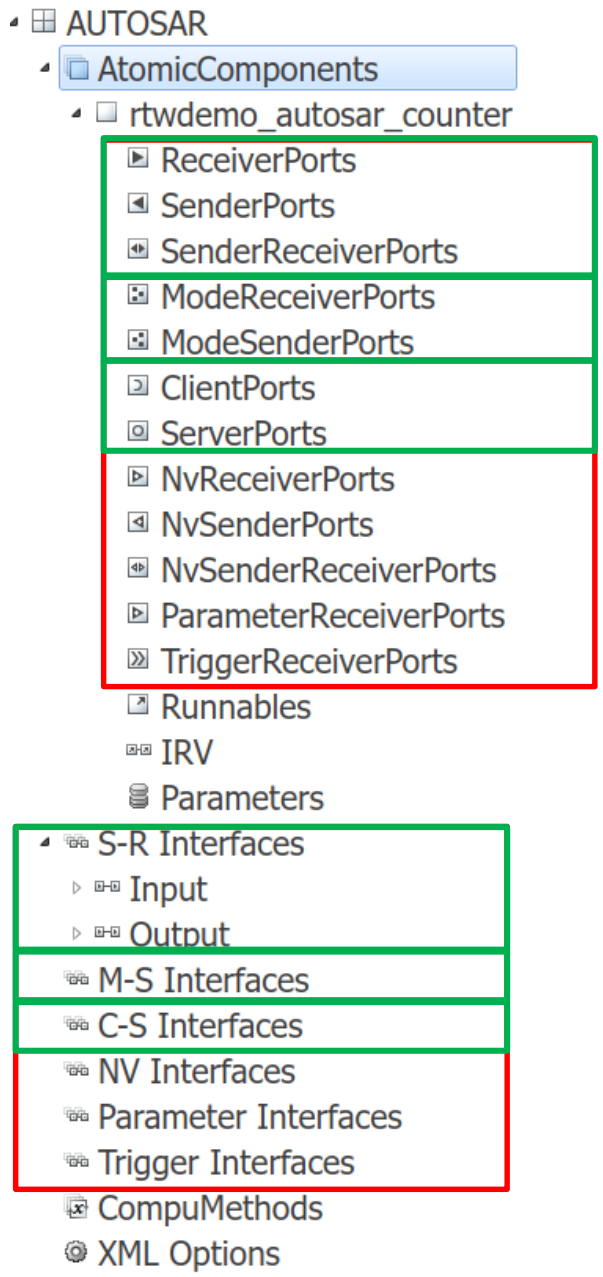
Code-generation through Mapping

Supported AUTOSAR Design Workflows



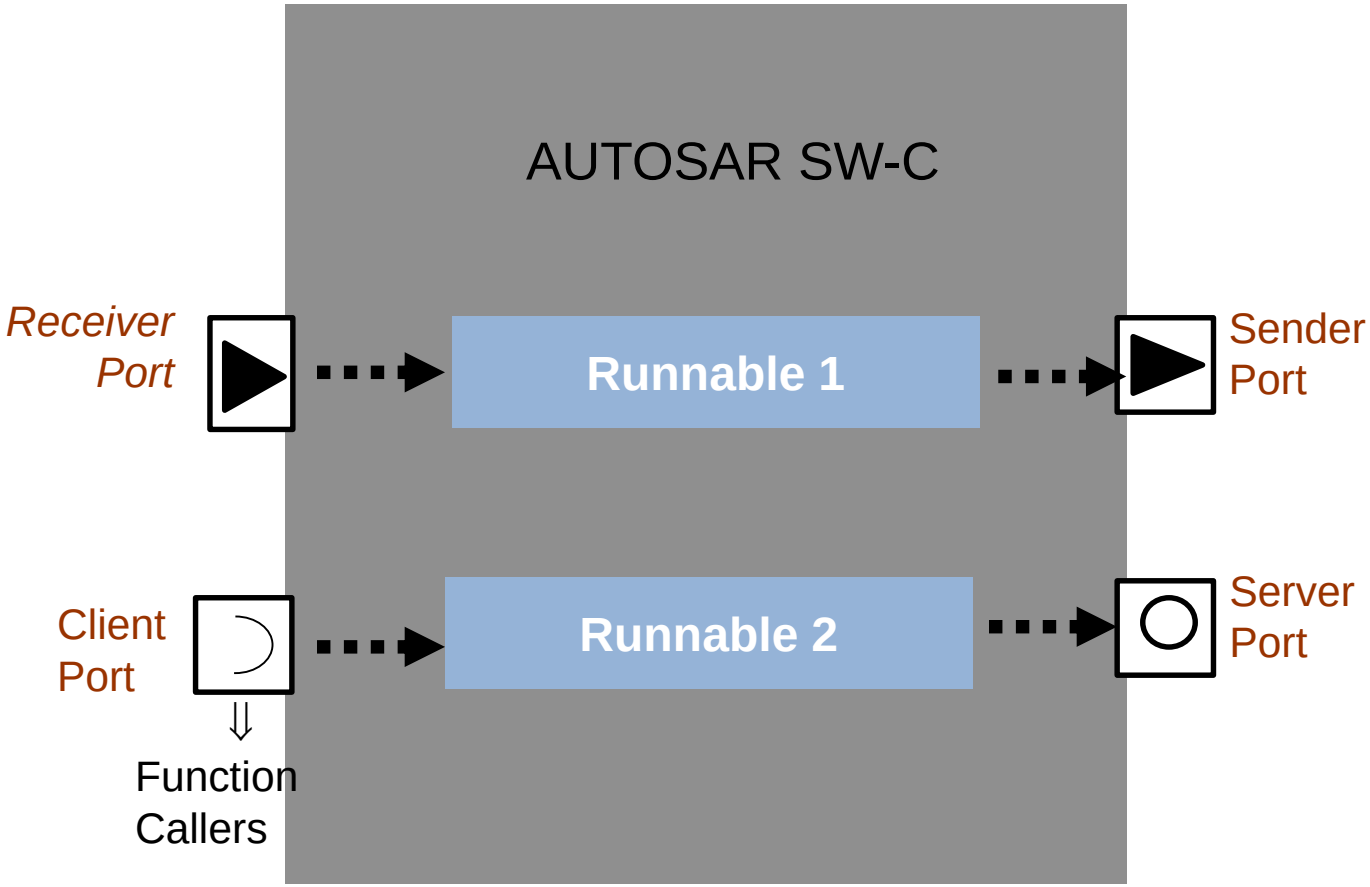
Modeling AUTOSAR Communication

- 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



Runnable Entities

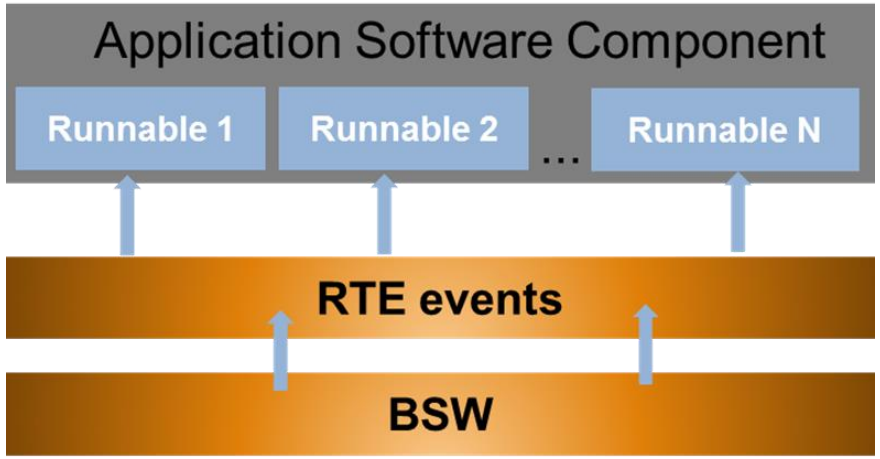
- Each AUTOSAR SW-C is composed by one or more runnables/runnable entities

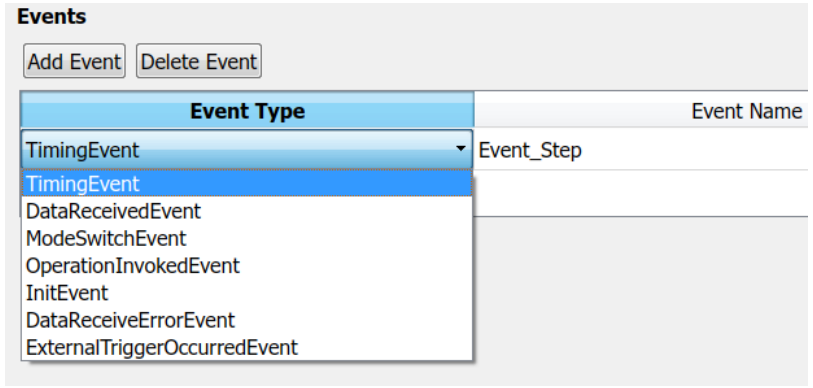


Supported Events for a Runnable

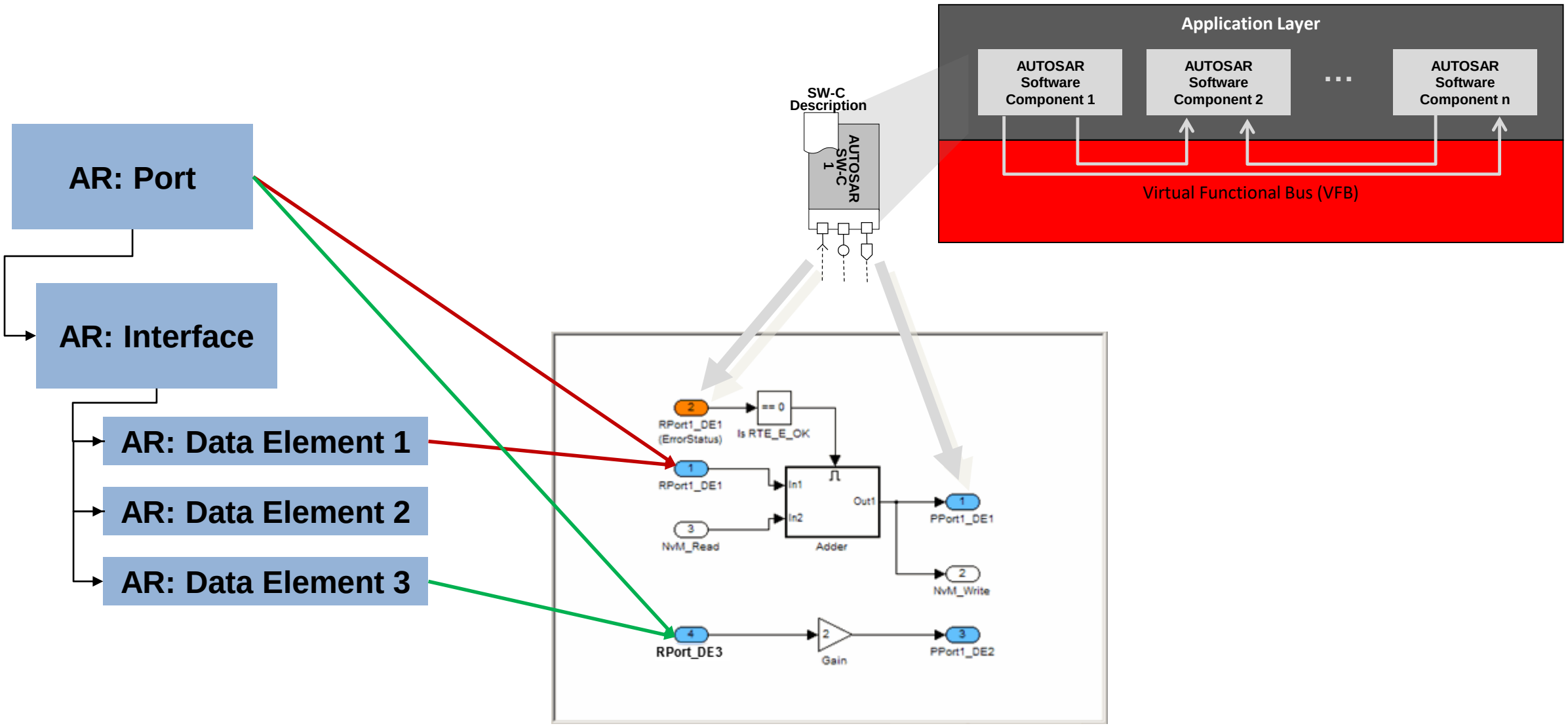
Each Runnable should have at least one event attached

- TimingEvent - Periodically scheduled Runnables
- DataReceivedEvent - Trigger the runnable when data is received
- ModeSwitchEvent - Triggered onEntry, onExit, or onTransition
- OperationInvokedEvent - Client-server type event
- InitEvent - designate an AUTOSAR runnable as an initialization runnable, and then map an initialization function to the runnable.
- DataReceiveErrorEvent - when the communication layer reports an error in data reception by the receiver component
- ExternalTriggerOccuredEvent – used to activate a runnable in an SWC as result of an explicit trigger by a runnable entity of some other SWC



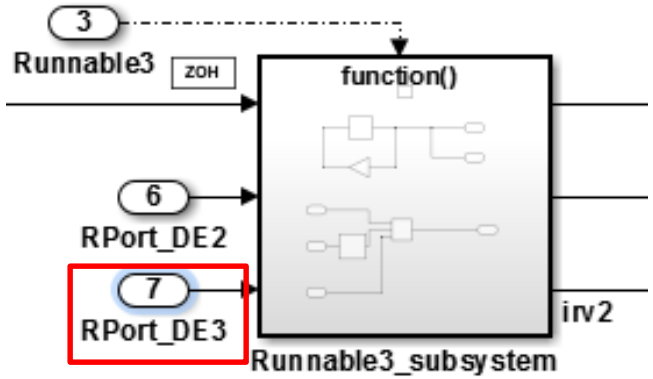


Mapping Simulink to AUTOSAR

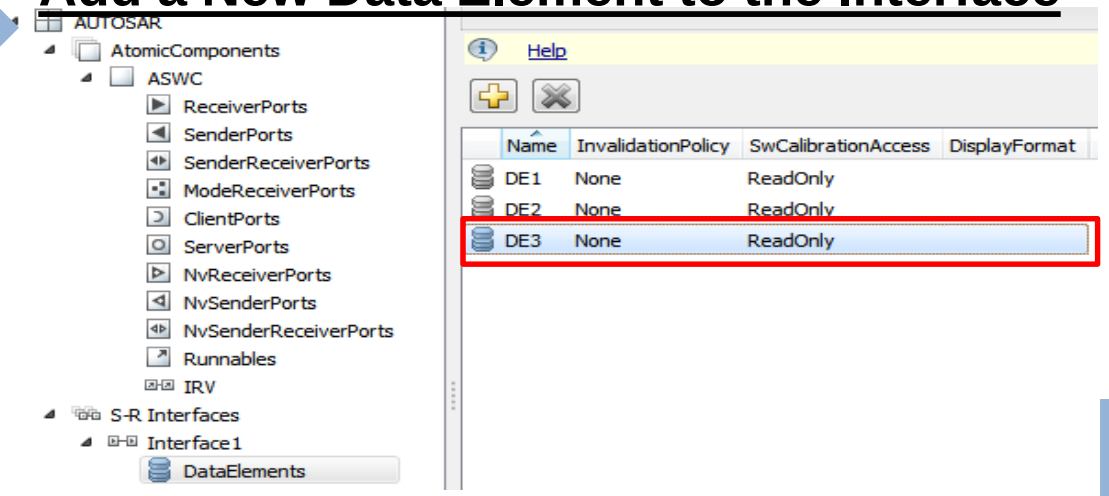


Example Mapping to a Receiver Port

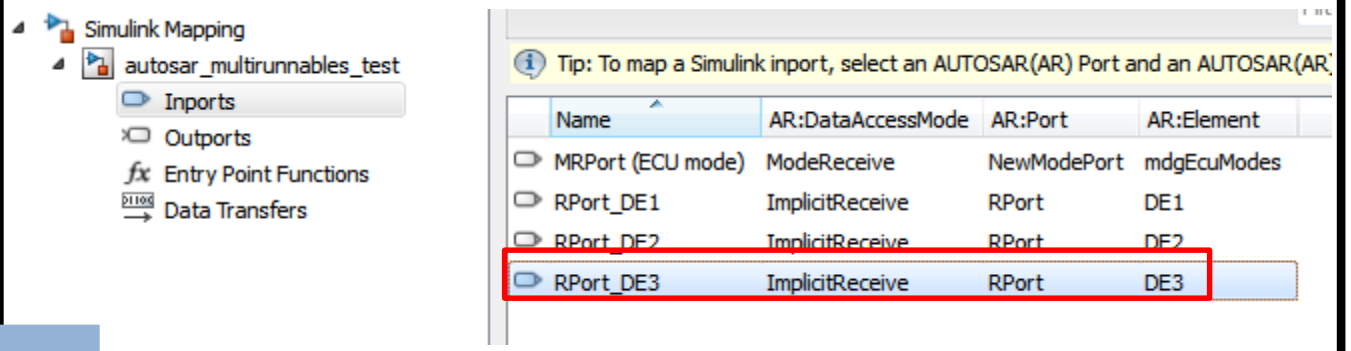
Add a New Simulink Port



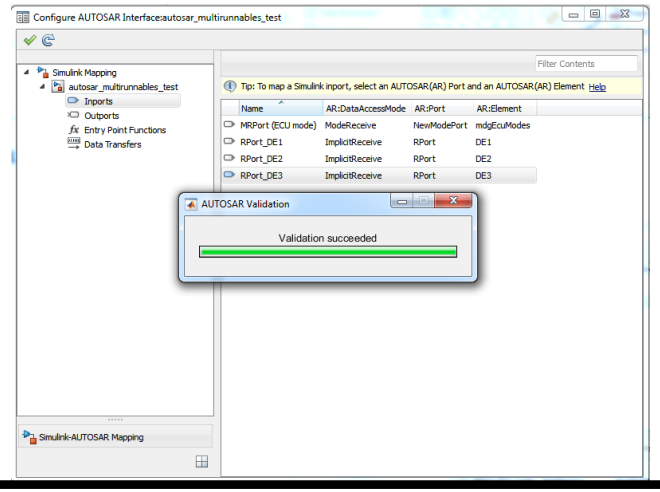
Add a New Data Element to the Interface



Map Simulink Port to AUTOSAR Port



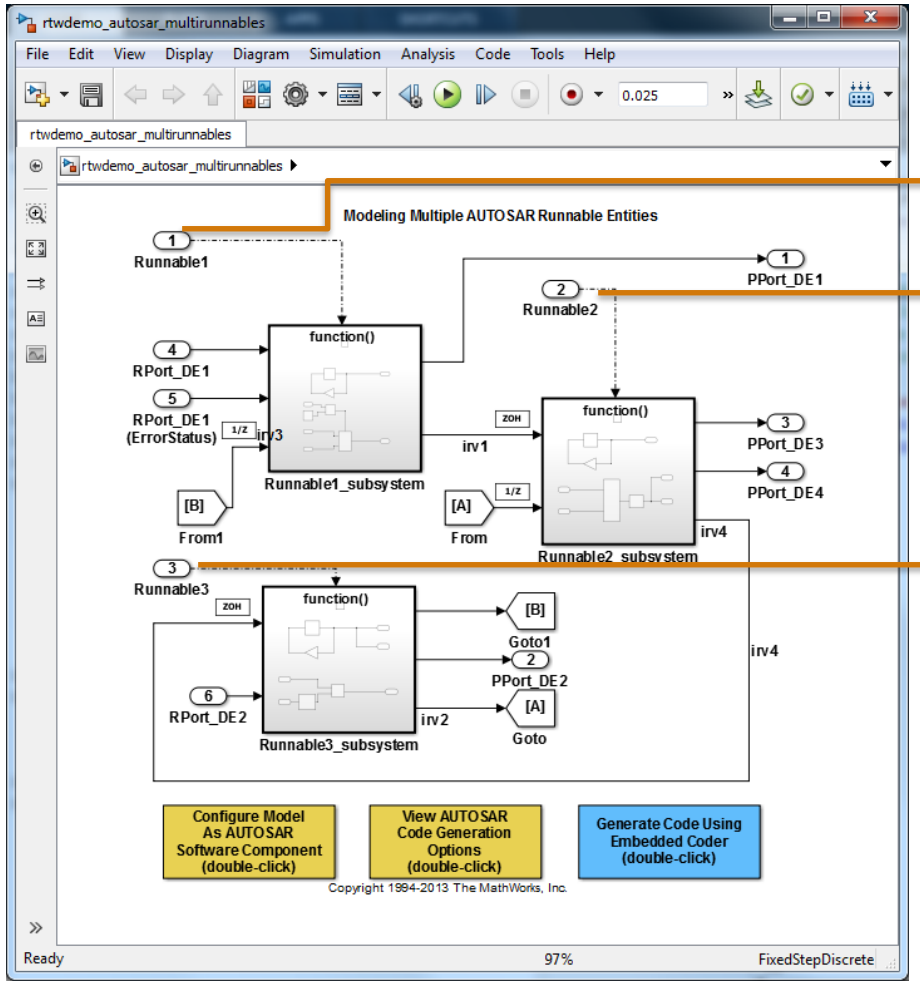
Validate AUTOSAR to Simulink Mapping

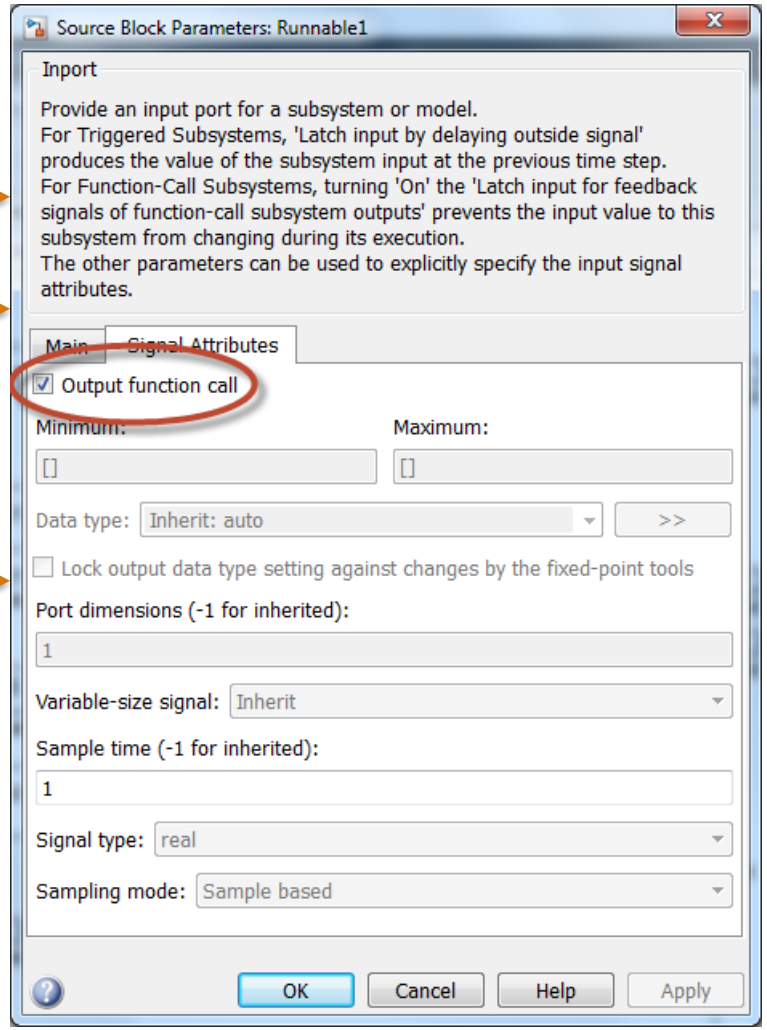


Using MATLAB for automating common tasks

```
%% 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');
```

Modeling Multiple Entry-Point Functions



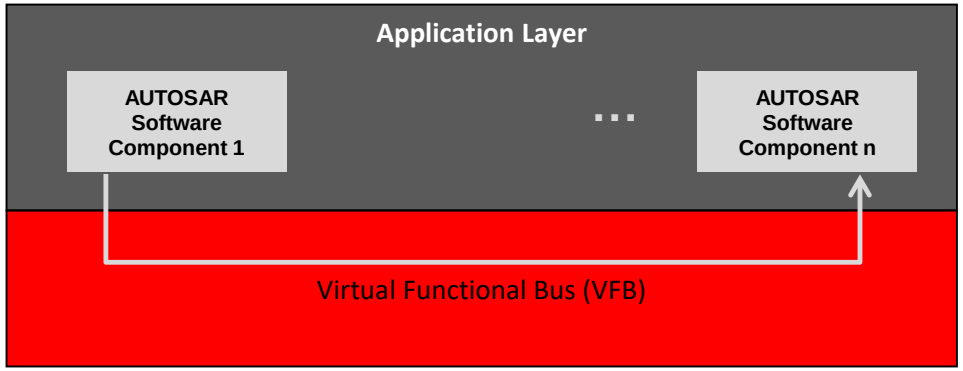


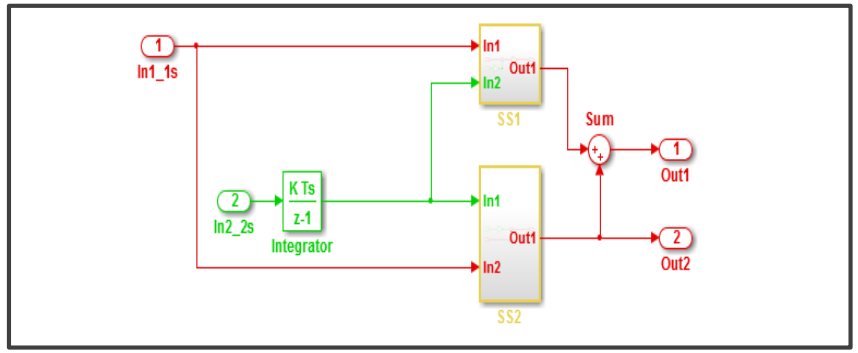
For the multi-runnable modeling pattern, each input port that represents a runnable trigger event will need to have the check box for Output function call checked as part of the inport properties.

This allows for function triggers from a test harness model to be passed across a model reference boundary.

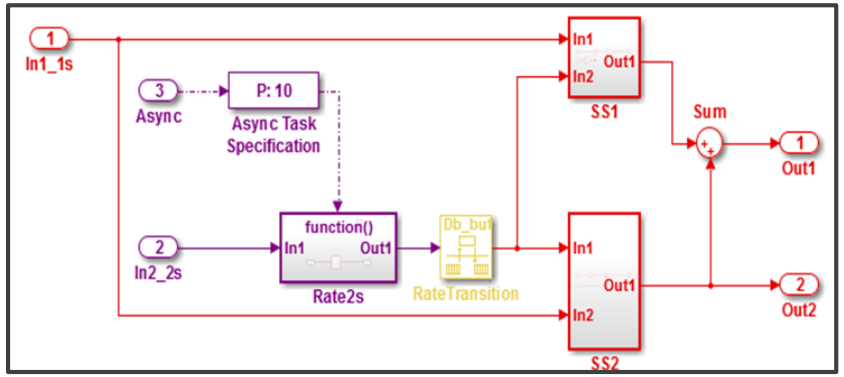
>> rtwdemo_autosar_multirunnables

Model AUTOSAR Components

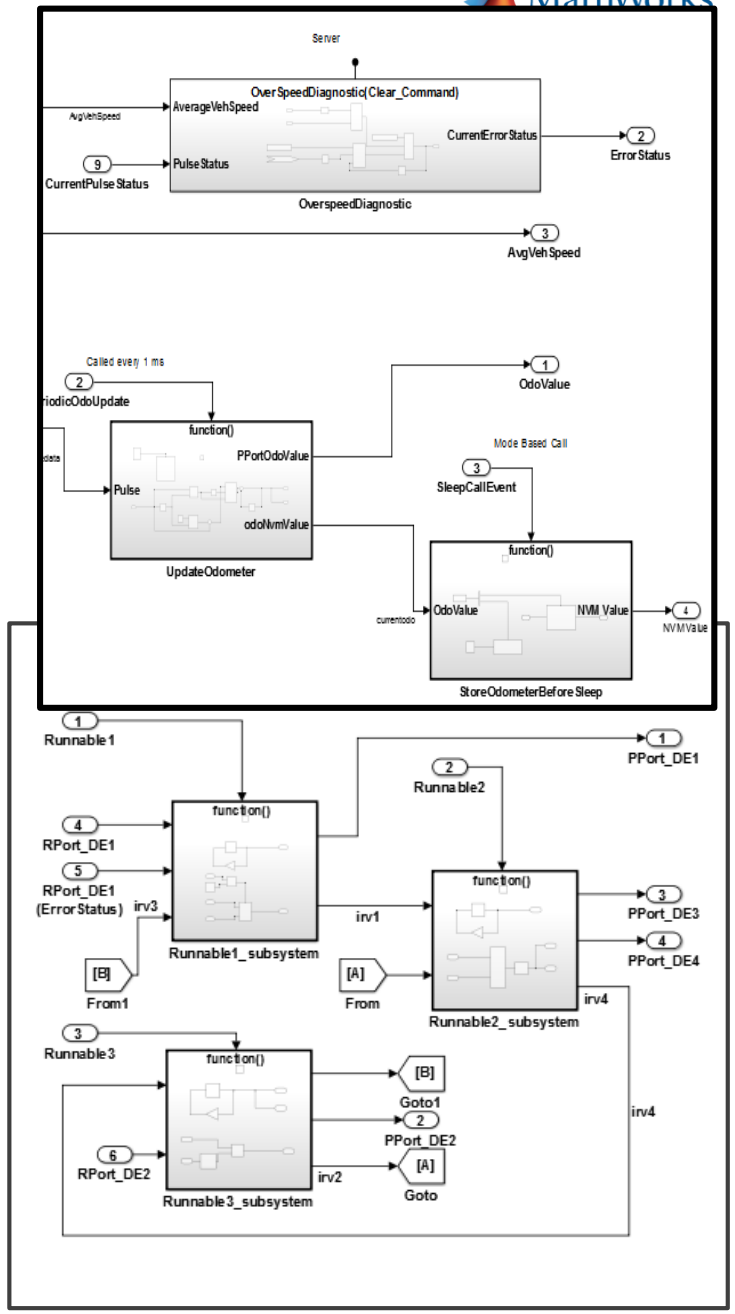




Periodic rate-based



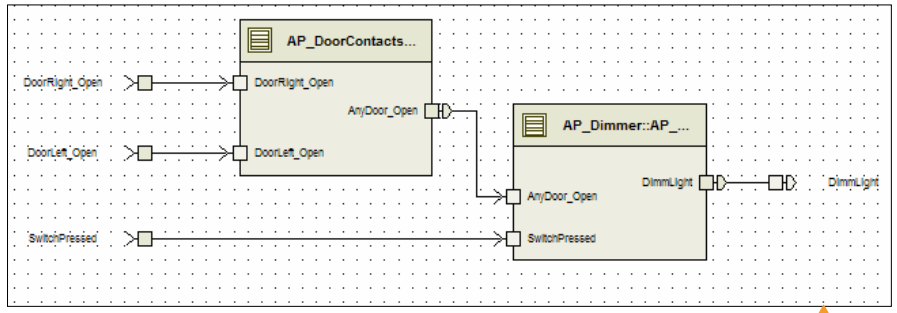
Periodic and Asynchronous



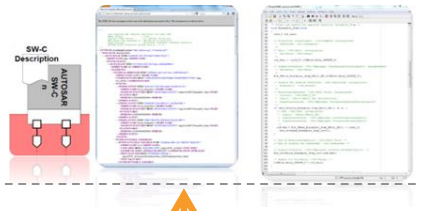
Multi-rate and Asynchronous

Bottom-Up Workflow (Starting from Simulink)

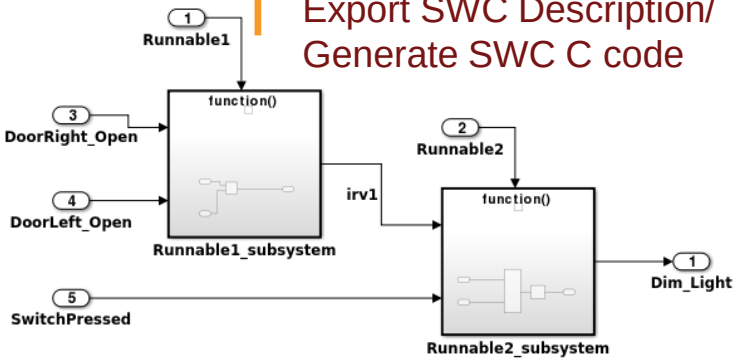
AUTOSAR Authoring Tool

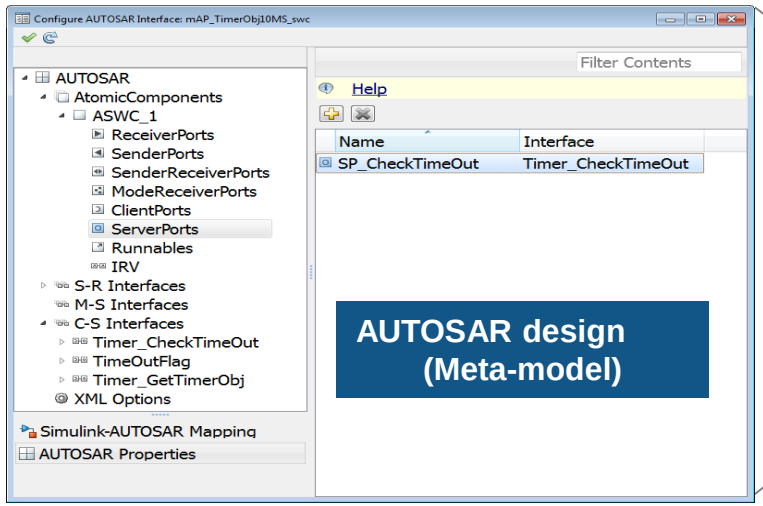


Import
SWC Description



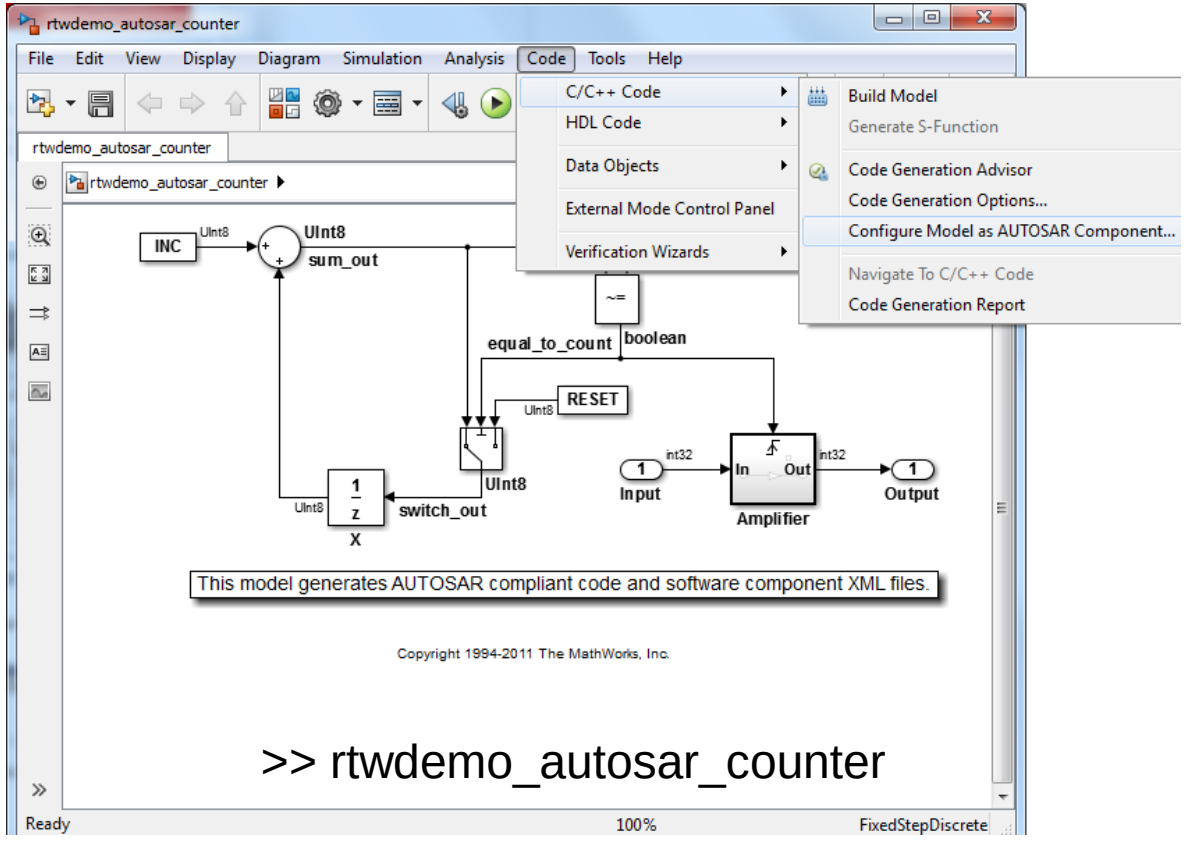
Export SWC Description/
Generate SWC C code



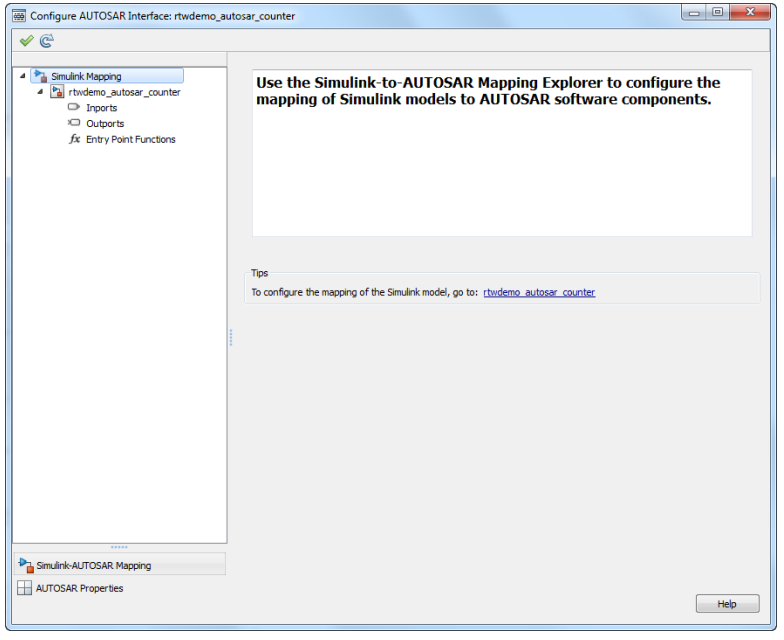


Launch AUTOSAR Configuration

- The method for Configuring AUTOSAR Properties is
  - selecting the Code Menu and then selecting C/C++ Code
  - Configure Model as an AUTOSAR Component. (This will bring up a dialog screen as shown.)



Configure AUTOSAR Interface (User Dialog)



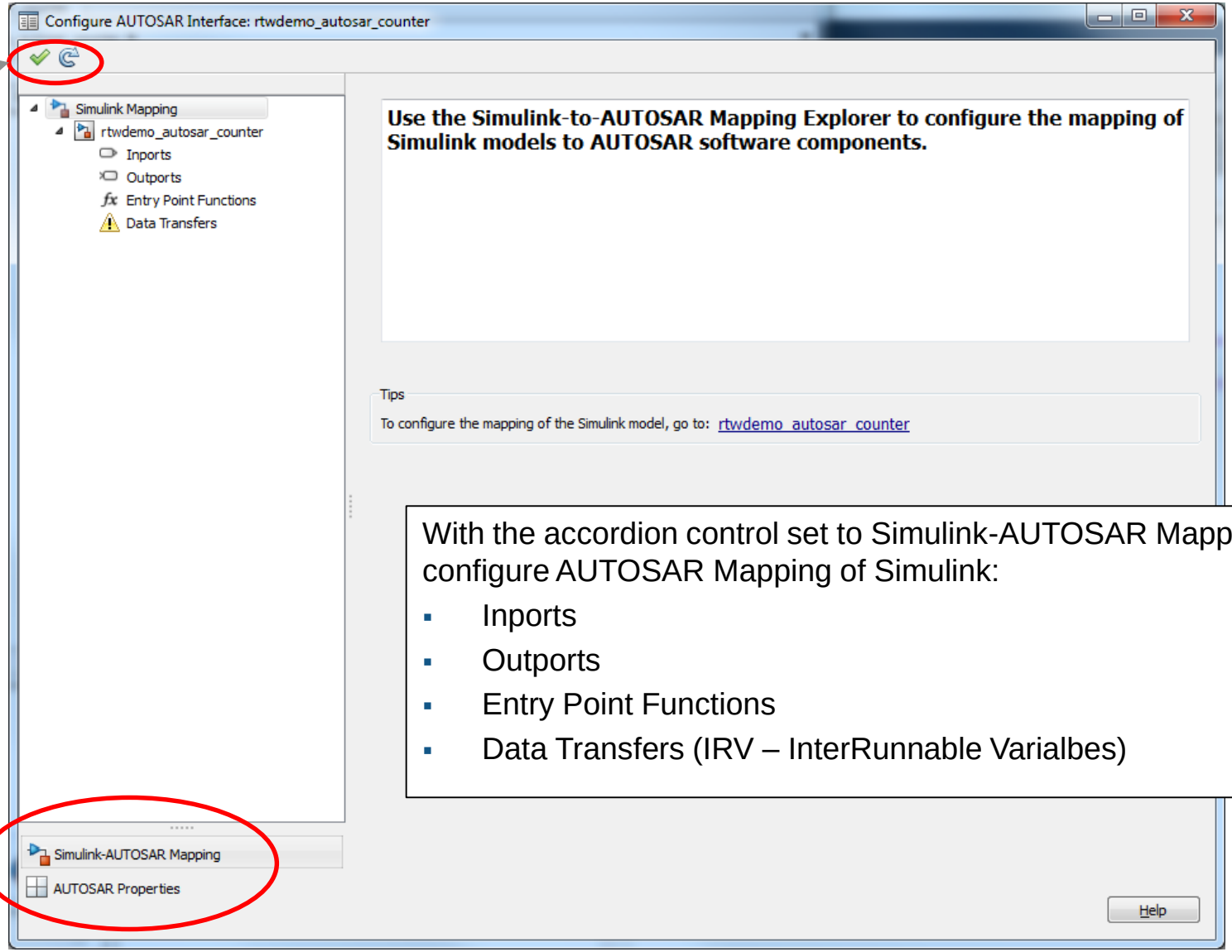
Simulink-AUTOSAR Mapping Editor

View / Edit AUTOSAR Properties and Simulink Mappings

The Green Check mark is to Validate the AUTOSAR Configuration.

The blue Circle arrow is to refresh the Simulink elements.

The AUTOSAR Interface Configuration is split between two areas: Simulink-AUTOSAR Mapping and AUTOSAR Properties. See this accordion based UI control at the bottom left corner of the dialog screen

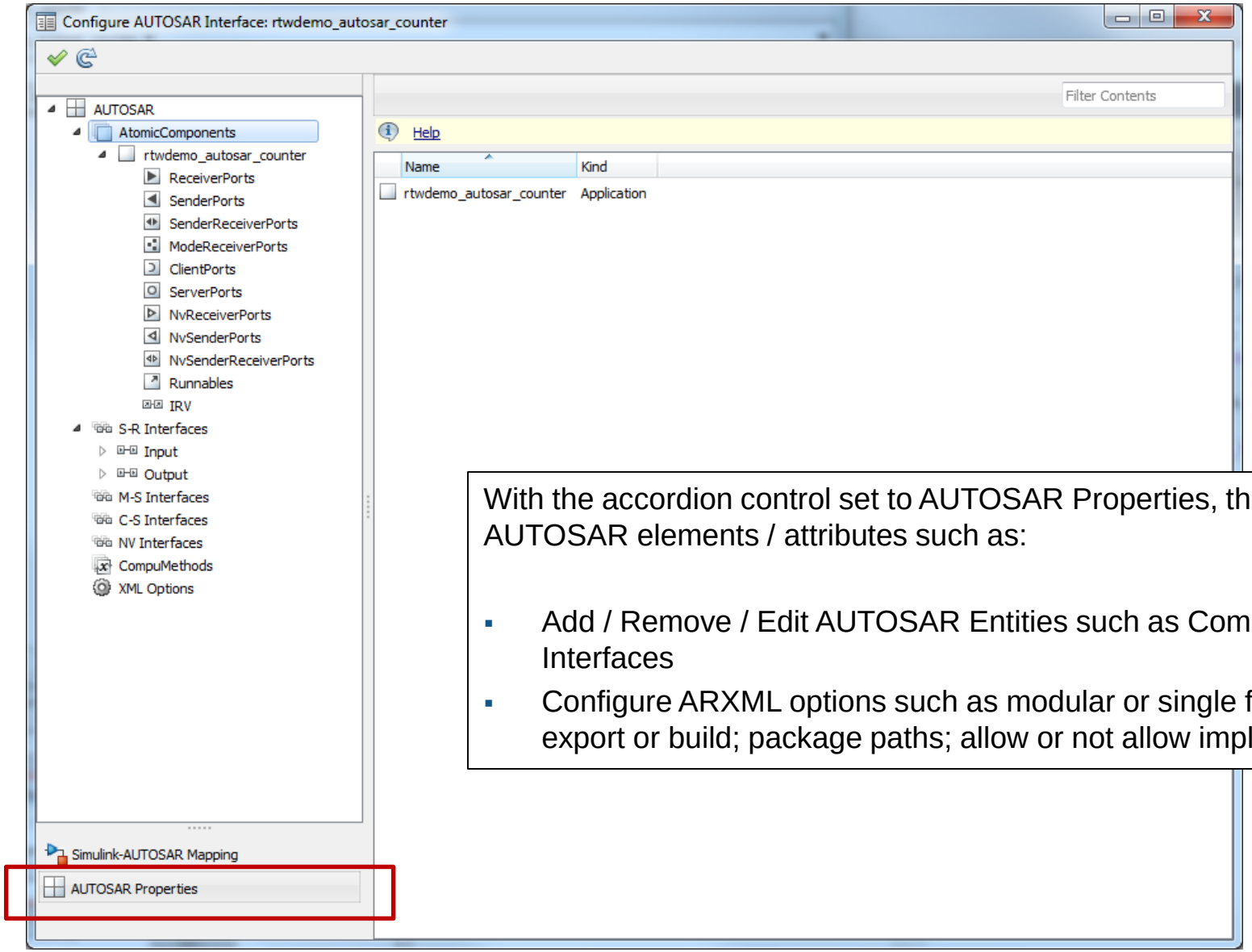


With the accordion control set to Simulink-AUTOSAR Mapping, configure AUTOSAR Mapping of Simulink:

- Inports
- Outports
- Entry Point Functions
- Data Transfers (IRV – InterRunnable Varialbes)

Editing AUTOSAR Properties

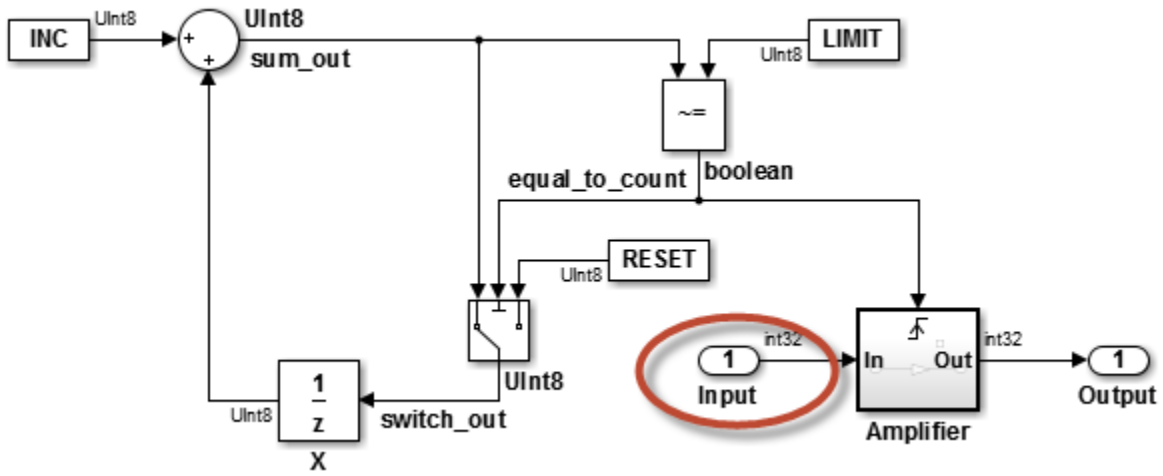
View / Edit AUTOSAR Properties and Simulink Mappings

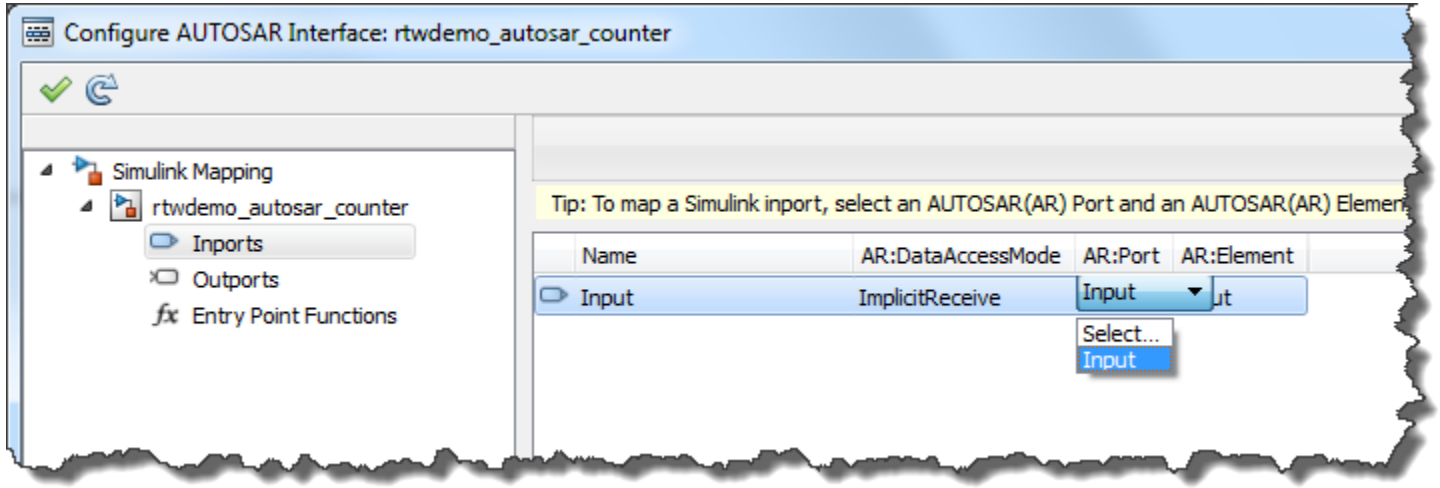


With the accordion control set to AUTOSAR Properties, the user can configure AUTOSAR elements / attributes such as:

- Add / Remove / Edit AUTOSAR Entities such as Components / Ports and Interfaces
- Configure ARXML options such as modular or single file generated on export or build; package paths; allow or not allow implementation types.

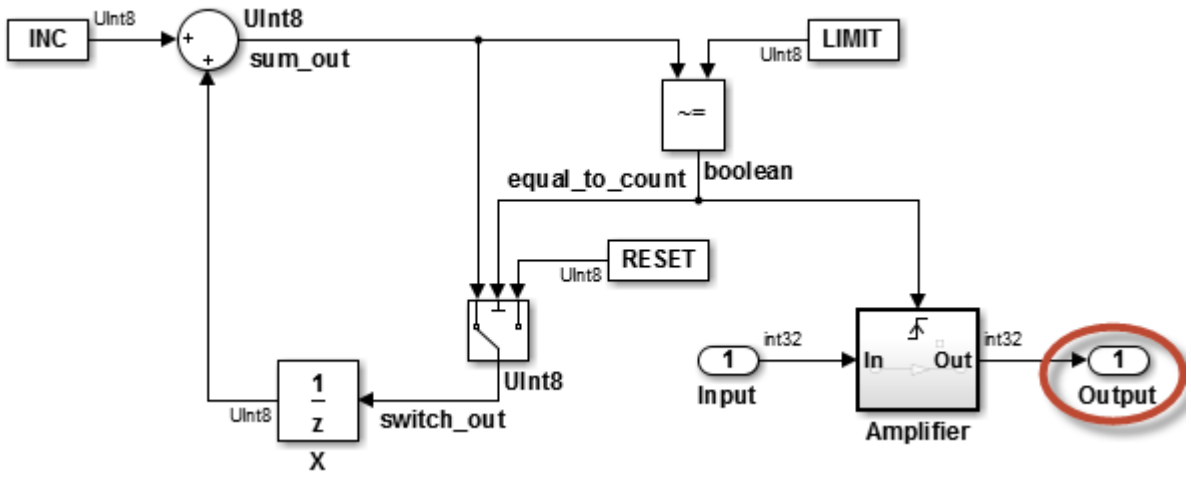
Mapping Inports to AUTOSAR Receiver Ports

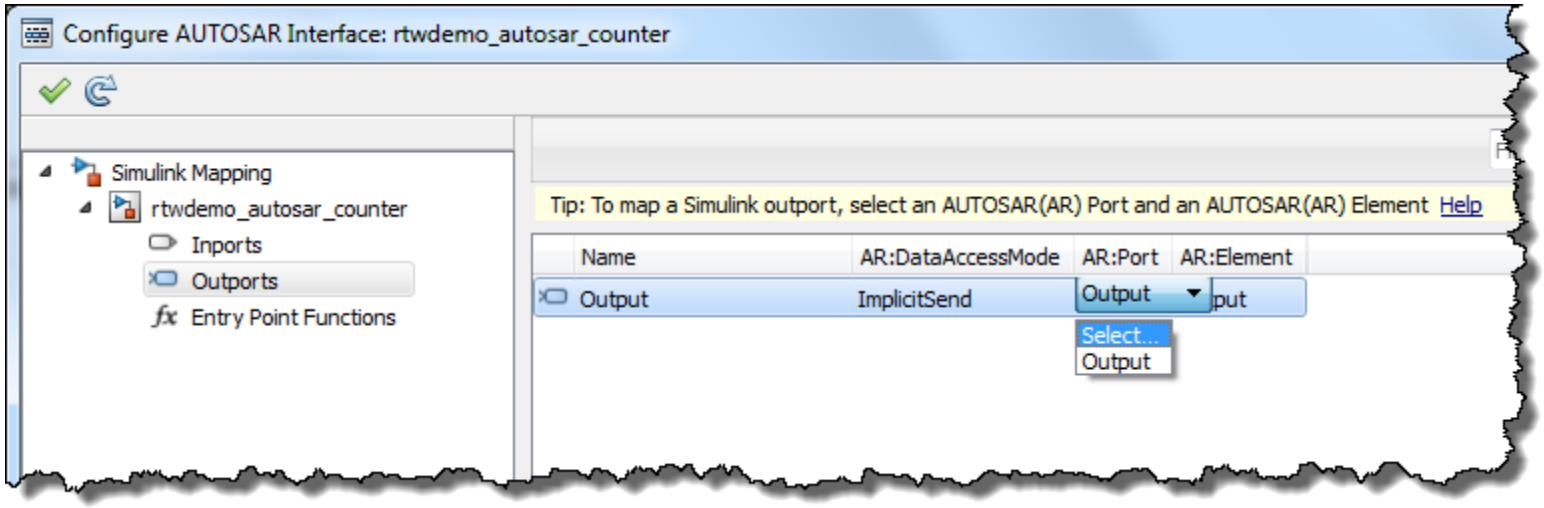




Here is an example of mapping a Simulink Inport to a AUTOSAR Port Data Element. In this case, you can see the Simulink Inport called Input is mapped to the AR:Port of Input. The actual selection of the AR:Element is a bit hidden in this view. Notice that the AR:DataAccessMode is also available for the user to select from this dialog.

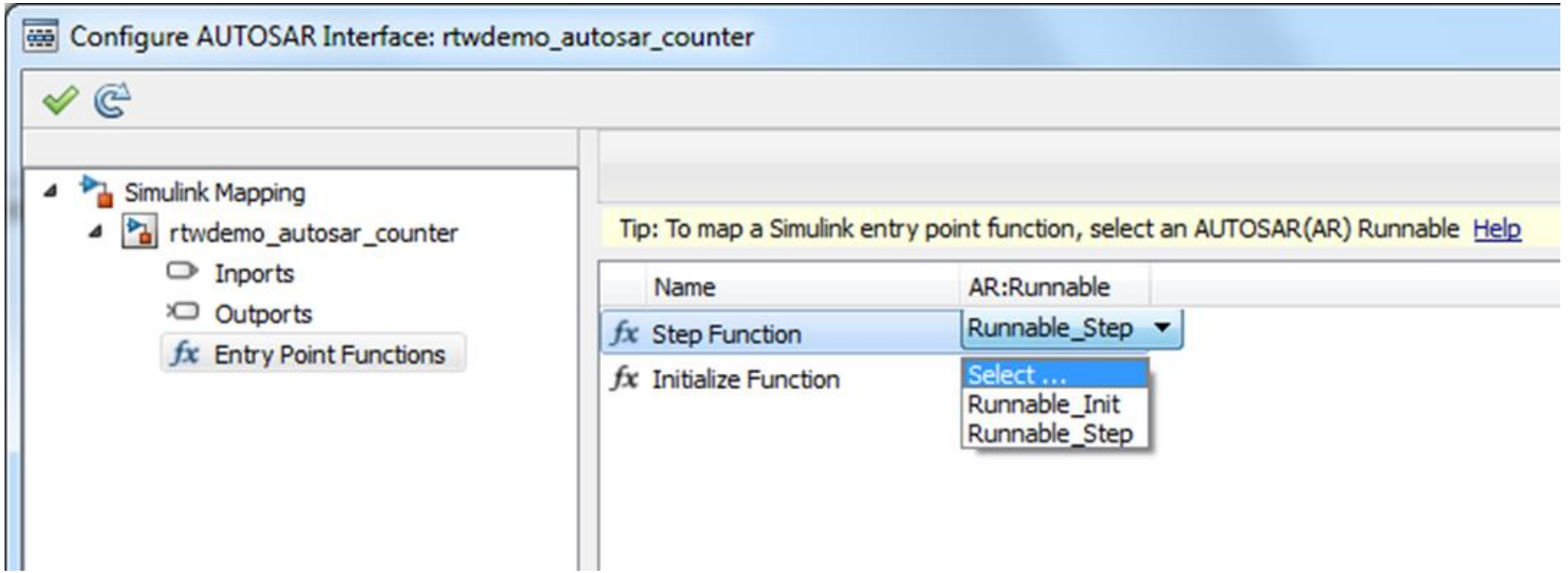
Map Outports to AUTOSAR Sender Ports





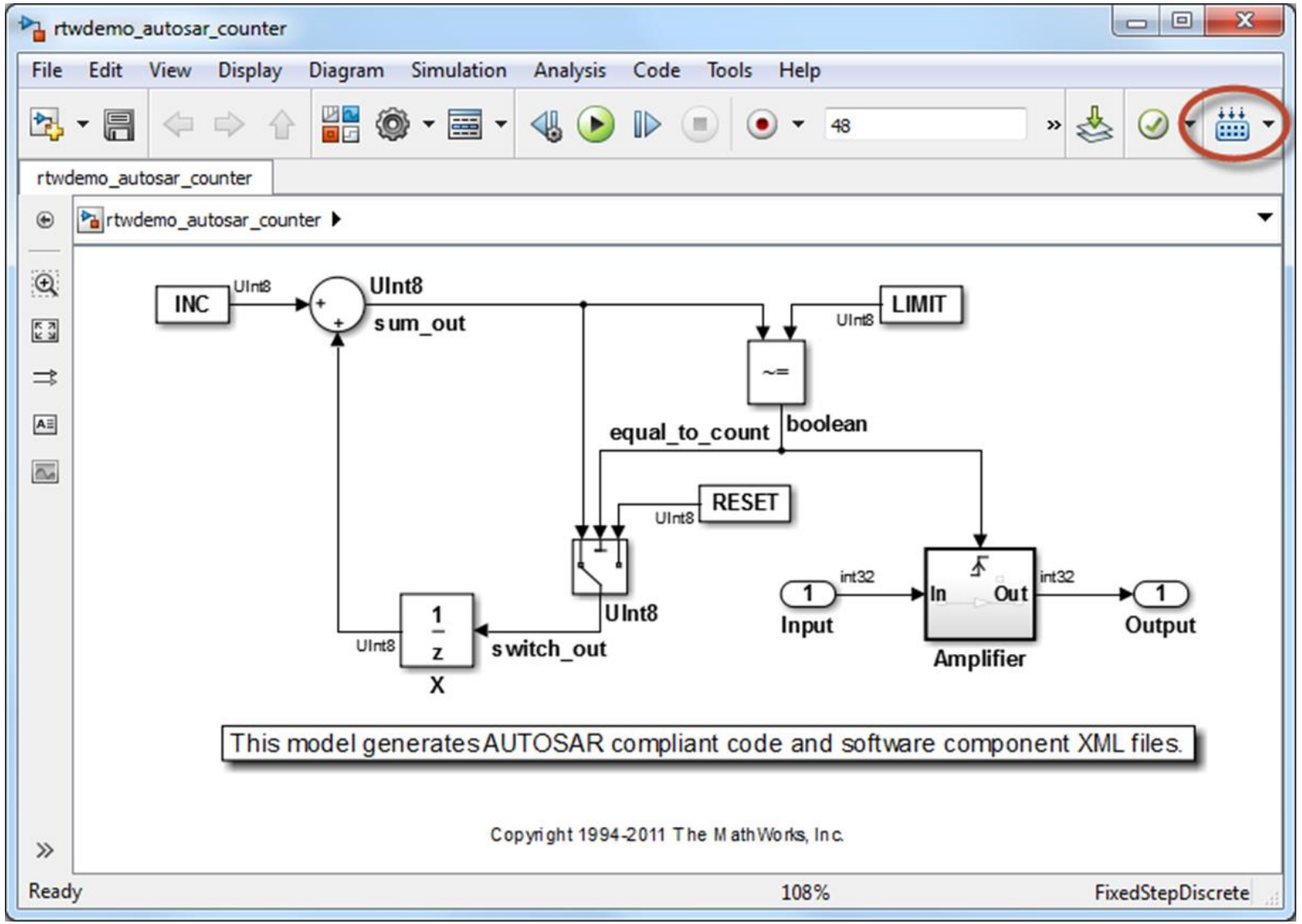
Here is an example of mapping a Simulink Inport to a AUTOSAR Port Data Element. In this case, you can see the Simulink Inport called Input is mapped to the AR:Port of Input. The actual selection of the AR:Element is a bit hidden in this view. Notice that the AR:DataAccessMode is also available for the user to select from this dialog.

Map Entry Point Functions for a Model



Here is an example of mapping a Simulink Function Entry Point to an AUTOSAR Runnable. Note: The Initialze Function most likely will get mapped to the Runnable_Step (In this example). This Initialize Function is for blocks within the Software Component that need initialization such as 1/z blocks. This is not the same as user initialization of signals or varaibles.

Generate Code

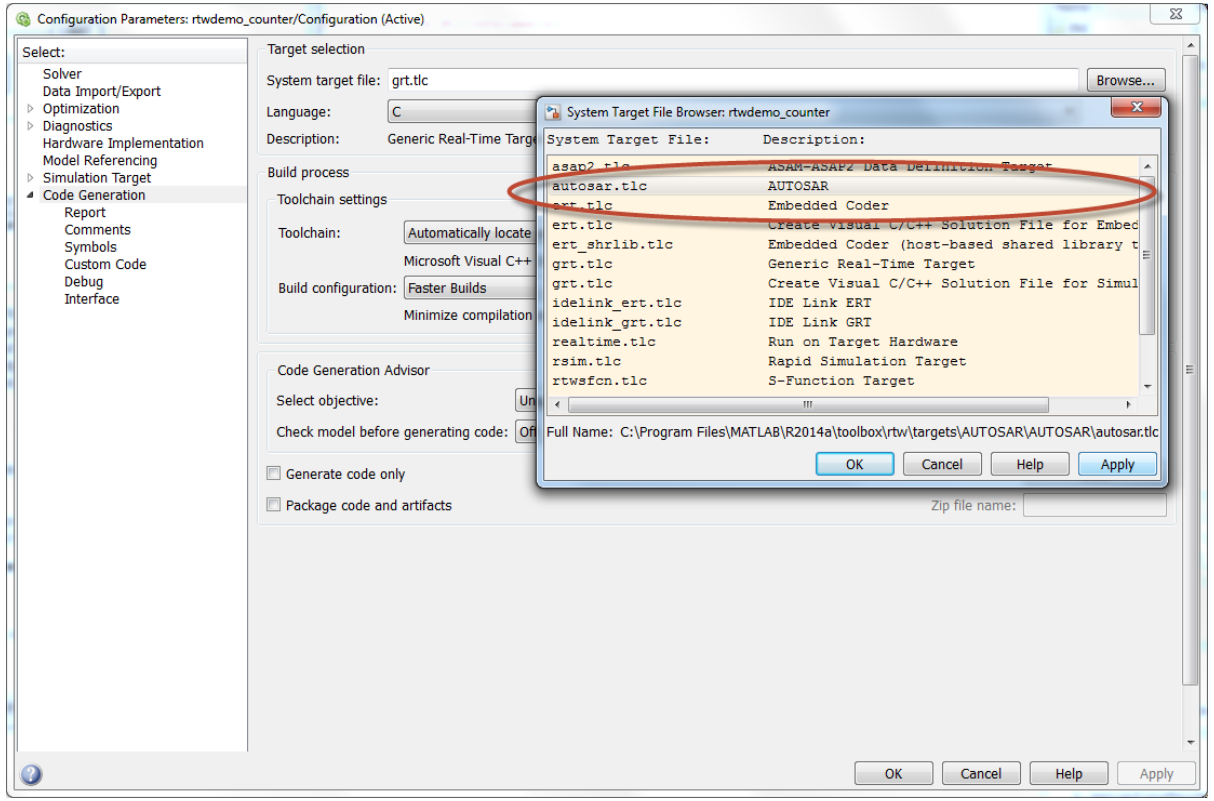


Build model into AUTOSAR compl aint Code (Control B or build Icon show to the left).

- Generates both C Code & AUT OSAR Software Component de scription files (ARXML files).
- Code uses RTE APIs to access AUTOSAR Ports such as Se nder Receiver ports or Client / Server Ports as needed.

Select AUTOSAR Target

- The mechanism for selecting an AUTOSAR target is similar to selecting an ERT target
  - via the Simulink Configuration Parameters menu, Code Generation Tab , System target file.

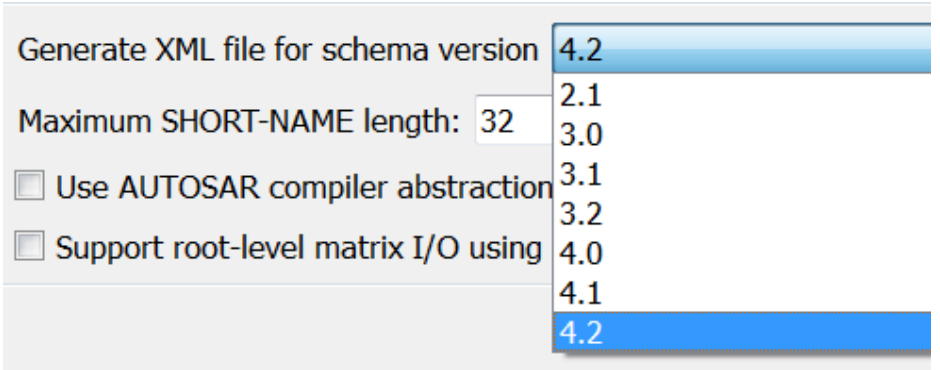


Overview of Generated ARXML

- Generated ARXML contains
  - Component and Internal Behavior
  - Datatypes
  - Implementation information
    - Lists all generated source code and ARXML files
  - Interfaces
  - Other entities

AUTOSAR Schema Versions

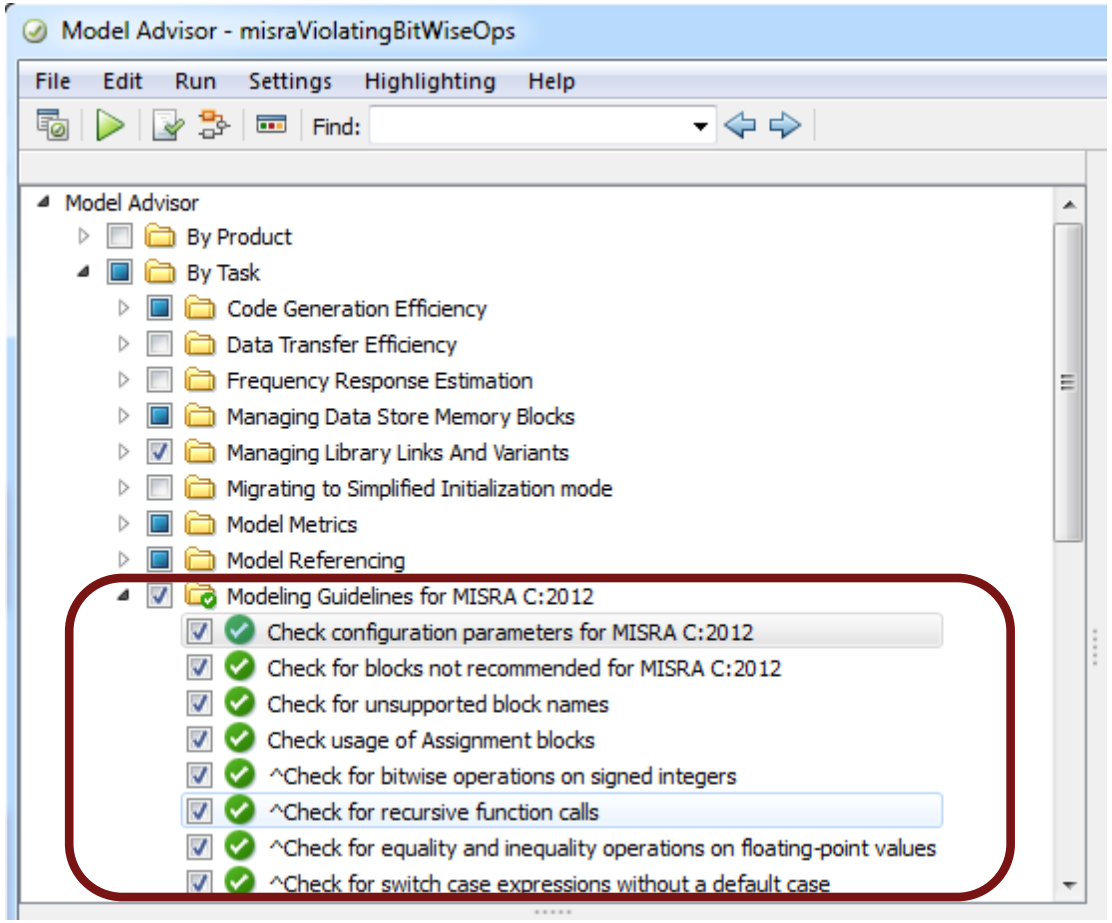
- Seamless support for AUTOSAR Releases
  - Import detects AUTOSAR 2.x – 4.x release from arxml file
  - User selects AUTOSAR release from configuration set options for code generation and arxml export

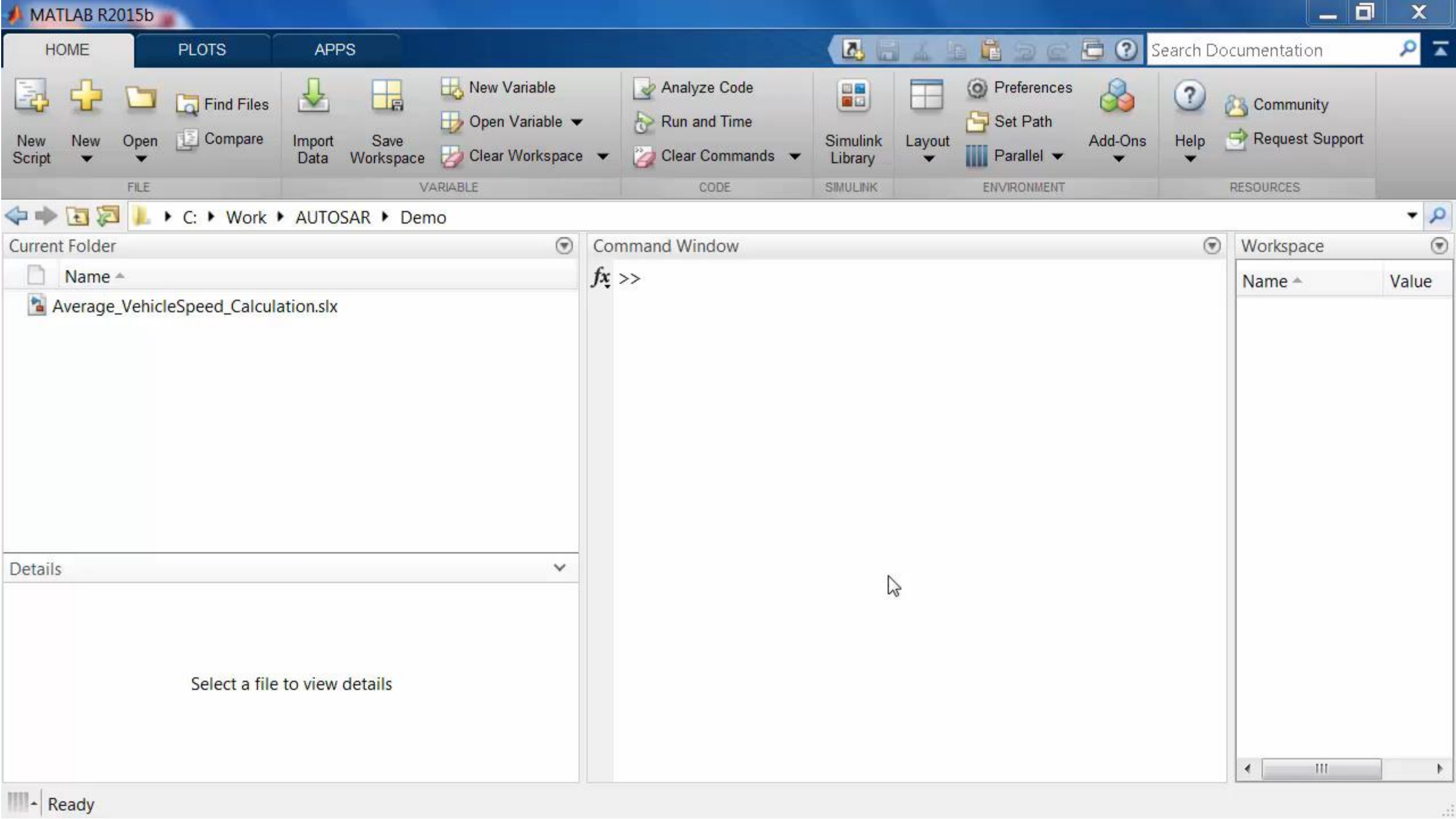


MATLAB Release	AUTOSAR Release
R2015b, R2016a/b, R2017a	2.1, 3.0, 3.1, **3.2** (Rev 3.2.2), 4.0, 4.1, **4.2** (Rev 4.2.1, 4.2.2)
R2014b, R2015a	2.1, 3.0, 3.1, 3.2, 4.0, **4.1** (Rev 4.1.1)
R2012a/b, R2013a/b, R2014a	2.1, 3.0, 3.1, 3.2, **4.0** (Rev 4.0.2)
R2011b	2.0, 2.1, 3.0, 3.1, 3.2
R2010a/b, R2011a	2.0, 2.1, 3.0, 3.1
R2009a/b	2.0, 2.1, 3.0
R2008a/b	2.0, 2.1

MISRA C:2012 for AUTOSAR target

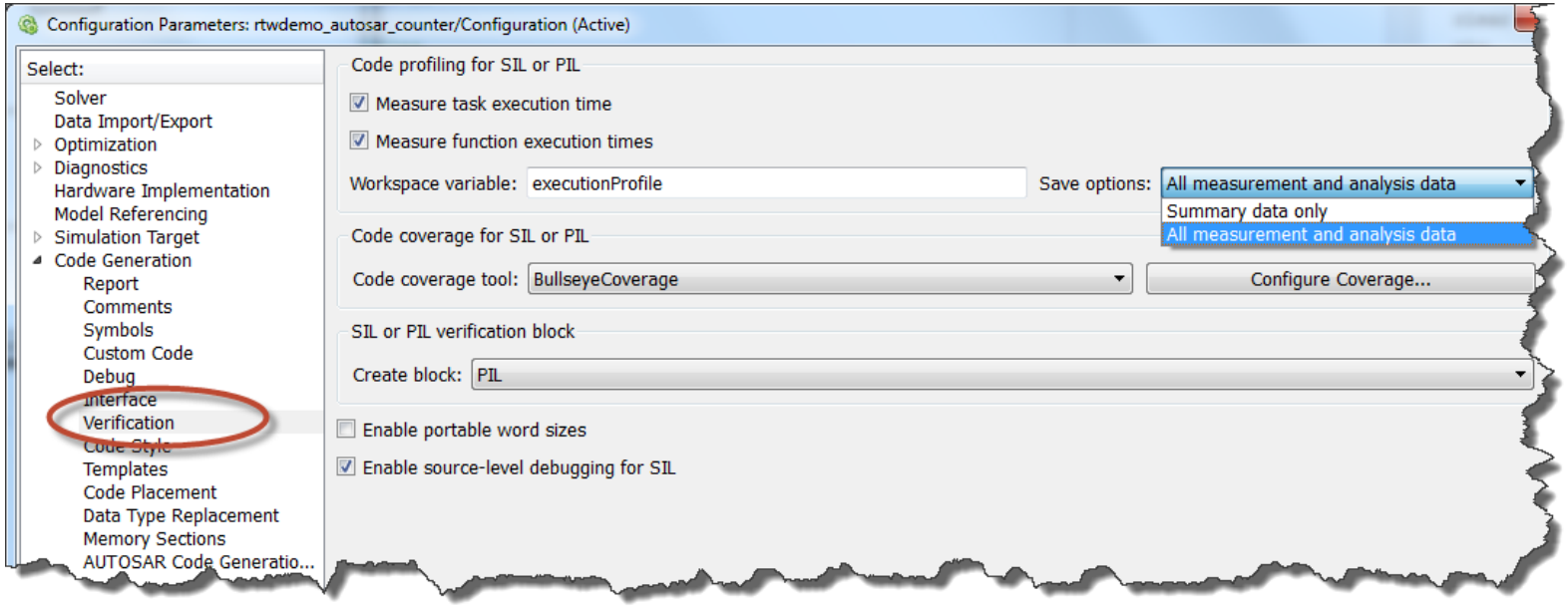
- 100% Compliance with MISRA C:2012 Mandatory and Required rules

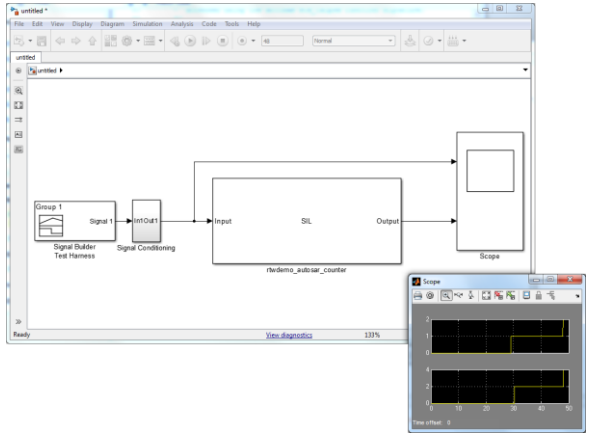



MATLAB R2015b
HOME
PLOTS
APPS
Search Documentation
New Script
New
Open
Find Files
Compare
Import Data
Save Workspace
New Variable
Open Variable
Clear Workspace
Analyze Code
Run and Time
Clear Commands
Simulink Library
Layout
Preferences
Set Path
Parallel
Add-Ons
Help
Community
Request Support
FILE
VARIABLE
CODE
SIMULINK
ENVIRONMENT
RESOURCES
C: ▸ Work ▸ AUTOSAR ▸ Demo
Current Folder
Name
Average_VehicleSpeed_Calculation.slx
Details
Select a file to view details
Command Window
>>
Workspace
Name
Value
Ready

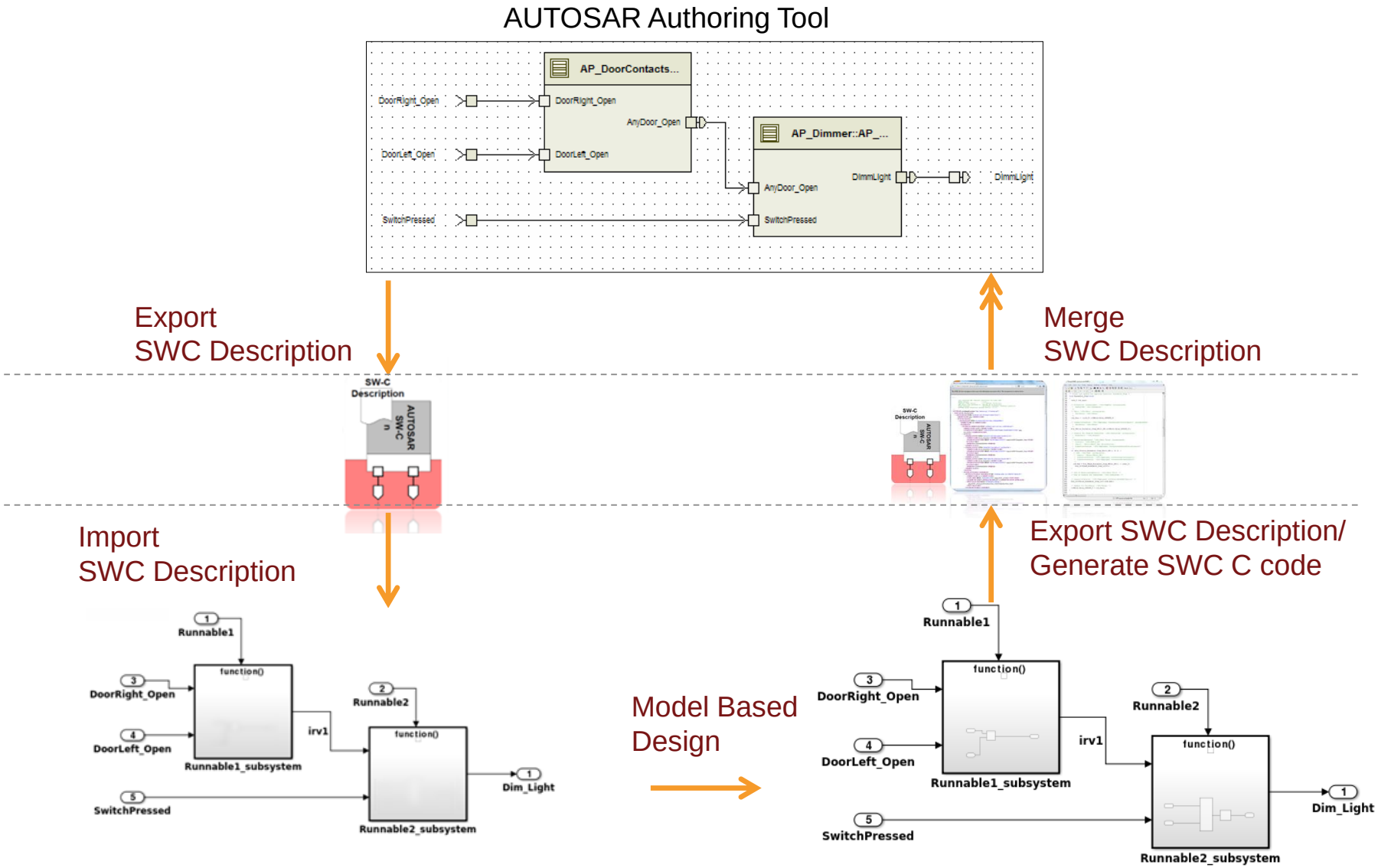
Verification with Software- and Processor-In-The-Loop (PIL)

- 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



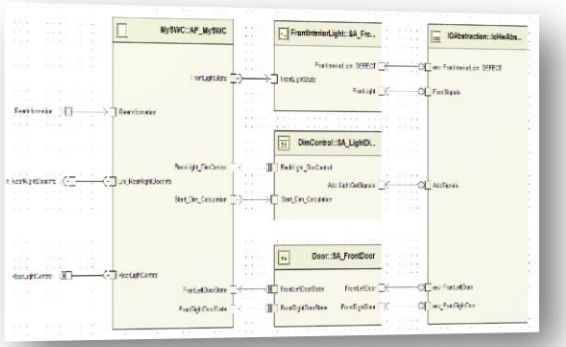


Top-Down Workflow (Starting from SWC Description)

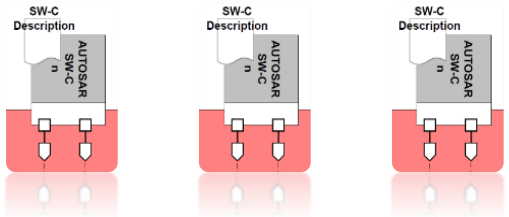


Top Down Workflow

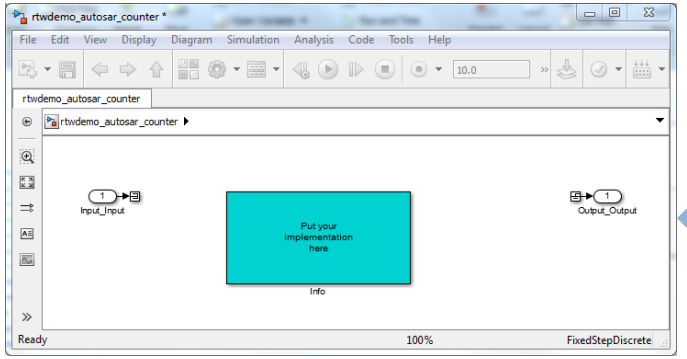
AUTOSAR Authoring Tool



ARXML FIles

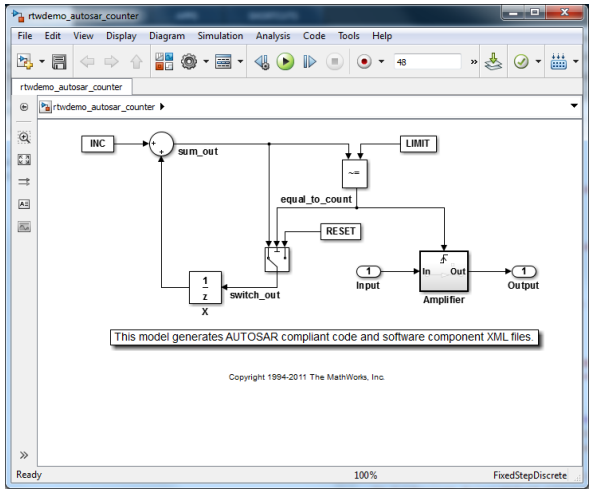


Import as new Simulink model



OR

Update existing Simulink model



Top Down Workflow

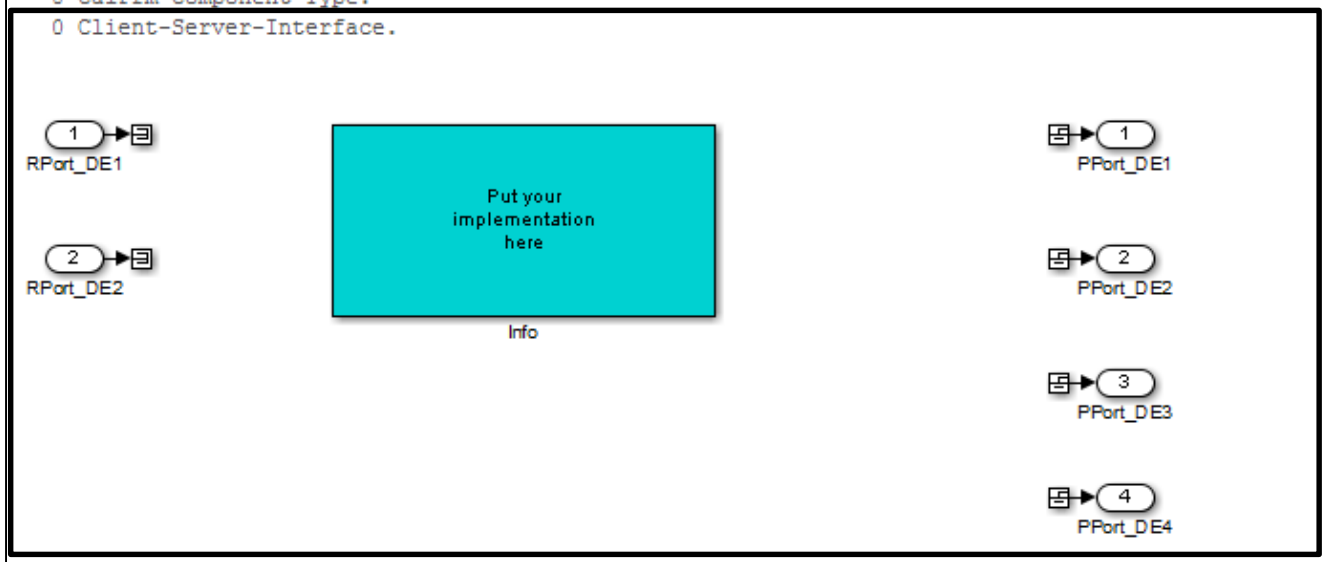
Starts with Authoring Tool, then user exports ARXML files from Authoring tool.

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

Importing ARXML Files

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

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

```
importerObj =

The file "C:\Backup\General_Work_PSPs\Documents\MAC US 2014\rtwdemo_autosar_multirunnables.arxml" contains:
  1 Application-Software-Component-Type:
    '/pkg/swc/ASWC'

  0 Sensor-Actuator-Software-Component-Type.
  0 CalPrm-Component-Type.
  0 Client-Server-Interface.
```



Top Down Workflow

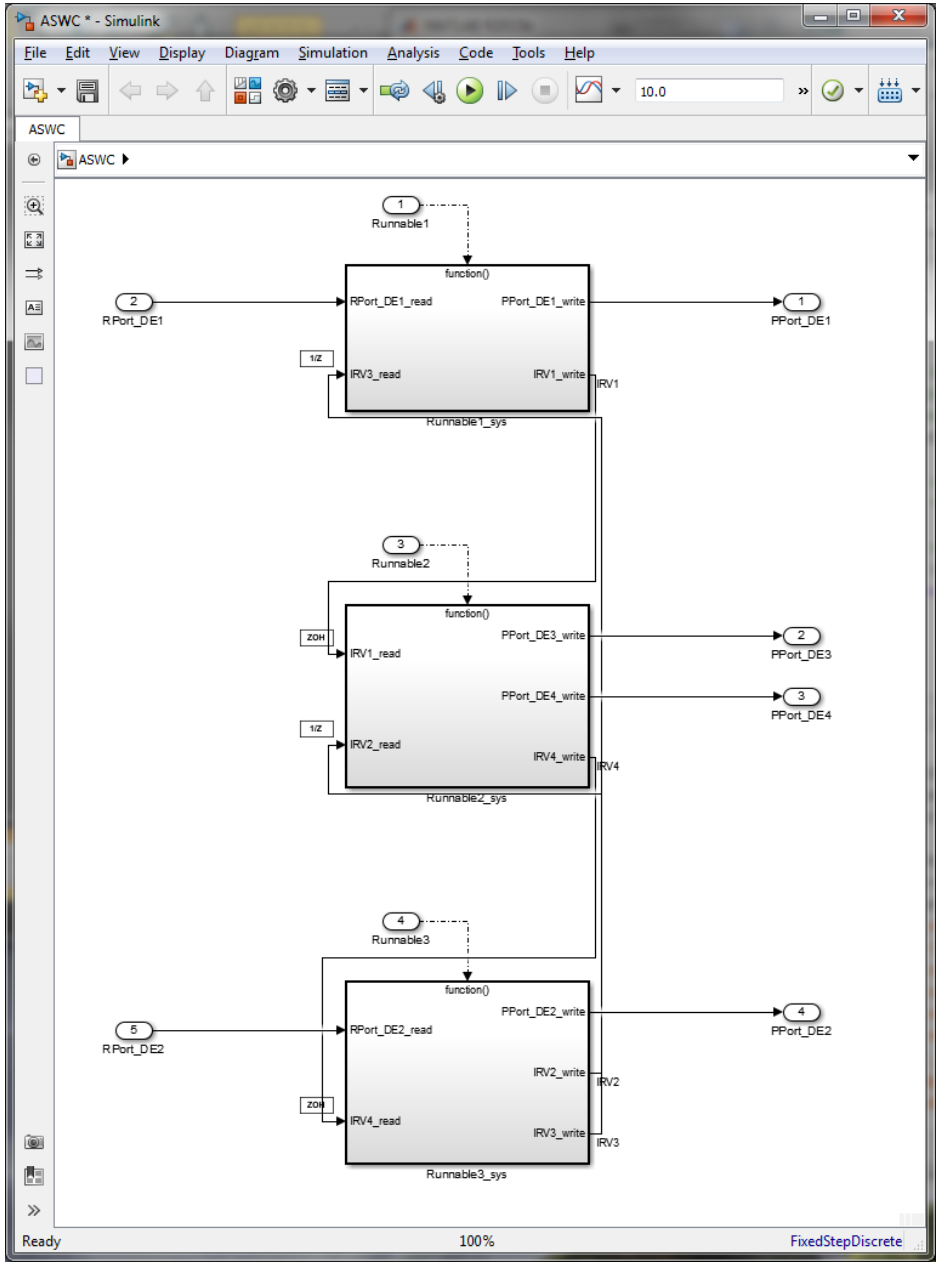
Commands to create a skeleton model with no internal behavior setup in the skeleton model (no runnables)

Note: After running the arxml.importer command, here is the package that is needed for the second command of create Component as Model

Import with Internal Behavior

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

%Import ARXML Files
%      After Creating the Importer Object, you will observe
%      what components are available.
importerObj = arxml.importer('rtwdemo_autosar_multirunnables.arxml') %#ok<NOPTS>

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

%Create new model with interfaces and internal behavior
model = importerObj.createComponentAsModel('/pkg/swc/ASWC', ...
    'CreateInternalBehavior',true);
```

```
>> importExample

importerObj =

The file "C:\00_mdInfo\AUTOSAR\AUTOSAR_Seminar\Demo2_TopDown\rtwdemo_autosar_multirunnables.arxml" contains:
  1 Application-Software-Component-Type:
    '/pkg/swc/ASWC'

  0 Sensor-Actuator-Software-Component-Type.
  0 Parameter-Software-Component-Type.
  0 Client-Server-Interface.
```



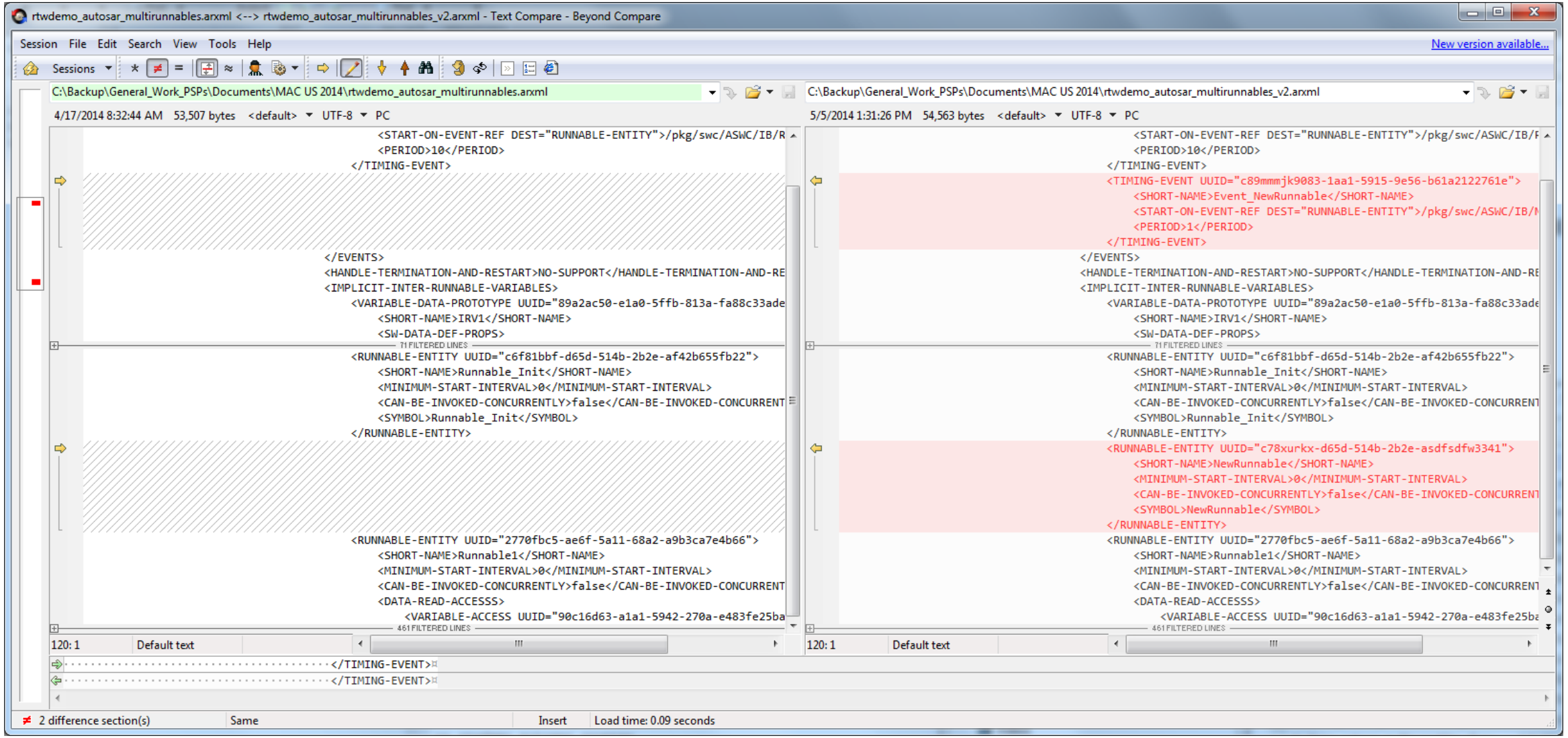
Top Down Workflow

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



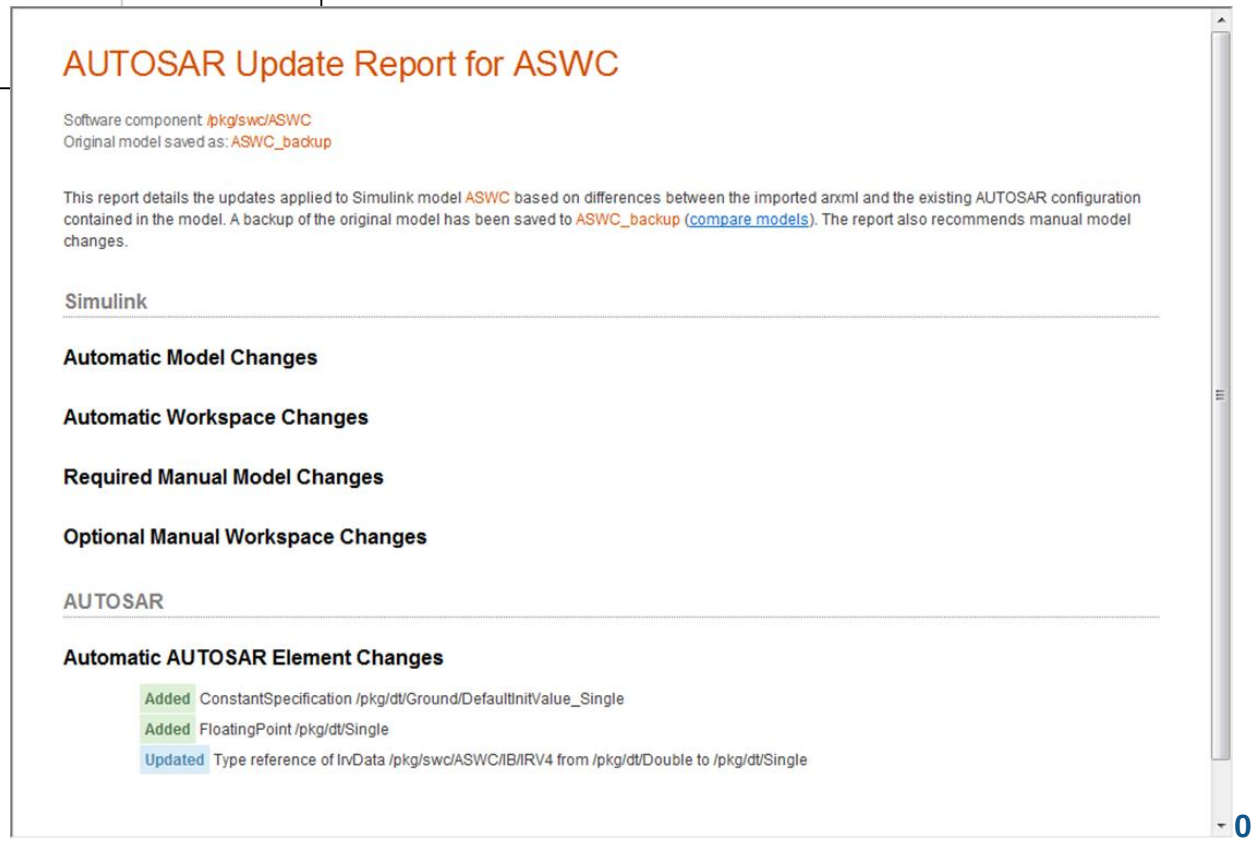
Update Existing Models from ARXML

```
1   %cleanup
2   bdclose('all');
3   clear;
4
5   open_system('ASWC'); % Model needs to be open in order to perform update Model Command
6
7   %Import ARXML Files
8   importerObj = arxml.importer('rtwdemo_autosar_multirunnables_v2.arxml')
9
10  %Update existing model
11  importerObj.updateModel('ASWC')
```

Top Down Workflow

Commands to update an existing Simulink model that already has an AUTOSAR configuration.

Notice that a report was created such that the user can understand what has changed in the model.



AUTOSAR Update Report for ASWC

Software component /pkg/swc/ASWC
Original model saved as: ASWC_backup

This report details the updates applied to Simulink model ASWC based on differences between the imported arxml and the existing AUTOSAR configuration contained in the model. A backup of the original model has been saved to ASWC_backup (compare models). The report also recommends manual model changes.

Simulink

Automatic Model Changes

Automatic Workspace Changes

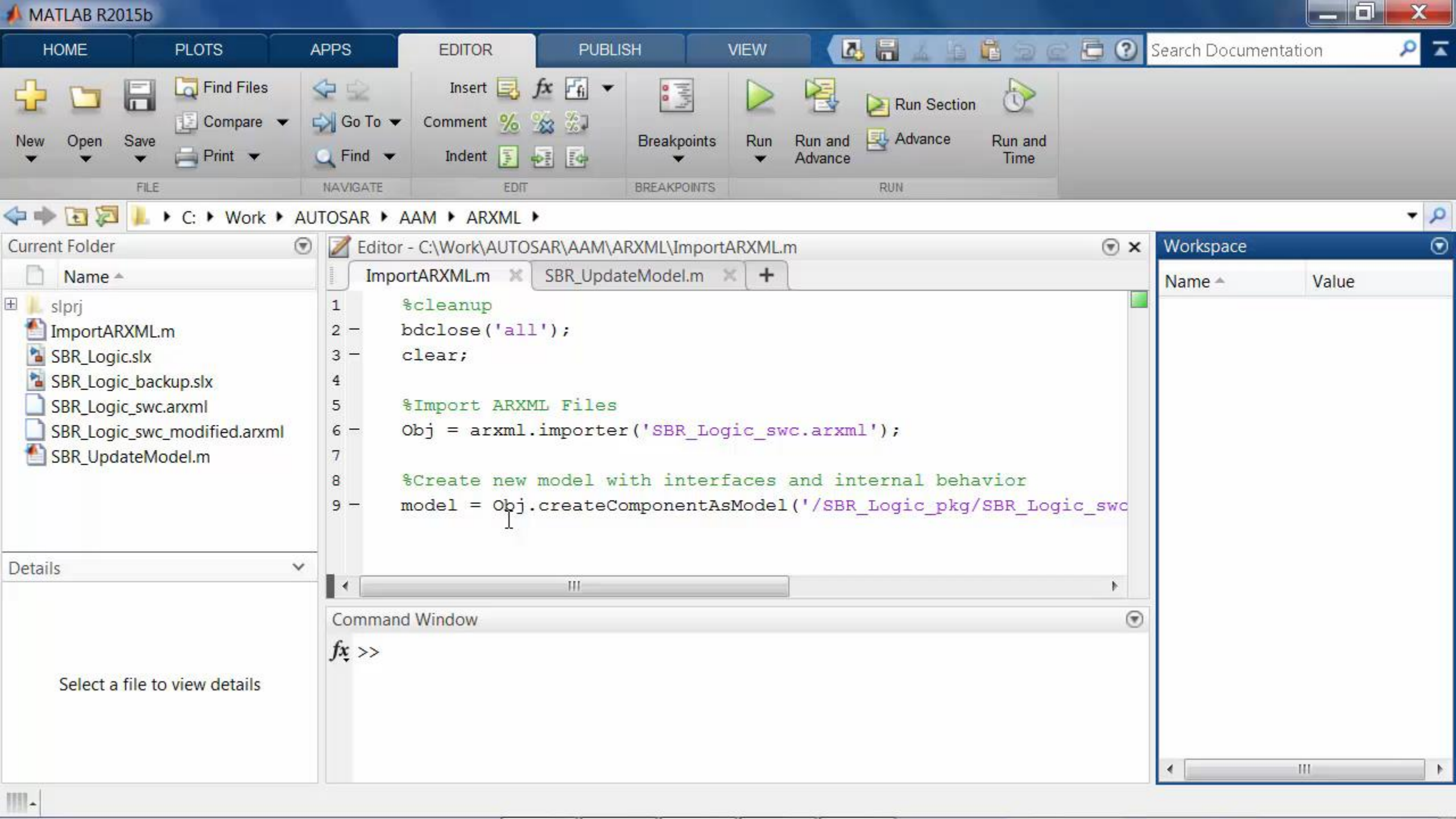
Required Manual Model Changes

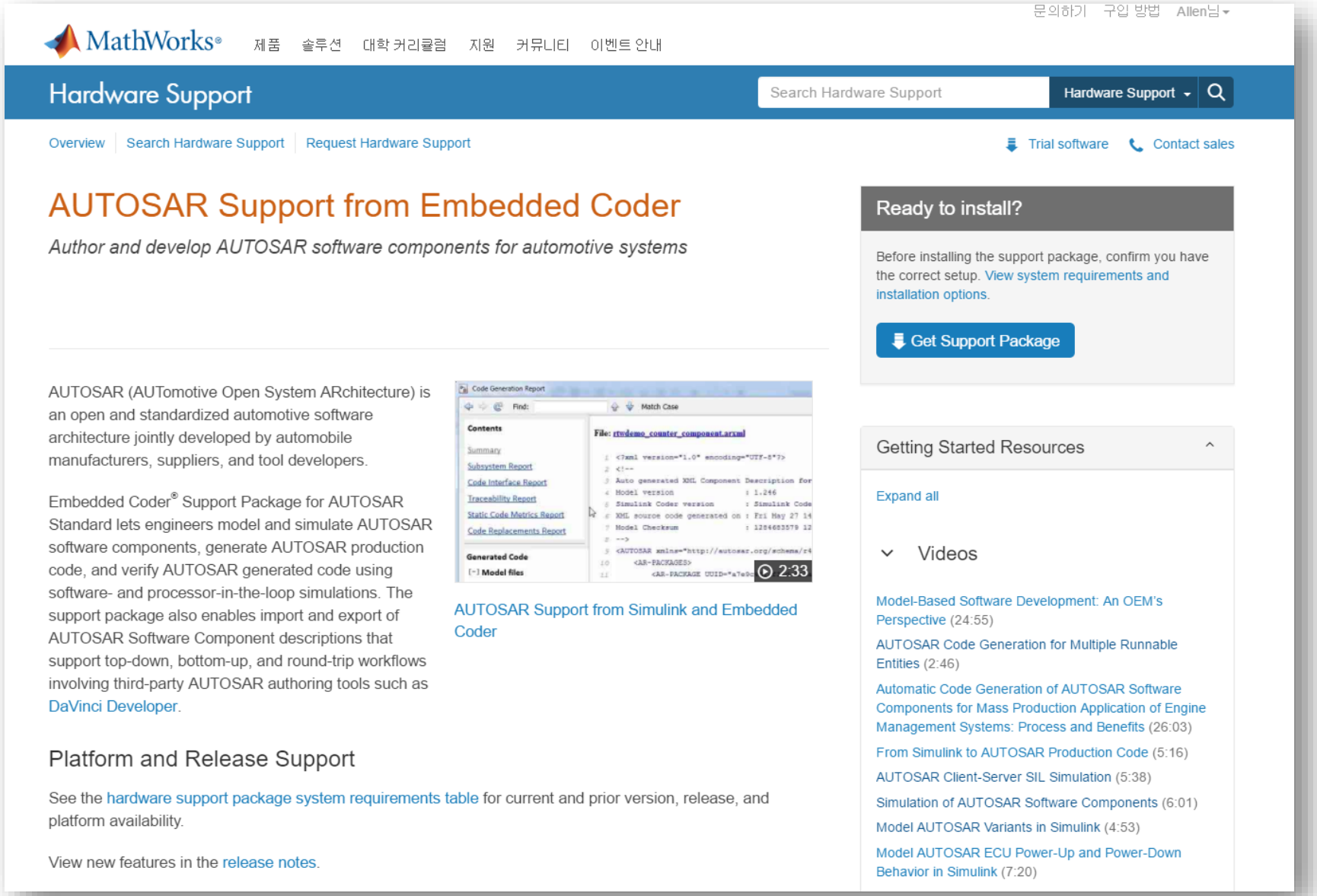
Optional Manual Workspace Changes

AUTOSAR

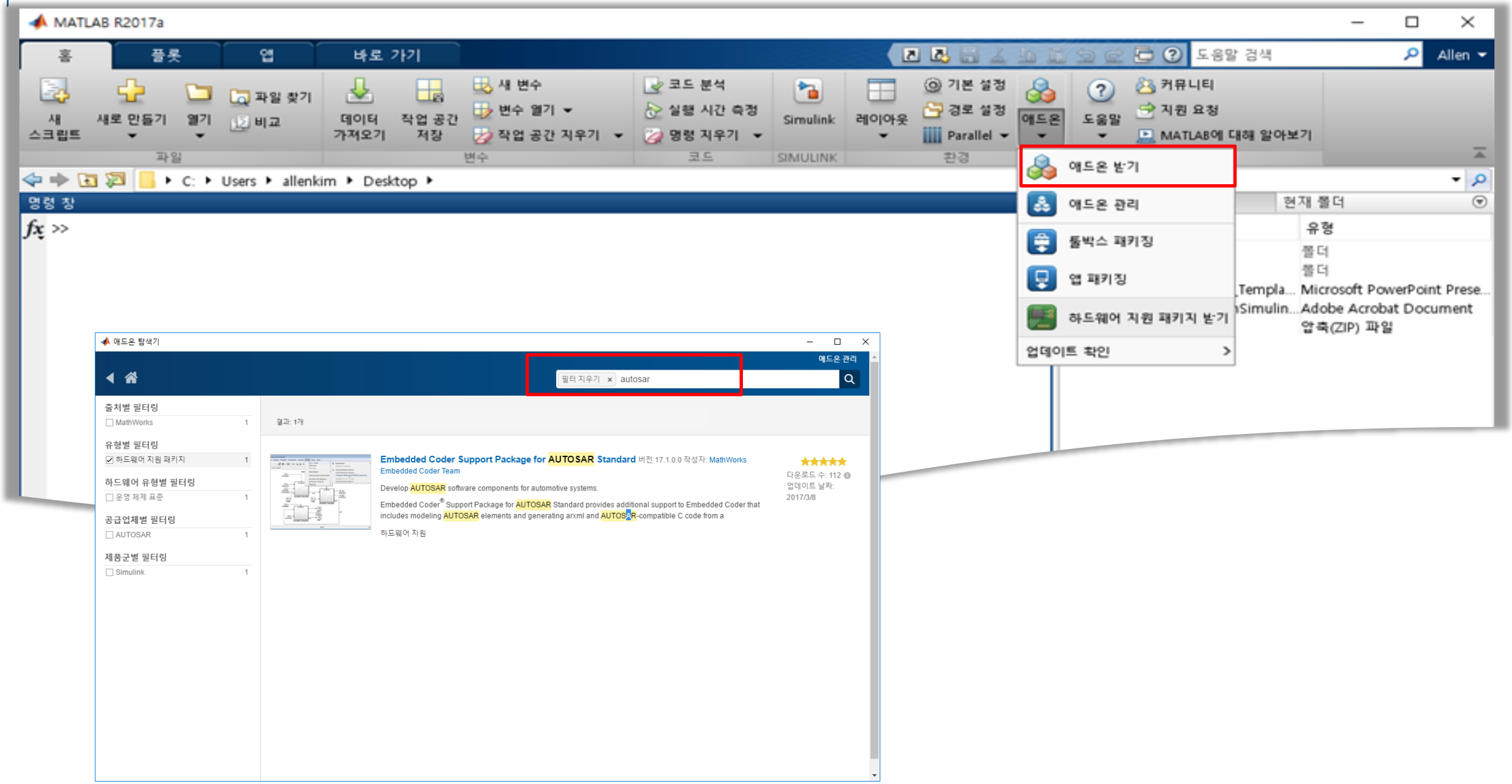
Automatic AUTOSAR Element Changes

Added ConstantSpecification /pkg/dt/Ground/DefaultInitValue_Single
Added FloatingPoint /pkg/dt/Single
Updated Type reference of IrvData /pkg/swc/ASWC/IB/IRV4 from /pkg/dt/Double to /pkg/dt/Single


MATLAB R2015b
HOME
PLOTS
APPS
EDITOR
PUBLISH
VIEW
Search Documentation
New
Open
Save
Find Files
Compare
Print
FILE
Go To
Find
NAVIGATE
Insert
Comment
Indent
EDIT
Breakpoints
BREAKPOINTS
Run
Run and Advance
Run Section
Advance
Run and Time
RUN
C: ▸ Work ▸ AUTOSAR ▸ AAM ▸ ARXML ▸
Current Folder
Name
slprj
ImportARXML.m
SBR_Logic.slx
SBR_Logic_backup.slx
SBR_Logic_swc.arxml
SBR_Logic_swc_modified.arxml
SBR_UpdateModel.m
Details
Select a file to view details
Editor - C:\Work\AUTOSAR\AAM\ARXML\ImportARXML.m
ImportARXML.m
SBR_UpdateModel.m
1 %cleanup
2 bdclose('all');
3 clear;
4
5 %Import ARXML Files
6 Obj = arxml.importer('SBR_Logic_swc.arxml');
7
8 %Create new model with interfaces and internal behavior
9 model = Obj.createComponentAsModel('/SBR_Logic_pkg/SBR_Logic_swc
Command Window
>>
Workspace
Name
Value



http://www.mathworks.com/hardware-support/autosar.html


MATLAB R2017a
홈
플롯
앱
바로 가기
도움말 검색
Allen
새 스크립트
새로 만들기
열기
파일 찾기
비교
데이터 가져오기
작업 공간 저장
새 변수
변수 열기
작업 공간 지우기
코드 분석
실행 시간 측정
명령 창 지우기
Simulink
레이아웃
기본 설정
경로 설정
Parallel
애드온
도움말
커뮤니티
지원 요청
MATLAB에 대해 알아보기
파일
변수
코드
SIMULINK
환경
C: ▸ Users ▸ allenkim ▸ Desktop ▸
명령 창
fx >>
애드온 받기
애드온 관리
툴박스 패키징
앱 패키징
하드웨어 지원 패키지 받기
업데이트 확인
현재 폴더
유형
폴더
폴더
Templa... Microsoft PowerPoint Prese...
Simulin... Adobe Acrobat Document
압축(ZIP) 파일
애드온 탐색기
애드온 관리
필터 지우기
autosar
출처별 필터링
MathWorks 1
유형별 필터링
하드웨어 지원 패키지 1
하드웨어 유형별 필터링
운영 체제 표준 1
공급업체별 필터링
AUTOSAR 1
제품군별 필터링
Simulink 1
결과: 1개
Embedded Coder Support Package for AUTOSAR Standard 버전 17.1.0.0 작성자: 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
하드웨어 지원
다운로드 수: 112
업데이트 날짜: 2017/3/8

Thank You!