

MATLAB EXPO

2021

Design and deploy Service-oriented Architectures (SOA) in Simulink

Shwetha Bhadravathi Patil



Aastha Kanwar



Mark Danielsen



Luigi Milia



Agenda

- Evolution of architectures and key challenges
- Simulink for service-oriented architectures (SOA)
- Simulink support for SOA industry standards
 - Simulink for AUTOSAR Adaptive
 - Simulink for Data Distribution Service (DDS)
- Conclusions and key takeaways

The rush for ~~Gold~~ Software

From the news...

- **General Motors** Adds 3,000 New Tech Jobs to Boost Virtual Testing and Software Expertise
- **Volkswagen's** Boss Warns the Troops: We Don't Want to End Up Like Nokia
- **Daimler** to Cut Out Suppliers to Fund Software Hiring Spree
- **Continental and Amazon Web Services** Create Platform for Automotive Software
- **Bosch** Consolidates All Automotive Software And Electronics Into New Division

<https://media.gm.com/media/us/en/gm/home.detail.html/content/Pages/news/us/en/2020/nov/1109-techjobs.html>
<https://www.bloomberg.com/news/articles/2020-11-06/vw-s-boss-warns-the-troops-we-don-t-want-to-end-up-like-nokia>
<https://www.ft.com/content/6173af2c-2ea8-4e90-876a-5cc189e3342b>
<https://www.continental.com/en/press/press-releases/20210415-continental-and-amazon-web-services-251210>
<https://www.forbes.com/sites/samabuelsamid/2020/07/21/bosch-consolidates-all-automotive-software-and-electronics-into-new-division/?sh=46032fb81320>

Industry trends

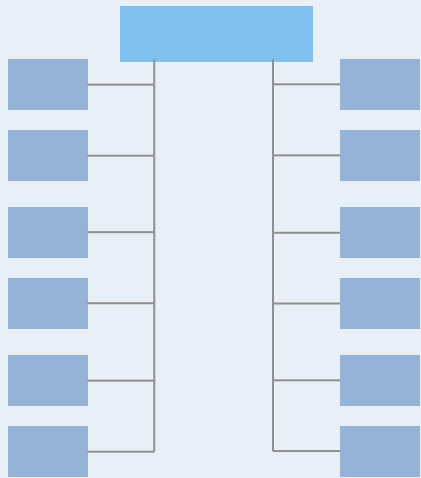


Software Defined Vehicle

Key enablers:

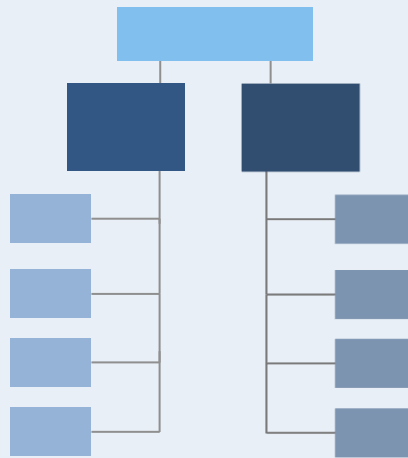
1. Centralization of computing power into high-performance, on-board computers
2. New approach to E/E and software architectures

Evolution of E/E architectures



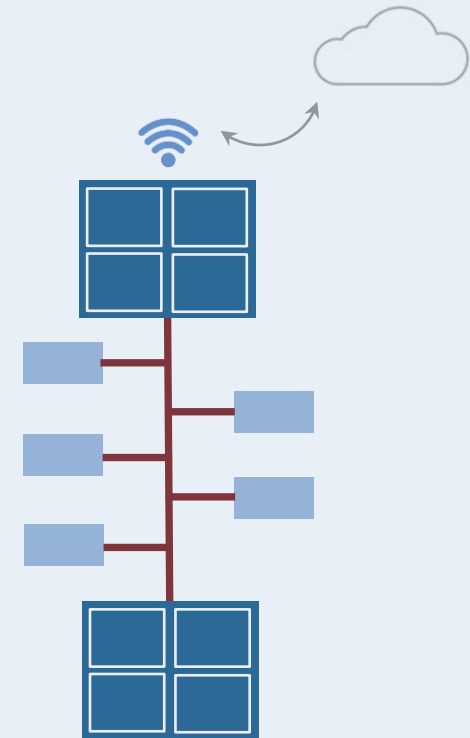
- ~100 ECUs
- One function per ECU
- Heavy and long wiring

Distributed



- Domain controllers
- Multi-core ECUs
- Gateways and networks

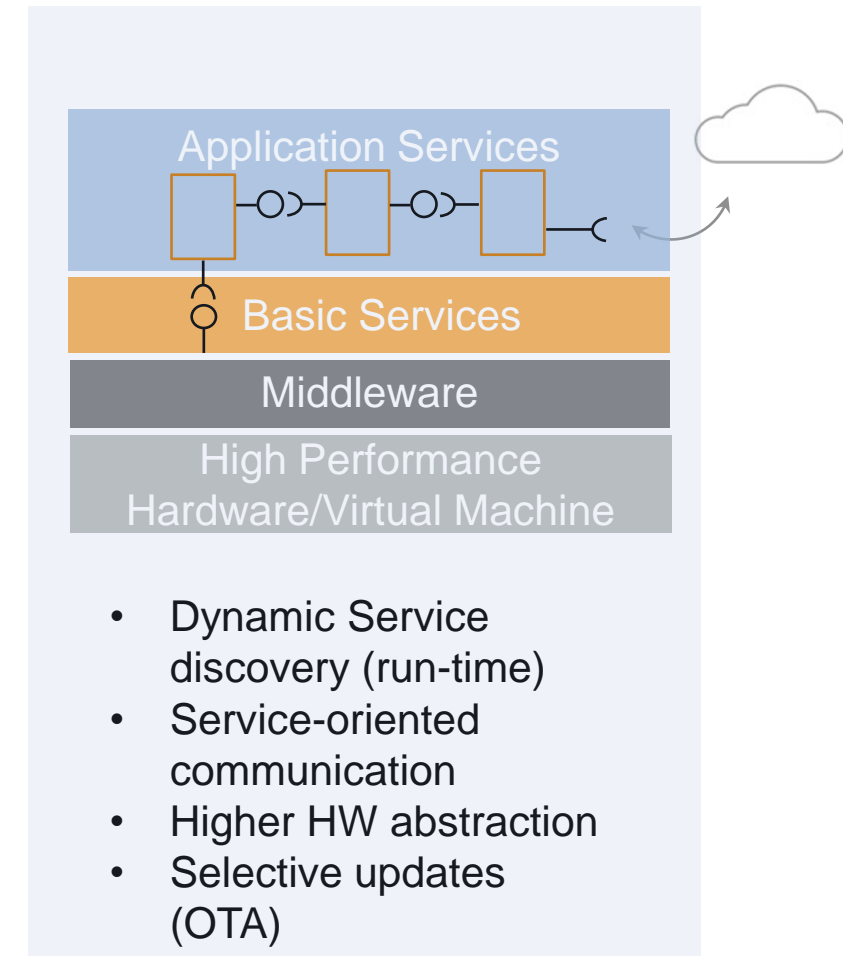
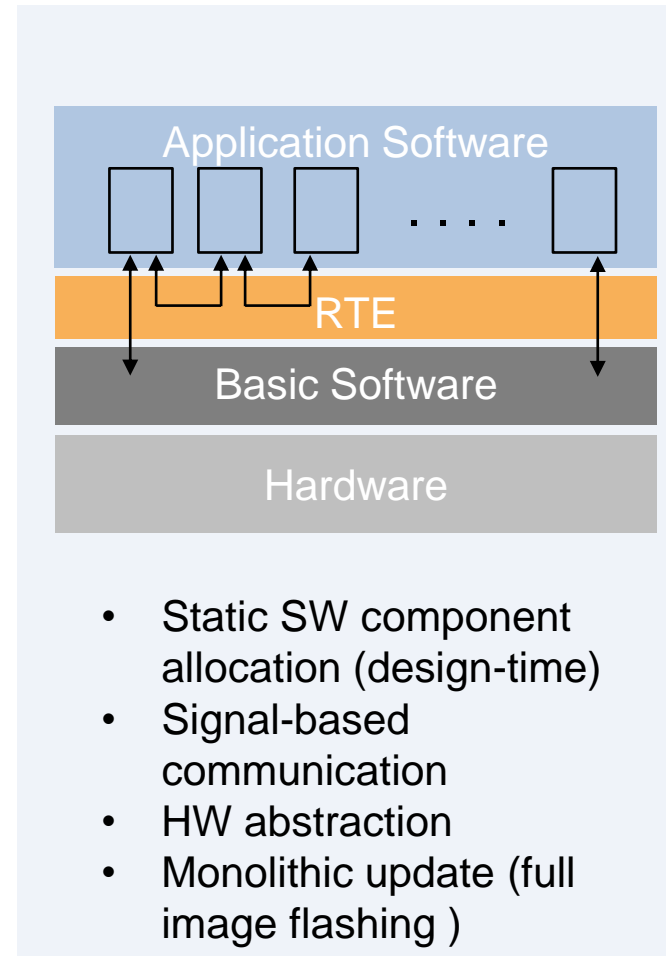
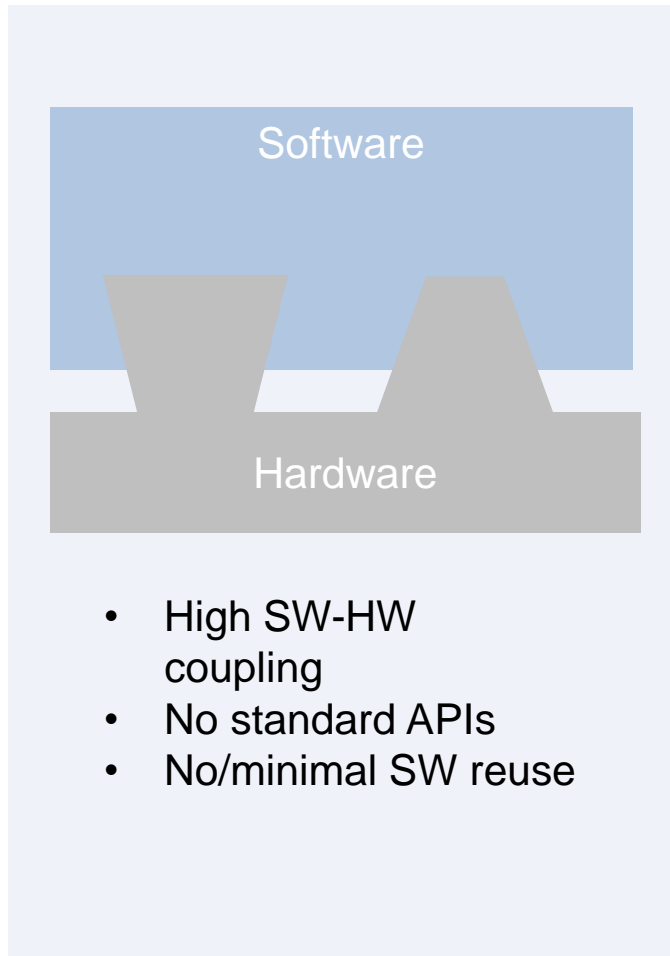
Consolidation



- Vehicle/zone controllers
- High-performance CPUs
(Many-core, GPU, FPGA)
- High-speed ethernet

Performance-Connectivity

Towards service-oriented architectures



Legacy

Component-based

Service-oriented

SOA – What's it all about?

- SOA consists of services that communicate across different platforms over messages.
- SOA provides flexibility to add, remove, or update components without impacting the entire, typically large, software system
- SOA is used by multiple industrial standards including:
 - AUTOSAR Adaptive Platform
 - DDS (Data Distribution Services)
 - ROS (Robot Operating System)

AUTOSAR Blockset

Design and simulate AUTOSAR software

DDS Blockset

Design and simulate DDS applications

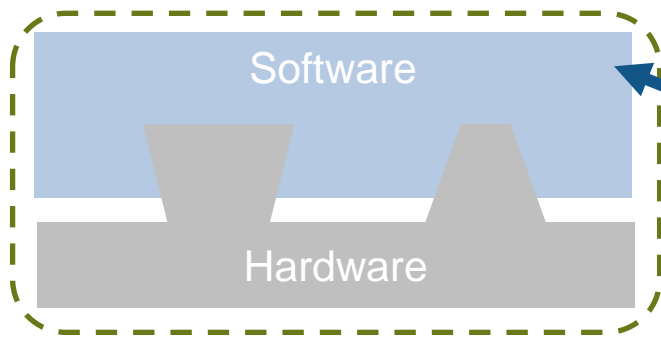
ROS Toolbox

Design, simulate, and deploy ROS-based applications

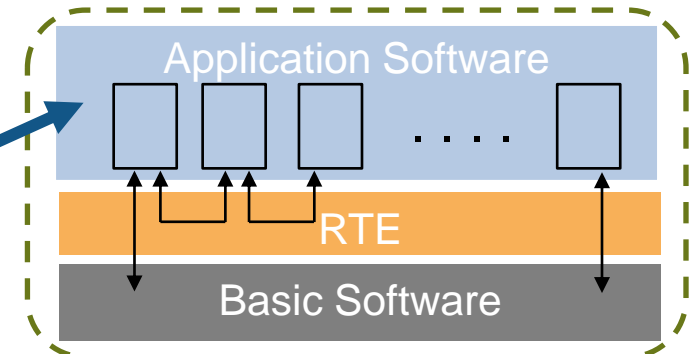
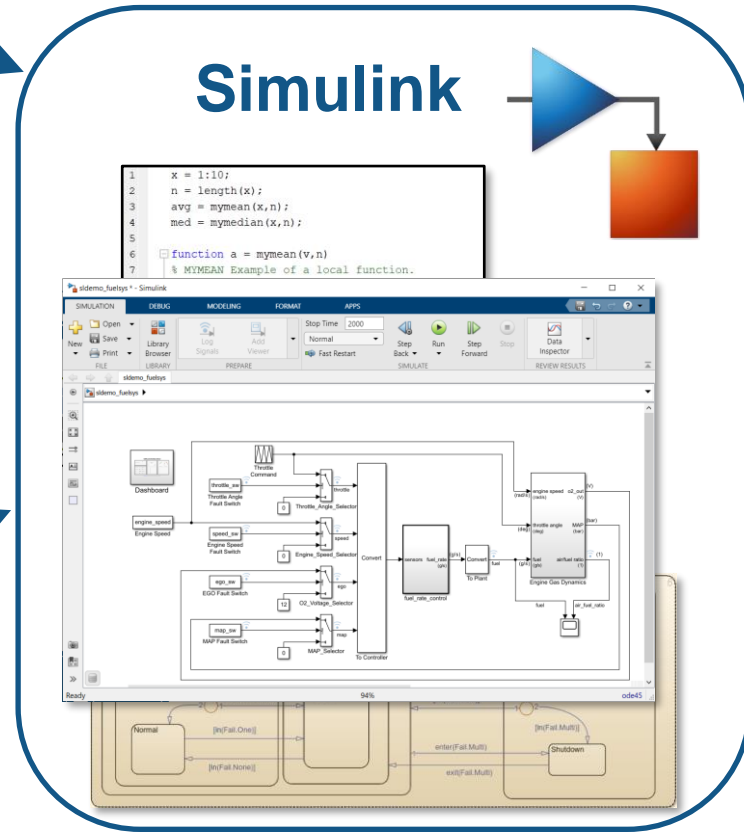
Key challenges

- Service-oriented applications are still new for Automotive industry
- Coexistence of legacy and SOA architectures
- Reuse of existing expertise, workflows and software assets (don't start from scratch)

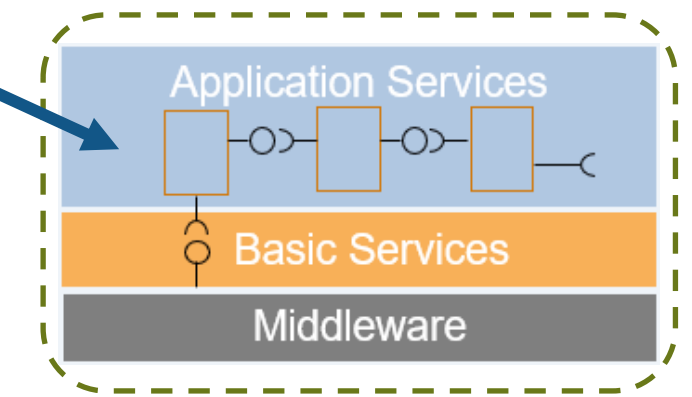
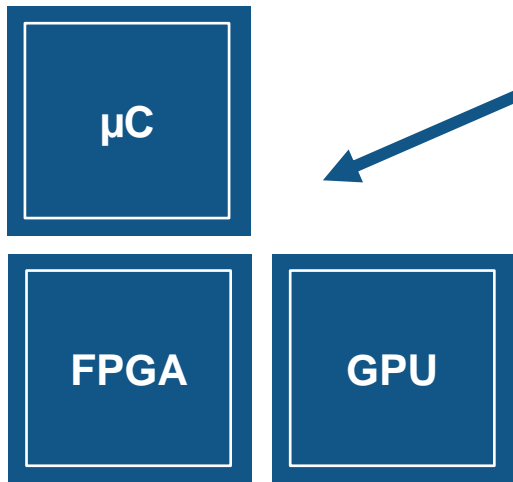
Simulink : Deploy software to different targets and standards



Legacy ECU



AUTOSAR Classic



AUTOSAR Adaptive / ROS / DDS

Poll Question #1

Are you already working on SOA based applications? (Check all that apply)

Yes, I'm working on a production or pilot project

Yes, I plan to reuse existing Simulink models / use MBD for SOA

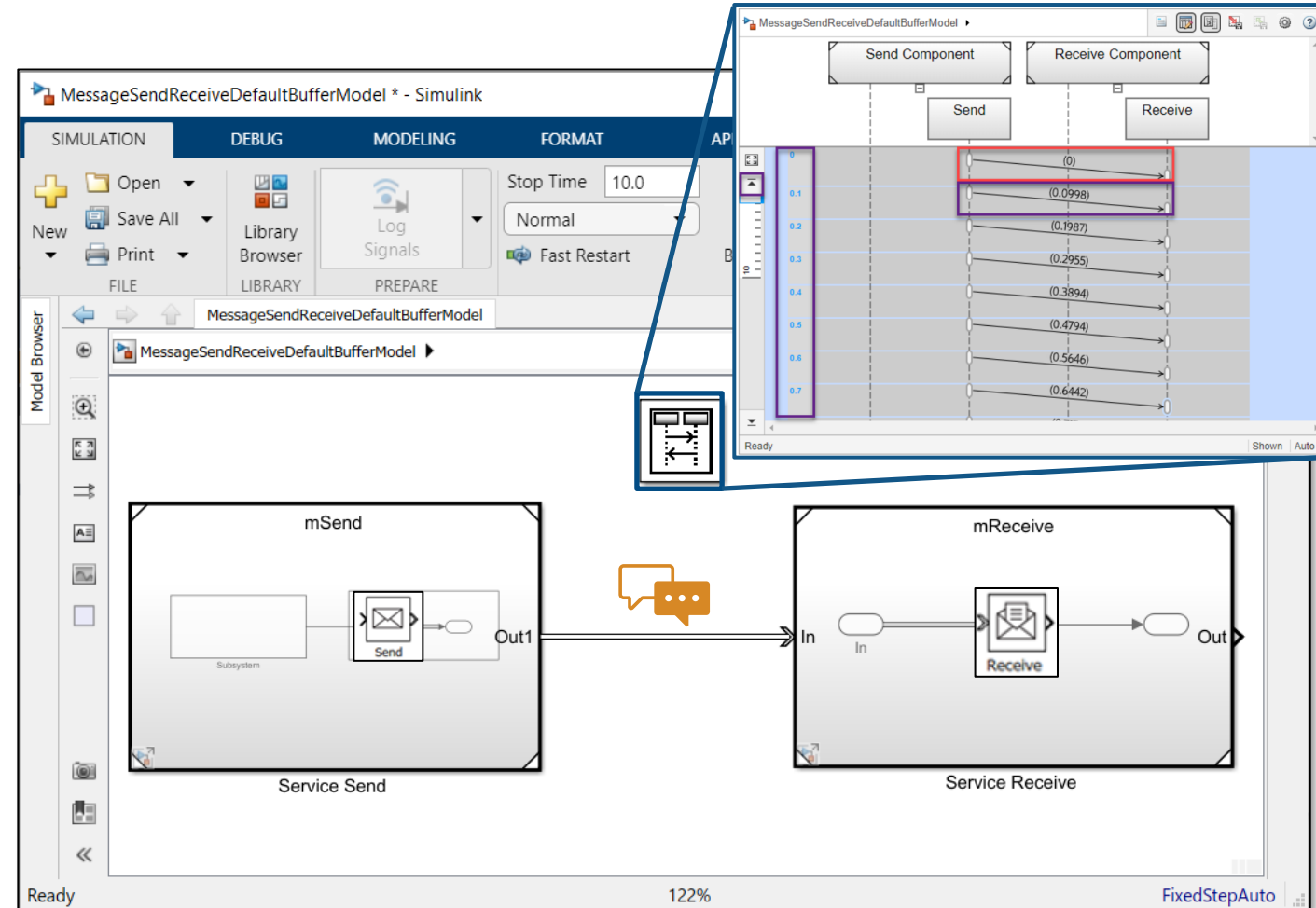
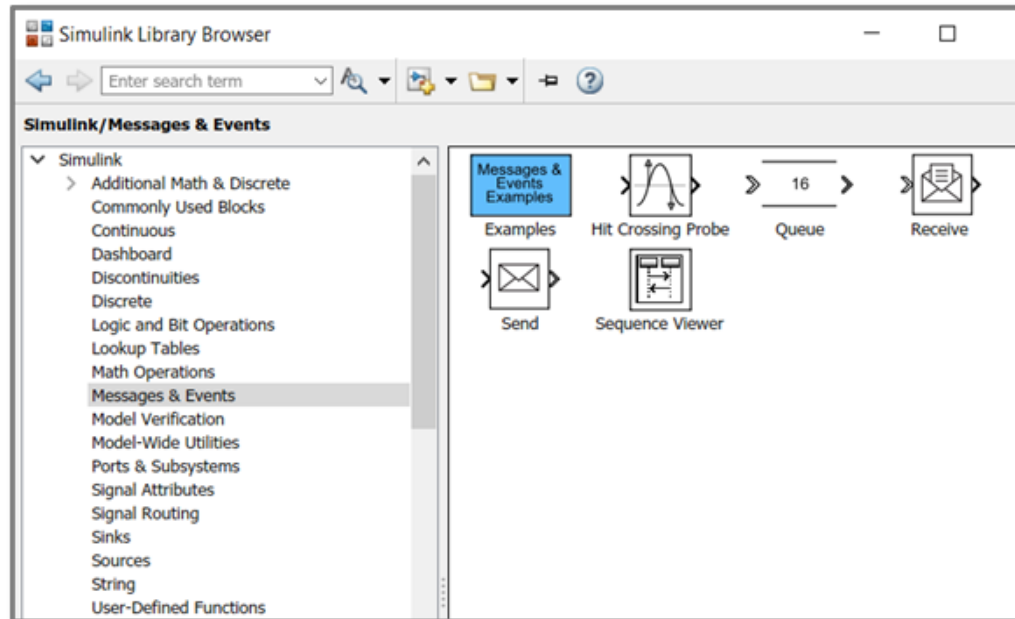
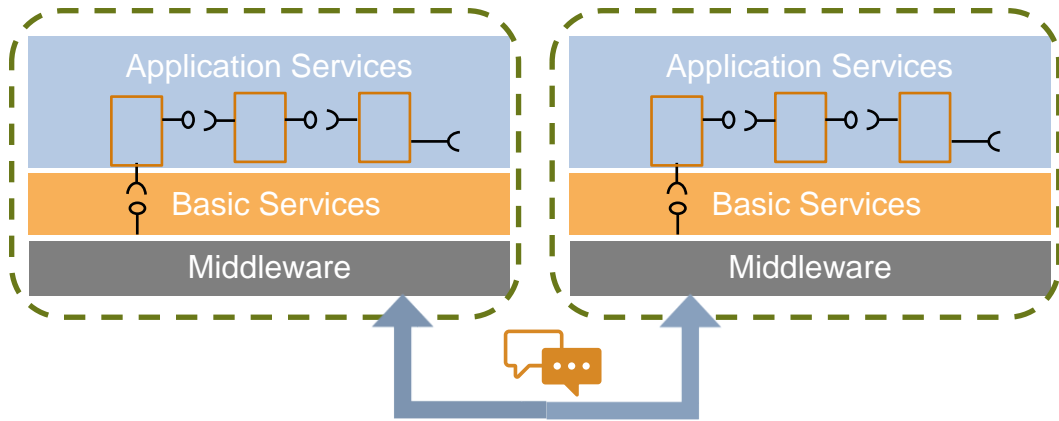
No, but I'm going to work on SOA in near future

No, I'm not planning to work on SOA

Agenda

- Evolution of architectures and key challenges
- **Simulink for service-oriented architectures**
- Conclusions and key takeaways

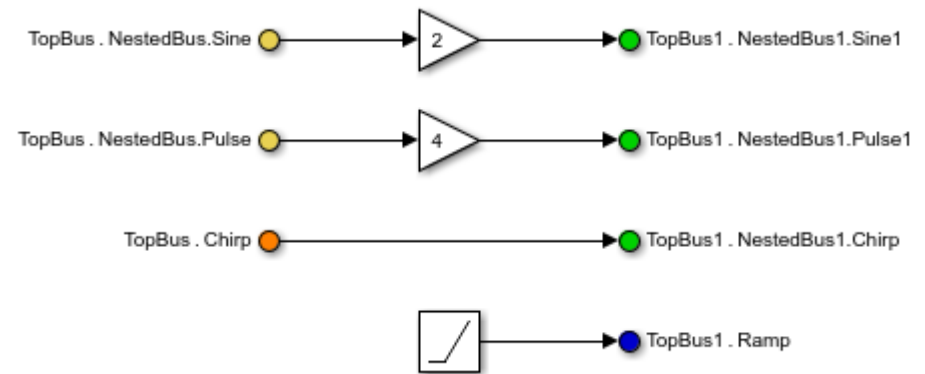
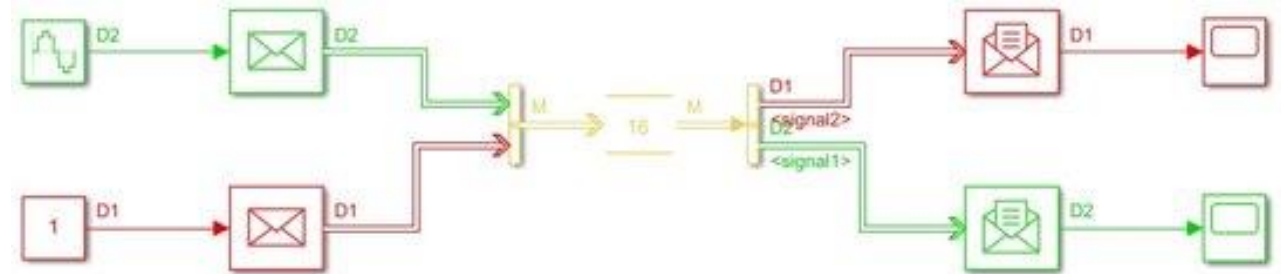
Simulink Messages for Service-oriented communication



You can model service-oriented communication using messages (Send/Receive).

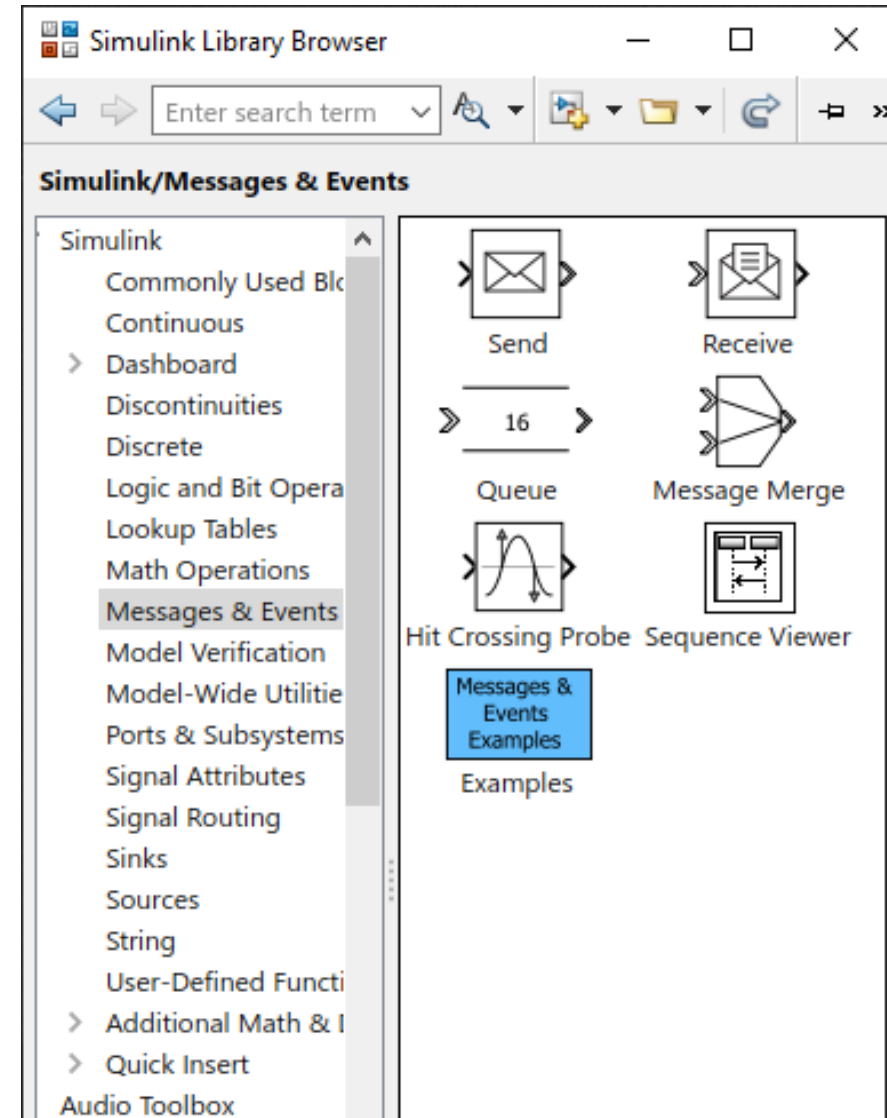
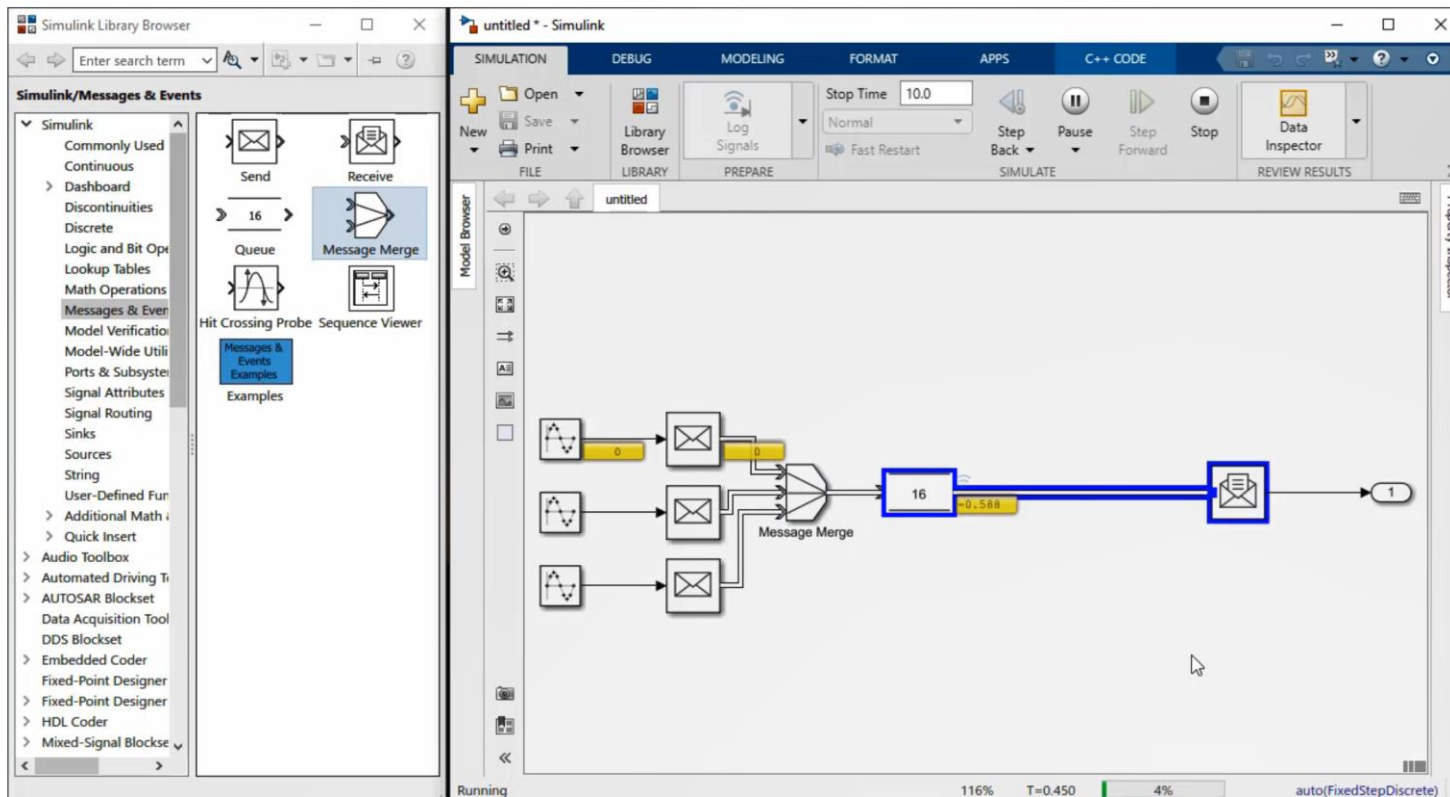
Bus Support for Messages

- Message lines can be combined via a bus
- Can use a queue on the bus
- Accessed either via a bus selector or bus element port



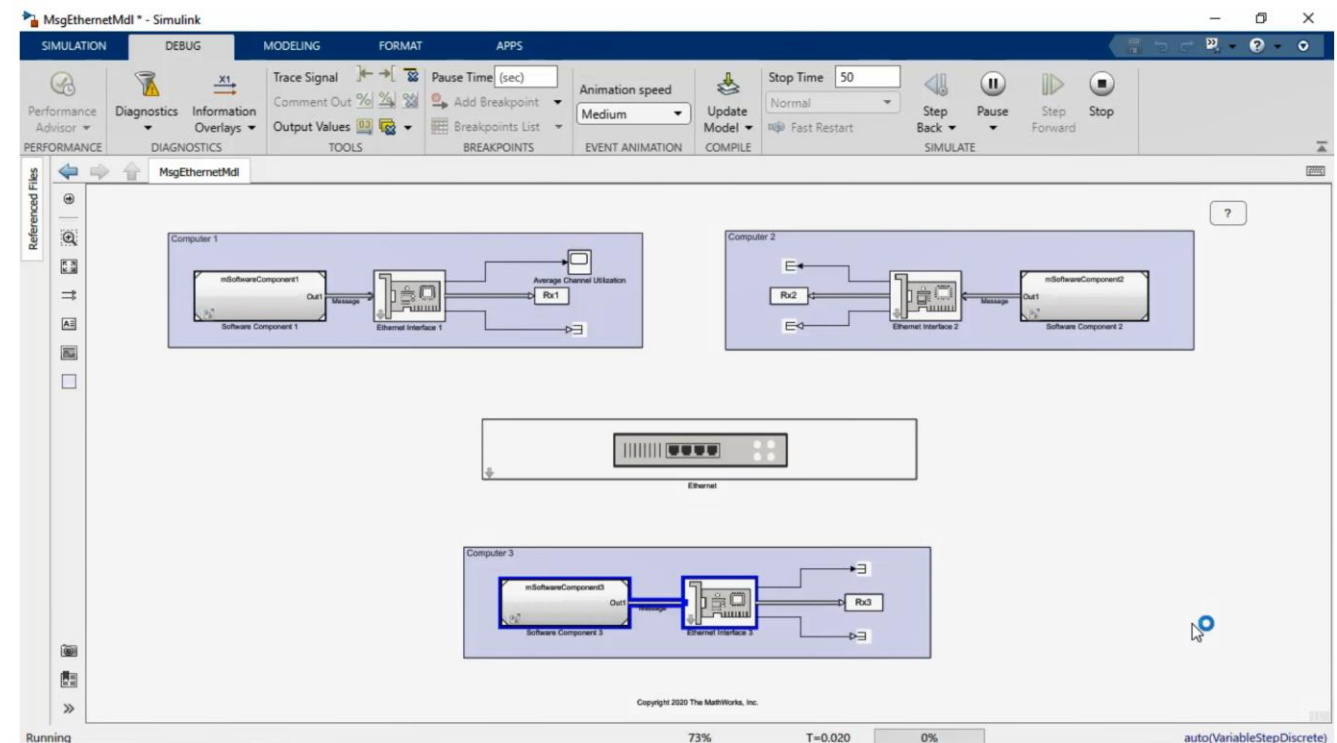
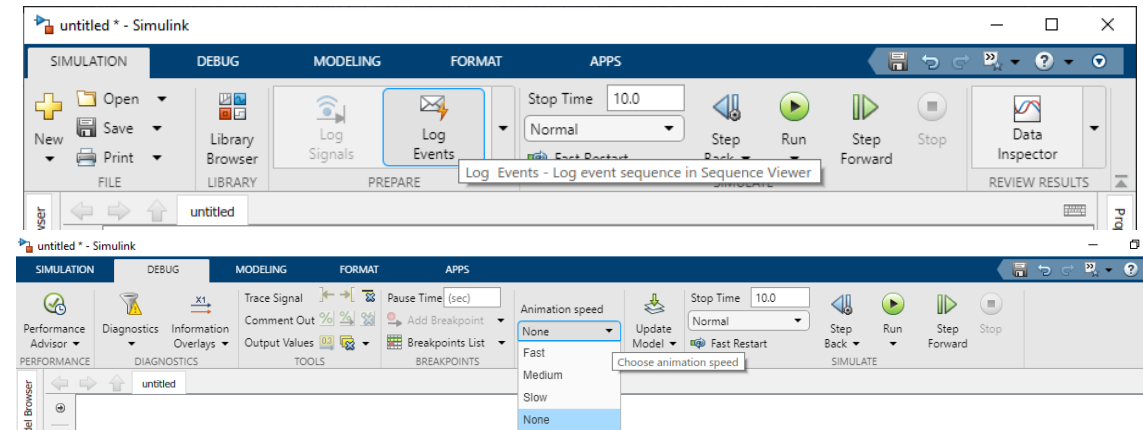
Message Merge

- Combine multiple message lines into a single message line
- Specify both nonvirtual and virtual buses to carry Messages.



Event-Based Logging and Animation

- Inspect and animate events in the model involving:
 - Function-Call Subsystems
 - Simulink Messages
 - Simulink Functions
- Using
 - Event Logging
 - Event Animation
 - Sequence Viewer tools.



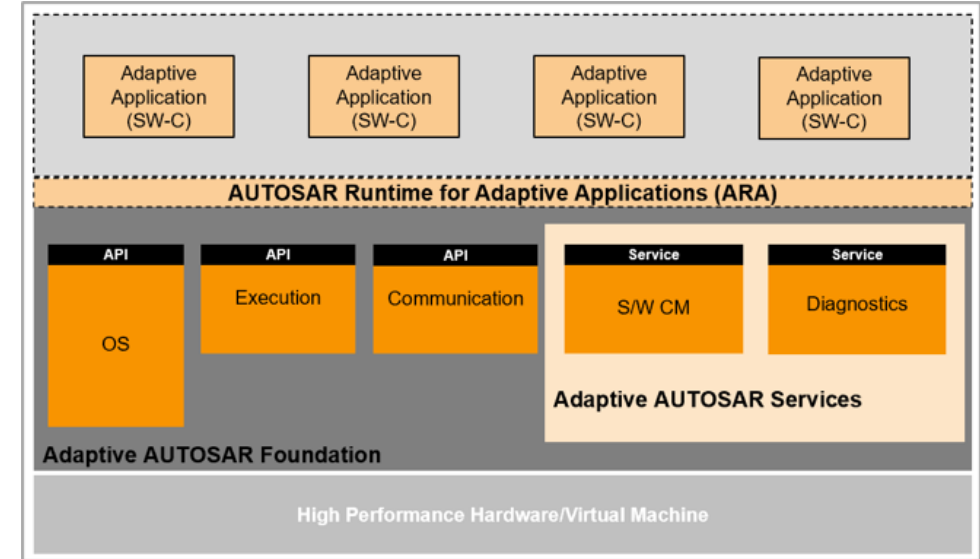
Agenda

- Evolution of architectures and key challenges
- Simulink for service-oriented architectures (SOA)
- **Simulink support for SOA industry standards**
 - Simulink for AUTOSAR Adaptive
 - Simulink for Data Distribution Service (DDS)
- Conclusions and key takeaways

Simulink support for SOA industry standards

AUTOSAR Adaptive Platform implements the AUTOSAR Runtime for Adaptive Applications (ARA) for automotive industry.

Model, simulate, test and generate code for AUTOSAR Adaptive applications in Simulink.



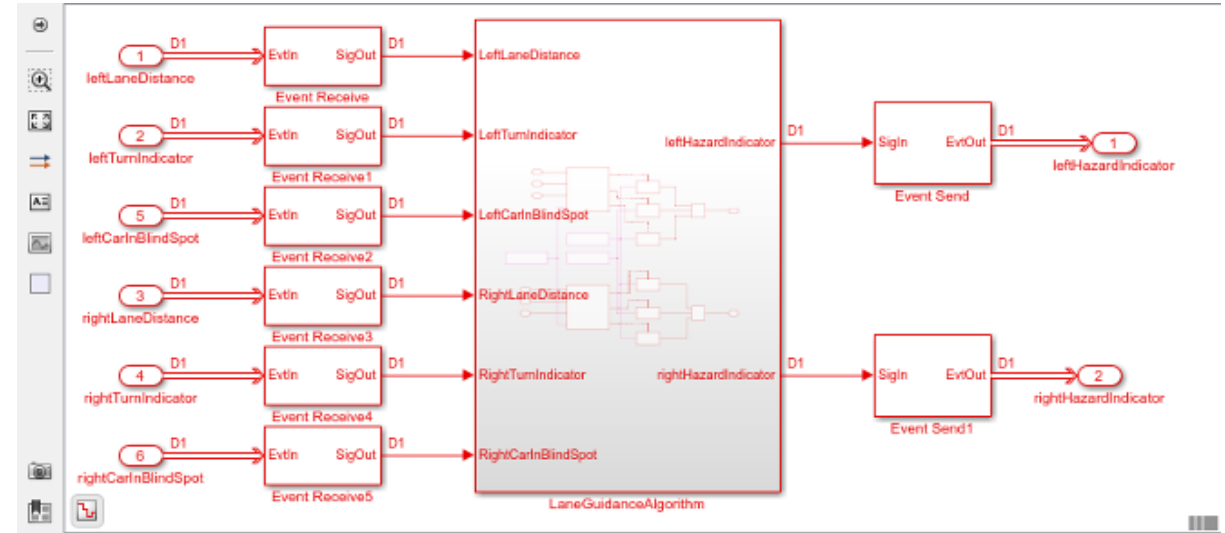
AUTOSAR Blockset

Design and simulate AUTOSAR software

[Request a free trial](#)

Simulink support for AUTOSAR Adaptive

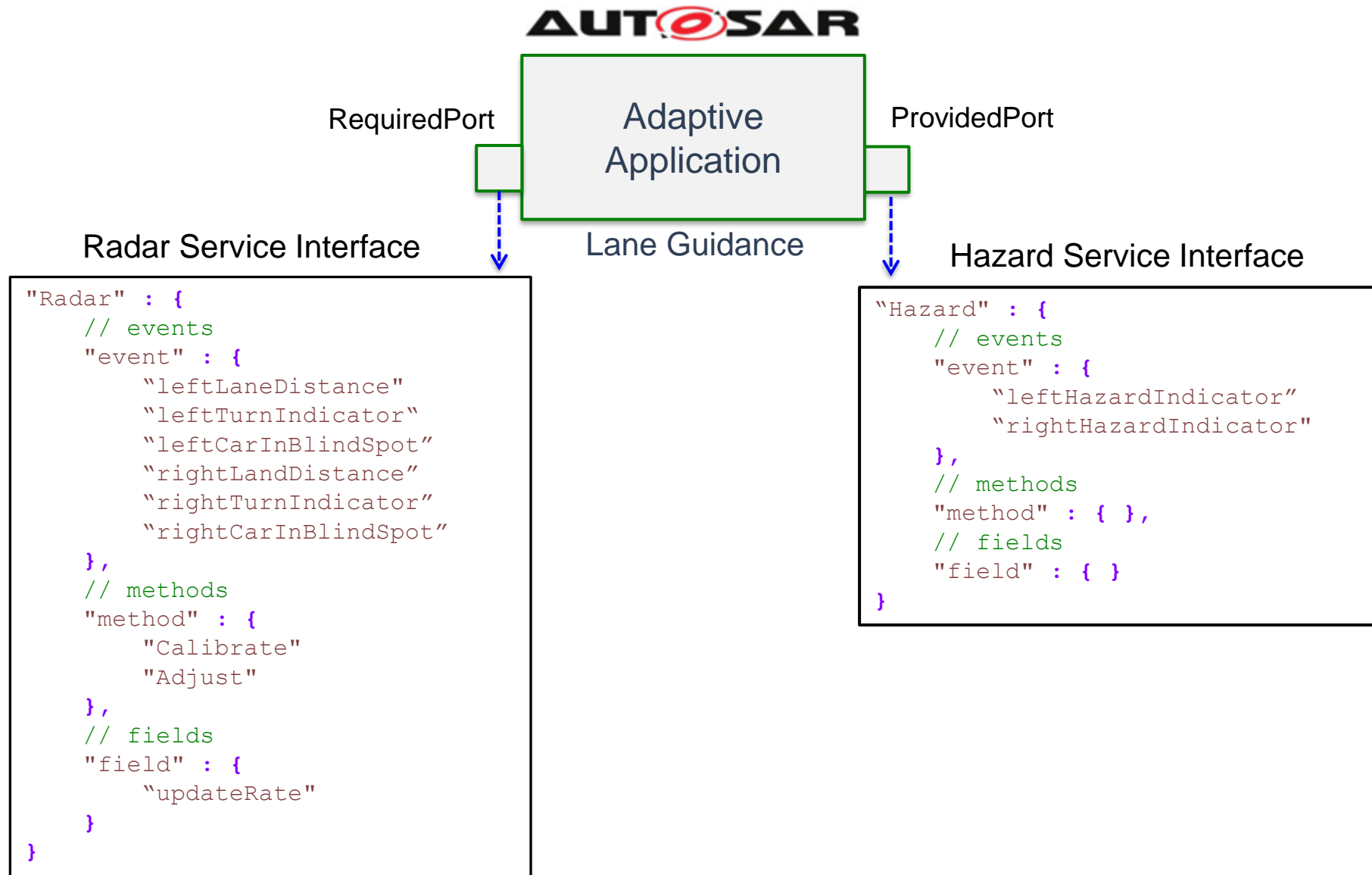
- In AUTOSAR Adaptive, services implement **communication** through:
 - Events
 - Methods
 - Fields
- In Simulink, **Events** can be modeled as **Messages** and then configured for code generation **using AUTOSAR Blockset**.



Name	SwCalibrationAccess	DisplayFormat
leftHazardIndicator	ReadOnly	
rightHazardIndicator	ReadOnly	

AUTOSAR Adaptive C++ compliant code is generated by Embedded Coder.

Adaptive SW architecture concepts



Modelling an AUTOSAR Adaptive application in Simulink

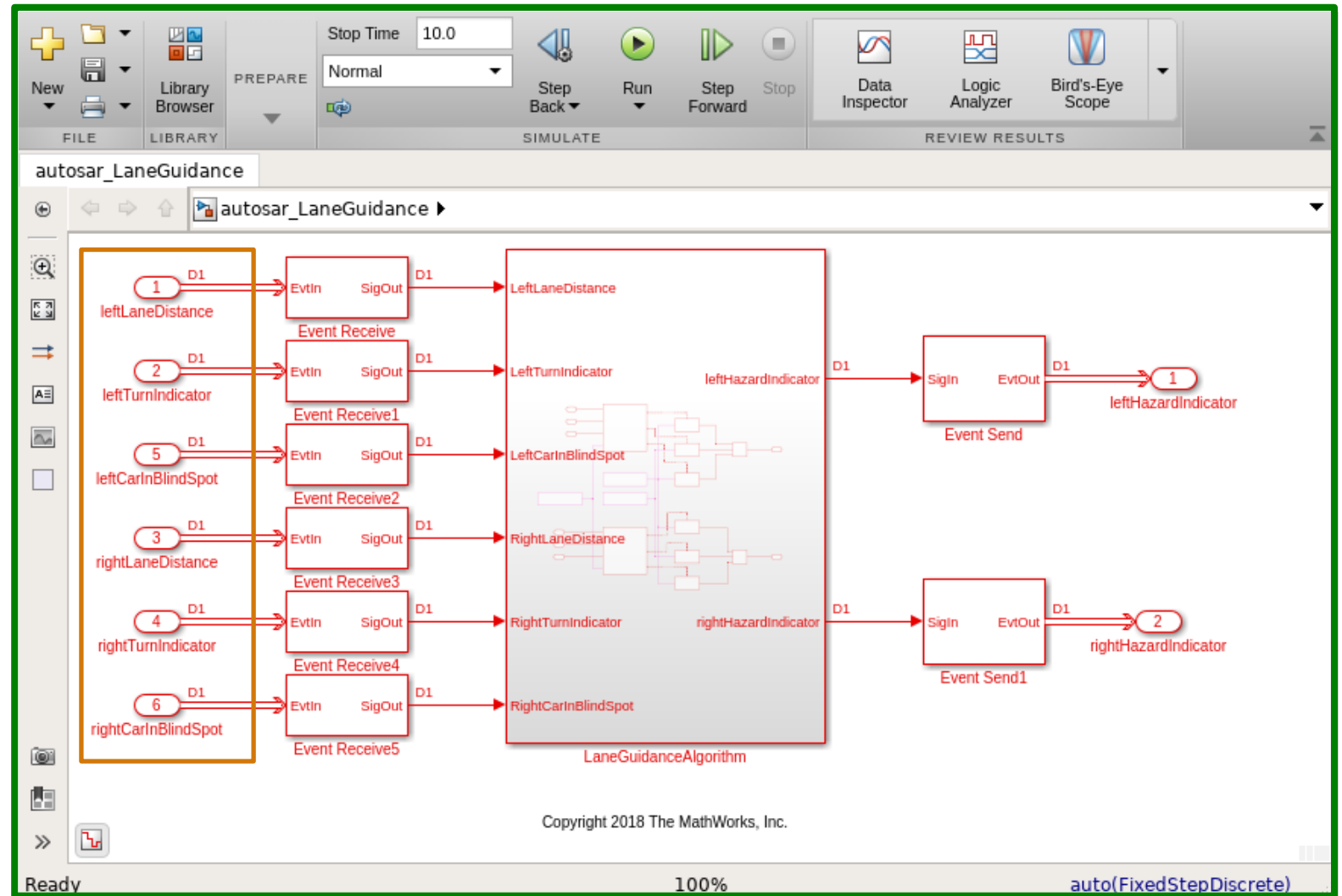


Adaptive Application

RequiredPort

```

"Radars" : {
  // events
  "event" : {
    "leftLaneDistance"
    "leftTurnIndicator"
    "leftCarInBlindSpot"
    "rightLaneDistance"
    "rightTurnIndicator"
    "rightCarInBlindSpot"
  },
  // methods
  "method" : {
    "Calibrate"
    "Adjust"
  },
  // fields
  "field" : {
    "updateRate"
  }
}
    
```



Modelling an AUTOSAR Adaptive application in Simulink

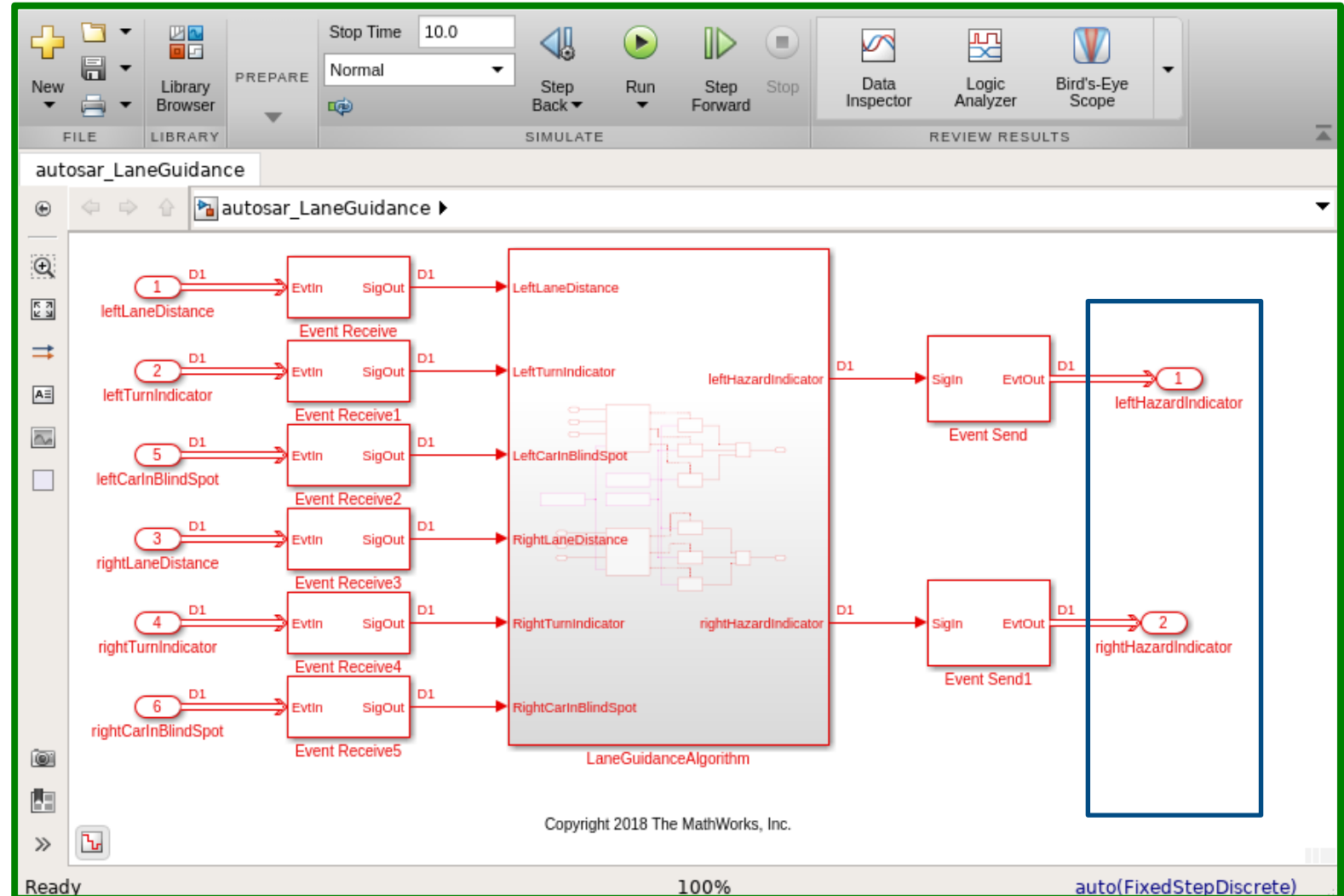


Adaptive Application

ProvidedPort

```

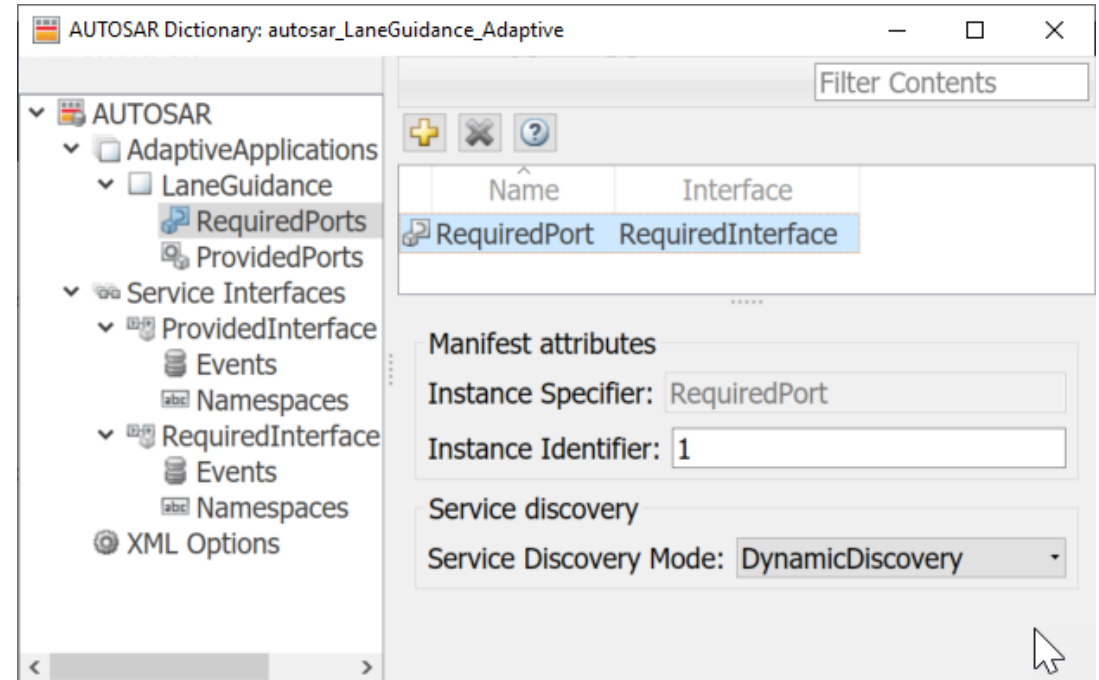
"Hazard" : {
  // events
  "event" : {
    "leftHazardIndicator"
    "rightHazardIndicator"
  },
  // methods
  "method" : { },
  // fields
  "field" : { }
}
    
```



Dynamic Service Discovery

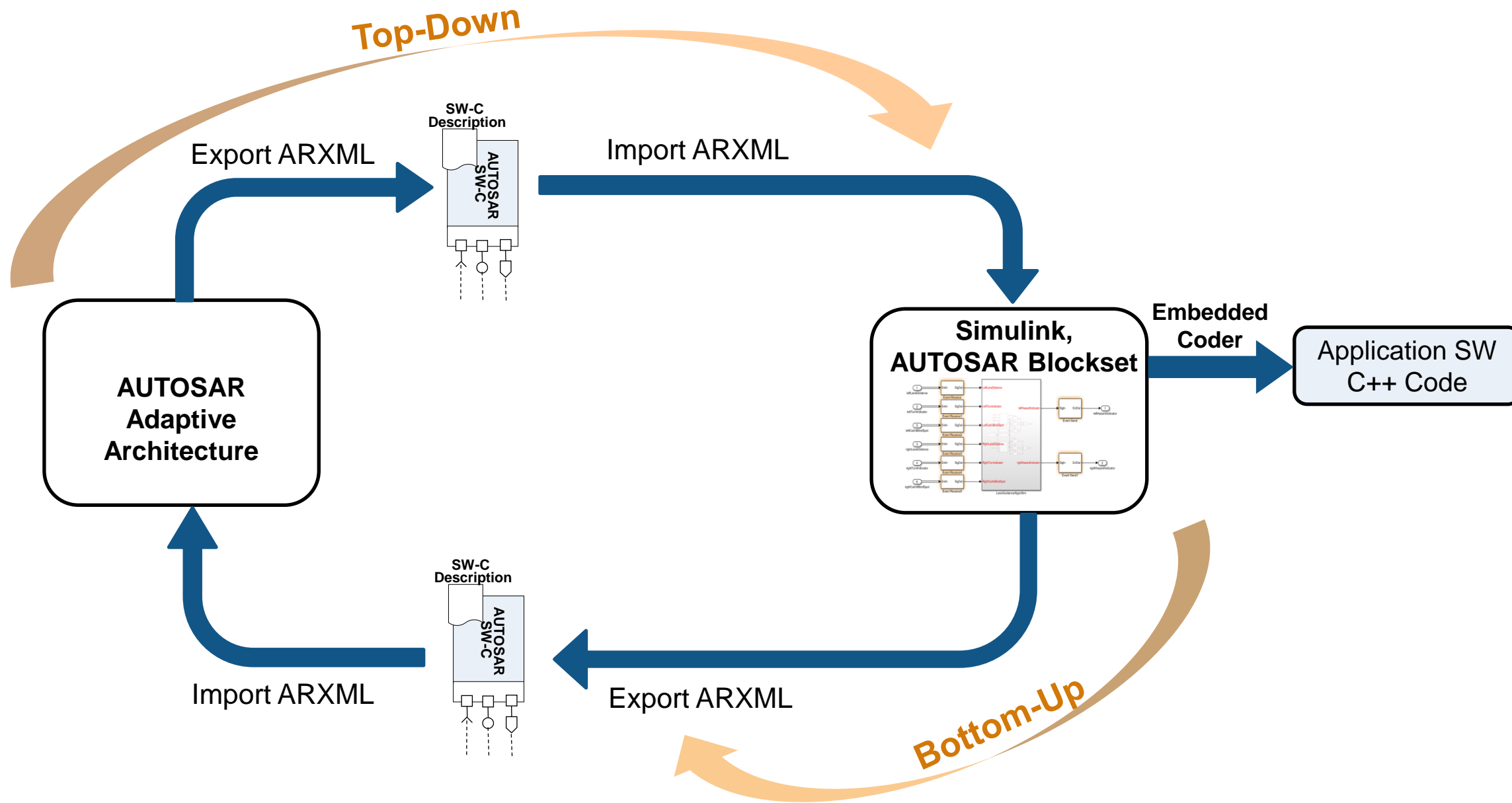
Find adaptive services by using dynamic discovery

- Configure AUTOSAR adaptive applications to **discover and subscribe to adaptive services** as they become available
- You can also configure service port programmatically as OneTime or DynamicDiscovery



```
apiObj = autosar.api.getAUTOSARProperties("autosar_LaneGuidance");
apiObj.set("/LaneGuidance_pkg/LaneGuidance_swc/LaneGuidance/RequiredPort/",
"ServiceDiscoveryMode", "DynamicDiscovery")
```

AUTOSAR Adaptive workflows



AUTOSAR Adaptive in action

- Create model from ARXML

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <!--
3 Auto generated XML Component Description for model autosar_LaneGuidance
4 Model version : 1.224
5 Simulink Coder version : Simulink Coder 9.2 (R2019b) 23-May-2019
6 XML source code generated on : Wed Jul 24 16:11:51 2019
7 Model Checksum : 3376303272 3457889089 3078584661 1517304406
8 -->
9 <AUTOSAR xmlns="http://autosar.org/schema/r4.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://
10 <AR-PACKAGES>
11 <AR-PACKAGE>
12 <SHORT-NAME>LaneGuidance_pkg</SHORT-NAME>
13 <AR-PACKAGE>
14 <AR-PACKAGE>
15 <SHORT-NAME>LaneGuidance_sw</SHORT-NAME>
16 <ELEMENTS>
17 <ADAPTIVE-APPLICATION-SW-COMPONENT-TYPE UUID="6574ed24-7dad-53cc-e7ac-01f60699f406">
18 <SHORT-NAME>LaneGuidance</SHORT-NAME>
19 <PORTS>
20 <R-PORT-PROTOTYPE UUID="a8adc3c3-bbb1-575e-fbc6-0fcf8164f622">
21 <SHORT-NAME>RequiredPort</SHORT-NAME>
22 <REQUIRED-COM-SPECS>
23 <QUEUED-RECEIVER-COM-SPEC>
24 <DATA-ELEMENT-REF DEST="VARIABLE-DATA-PROTOTYPE"/>LaneGuidance_pkg/LaneGuidance_if/R
25 <HANDLE-OUT-OF-RANGE>NONE</HANDLE-OUT-OF-RANGE>
26 <USES-END-TO-END-PROTECTION>false</USES-END-TO-END-PROTECTION>
27 <QUEUE-LENGTH>1</QUEUE-LENGTH>
28 </QUEUED-RECEIVER-COM-SPEC>

```

Configuration Parameters: LaneGuidance/Configuration (Active)

Search

Solver

Data Import/Export

Math and Data Types

▶ Diagnostics

Hardware Implementation

Model Referencing

Simulation Target

▼ Code Generation

Optimization

Report

Comments

Identifiers

Custom Code

Interface

Code Style

Verification

Templates

Code Placement

Data Type Replacement

AUTOSAR Code Generat...

Coverage

▶ HDL Code Generation

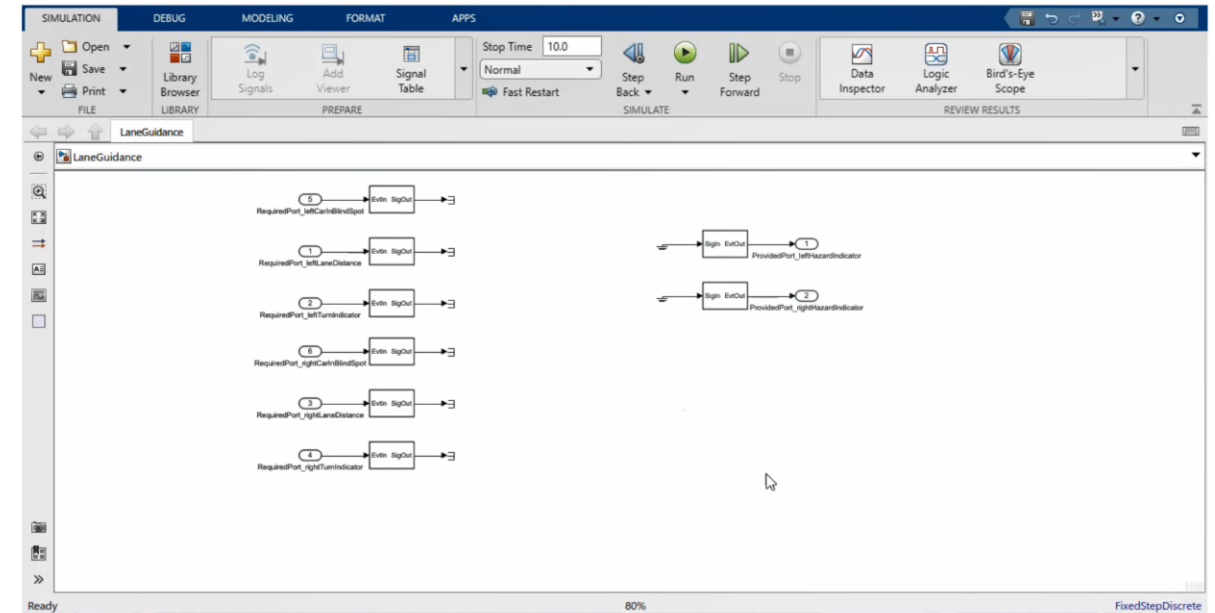
Generate XML file for schema version: 00046 (R18-10) | 00047 (R19-03) | 00048 (R19-11)

Maximum SHORT-NAME length: 128

XCP Slave Configuration: None

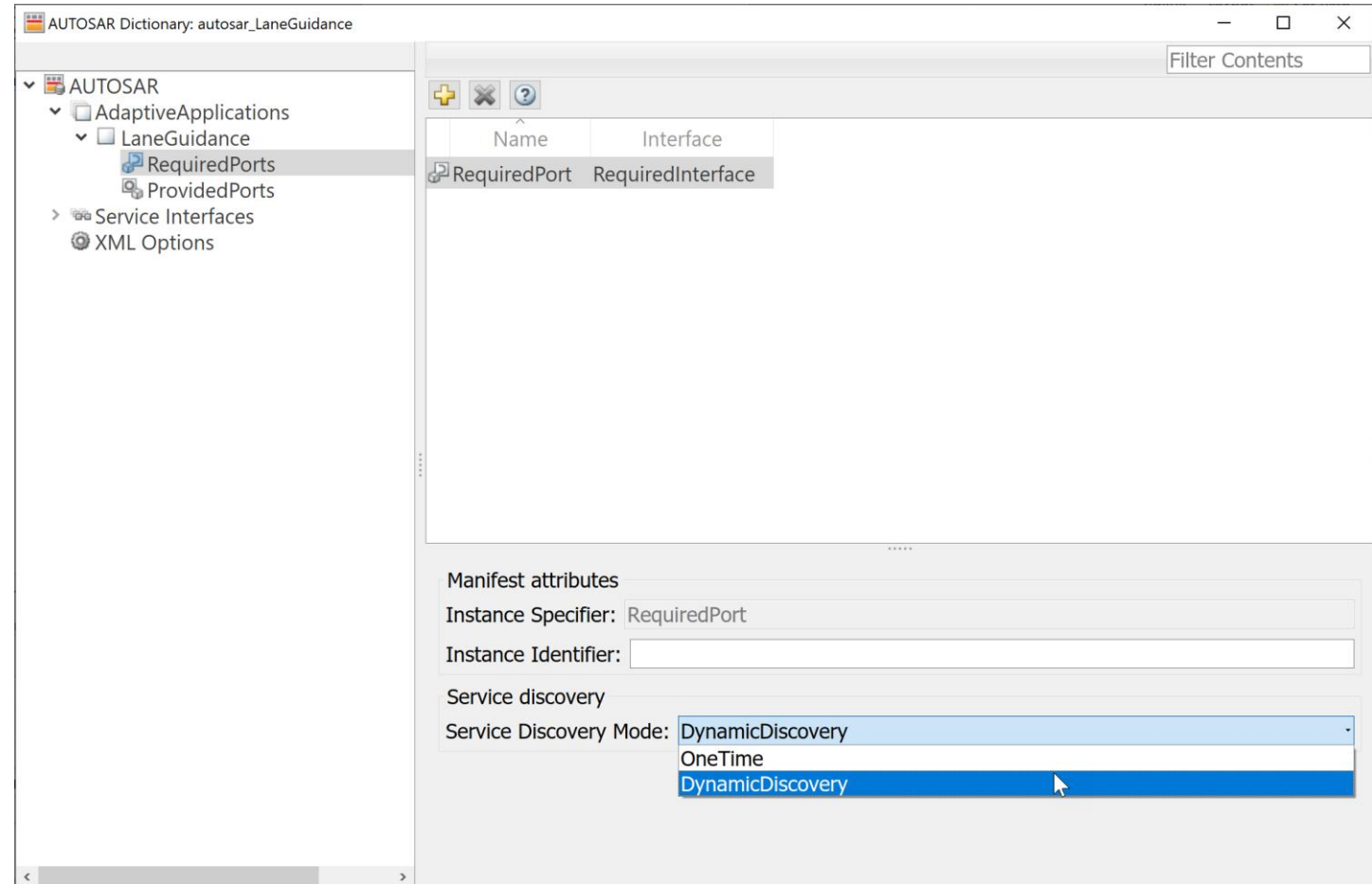
Transport layer: None

OK Cancel Help Apply



AUTOSAR Adaptive in action

- Create model from ARXML
- Configure Service Discovery
 - Subscribe to adaptive services
 - Only at startup, or
 - Dynamically, as they become available



AUTOSAR Adaptive in action

- Create model from ARXML
- Configure Service Discovery
- Verify AUTOSAR properties

The screenshot displays the MATLAB Simulink interface for the AUTOSAR Adaptive component. The main workspace shows a block diagram for 'autosar_LaneGuidance'. On the left, there are six input ports: 'leftLaneDistance' (1), 'leftTurnIndicator' (2), 'leftCarInBlindSpot' (5), 'rightLaneDistance' (3), 'rightTurnIndicator' (4), and 'rightCarInBlindSpot' (6). Each input port is connected to an 'Event Receive' block. These blocks feed into a central 'LaneGuidanceAlgorithm' block, which outputs 'LeftLaneDistance', 'LeftTurnIndicator', and 'LeftCarInBlindSpot'. The algorithm also outputs 'leftHazardIndicator', which is connected to an 'Event Send' block. The 'Event Send' block outputs 'leftHazardIndicator' (1).

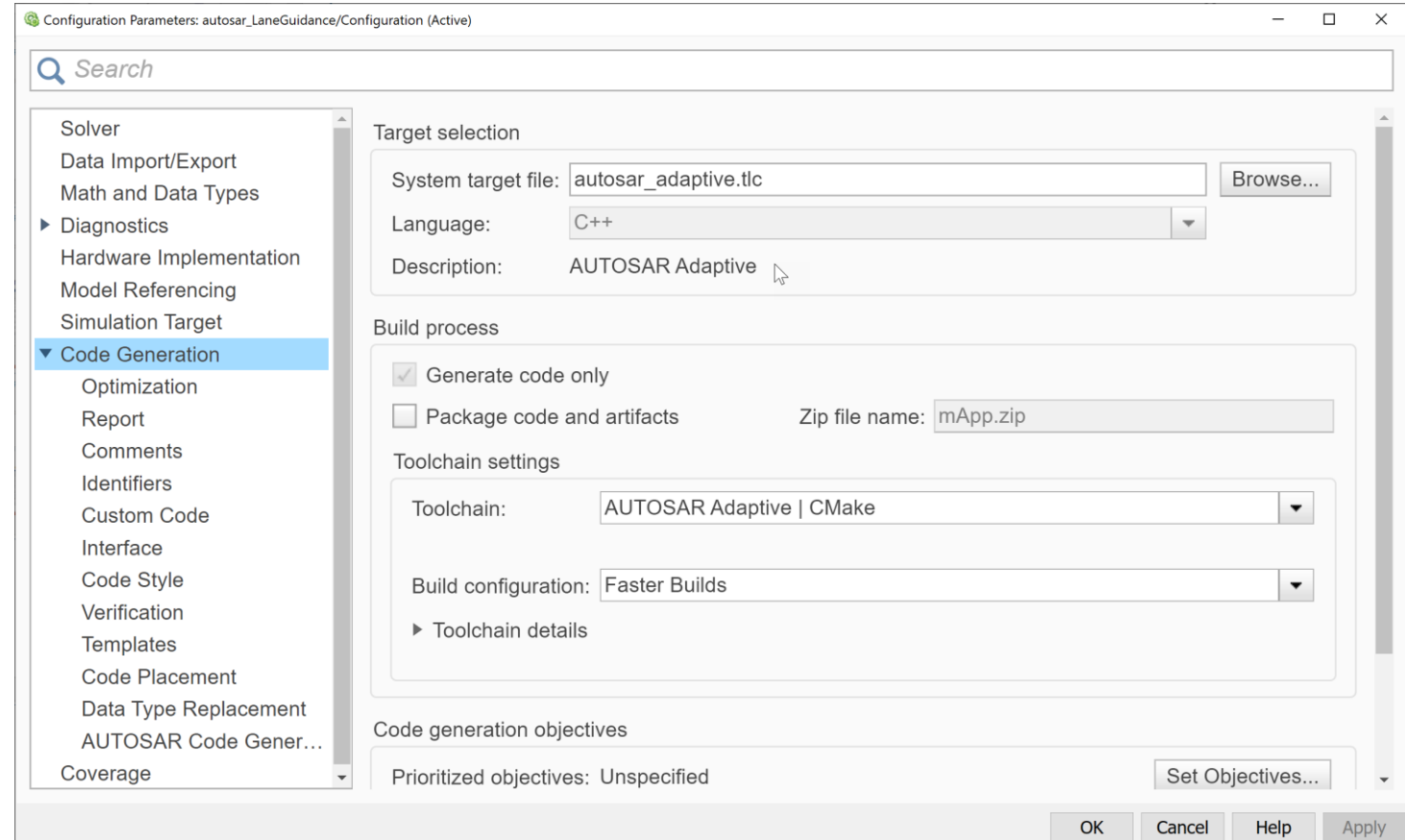
An 'AUTOSAR Validation' window is overlaid on the diagram, displaying 'Validation succeeded' with a green progress bar. Below the diagram, the 'Code Mappings - AUTOSAR SW Component (Adaptive)' window is open, showing a table of inports and outports.

Inports	Port	Event
leftLaneDistance	RequiredPort	LeftLaneDistance
leftTurnIndicator	RequiredPort	LeftTurnIndicator
rightLaneDistance	RequiredPort	RightLaneDistance
rightTurnIndicator	RequiredPort	RightTurnIndicator
leftCarInBlindSpot	RequiredPort	LeftCarInBlindSpot
rightCarInBlindSpot	RequiredPort	RightCarInBlindSpot

Copyright 2018 The MathWorks, Inc.

AUTOSAR Adaptive in action

- Create model from ARXML
- Configure Service Discovery
- Verify AUTOSAR properties
- **Generate code**



AUTOSAR Adaptive in action

- Create model from ARXML
- Configure Service Discovery
- Verify AUTOSAR properties
- Generate code

The screenshot displays the MATLAB AUTOSAR Adaptive tool interface. The top menu bar includes SIMULATION, DEBUG, MODELING, FORMAT, APPS, and AUTOSAR. The main workspace shows a block diagram of the 'autosar_LaneGuidance' model with various ports and signals. Below the diagram is the 'Code Mappings - AUTOSAR SW Component (Adaptive)' section, which includes a table mapping source ports to target ports and events.

Source	Port	Event
RequiredPort_leftLaneDistance	RequiredPort	LeftLaneDistance
RequiredPort_leftTurnIndicator	RequiredPort	LeftTurnIndicator

The right pane shows the generated C++ code for 'autosar_LaneGuidance.cpp'. The code includes comments for service discovery and function definitions for message reception.

```

23 // '<S9>/IfActionSS3'
24 // '<S9>/IfActionSS4'
25 // '<S9>/IfActionSS5'
26 //
27 void autosar_LaneGuidanceModelClass::autosar_LaneGuidance_IfActionSS(real_T
28   rtu_In1, real_T *rtu_Out1)
29 {
30   // Inport: '<S10>/In1'
31   *rtu_Out1 = rtu_In1;
32 }
33
34 void autosar_LaneGuidanceModelClass::RequiredPortLeftLaneDistanceReceive(ara::
35   com::SamplePtr< company::chassis::required::proxy::events::LeftLaneDistance::
36   SampleType const > samplePtr)
37 {
38   // Receive: '<S1>/Message Receive'
39   autosar_LaneGuidance_B.MessageReceive = *samplePtr;
40 }
41
42 void autosar_LaneGuidanceModelClass::RequiredPortLeftTurnIndicatorReceive(ara::
43   com::SamplePtr< company::chassis::required::proxy::events::LeftTurnIndicator::
44   SampleType const > samplePtr)
45 {
46   // Receive: '<S3>/Message Receive'
  
```

AUTOSAR Adaptive in action

- Create model from ARXML
- Configure Service Discovery
- Verify AUTOSAR properties
- Generate code

The screenshot displays the MATLAB AUTOSAR Adaptive tool interface. The top menu bar includes SIMULATION, DEBUG, MODELING, FORMAT, APPS, and AUTOSAR. The main workspace shows a block diagram of the 'autosar_LaneGuidance' model with various input and output ports. The 'Code' window on the right shows the generated C++ code for 'autosar_LaneGuidance.cpp'. A search bar is visible in the code window, and a file explorer overlay is shown, listing files such as 'autosar_LaneGuidance.cpp', 'autosar_LaneGuidance.h', and various interface files.

Code Mappings - AUTOSAR SW Component (Adaptive)

Source	Port	Event
RequiredPort_leftLaneDistance	RequiredPort	LeftLaneDistance
RequiredPort_leftTurnIndicator	RequiredPort	LeftTurnIndicator

Integrate Applications with third party Adaptive stack

Poll Question #2

Are you working on AUTOSAR Adaptive applications? (Check all that apply)

Yes, I'm working on a production or prototyping project

Yes, I'm using Model-Based Design for AUTOSAR Adaptive

Yes, I'm writing C++ code for AUTOSAR Adaptive

No, but I'm going to work on AUTOSAR Adaptive in near future

No, I'm not planning to work on AUTOSAR Adaptive

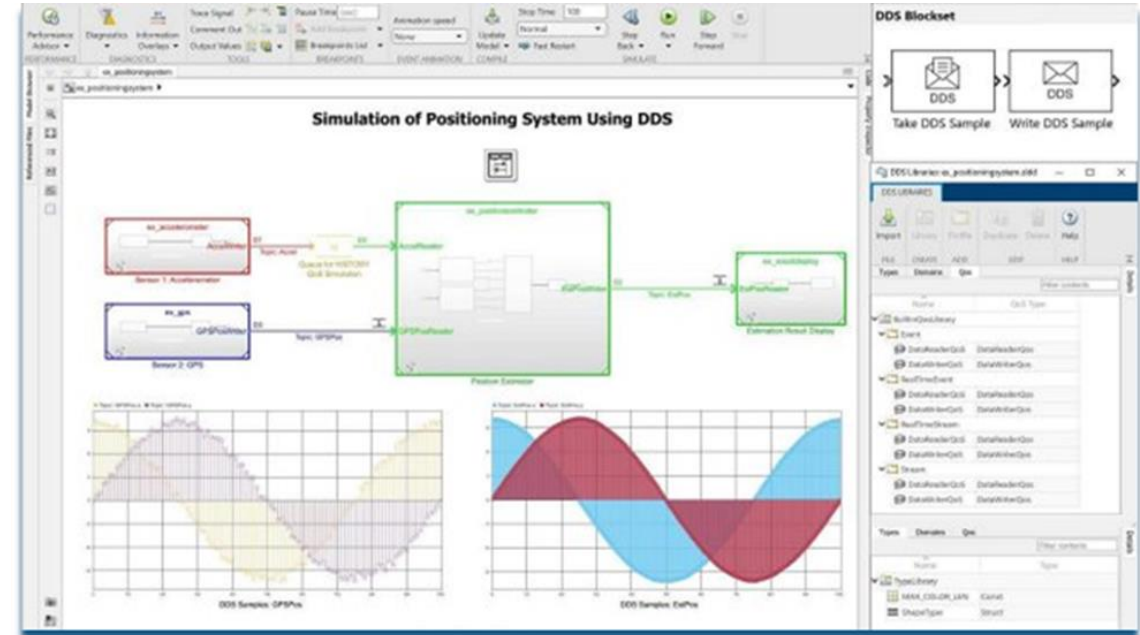
Data Distribution Services (DDS)



Data Distribution Services (DDS) uses SOA methodology, and directly addresses publish and subscribe communications for real-time and embedded systems.



DDS addresses the needs of applications that require real-time data exchange in industries like aerospace and defense, automotive, and robotics.



R2021a

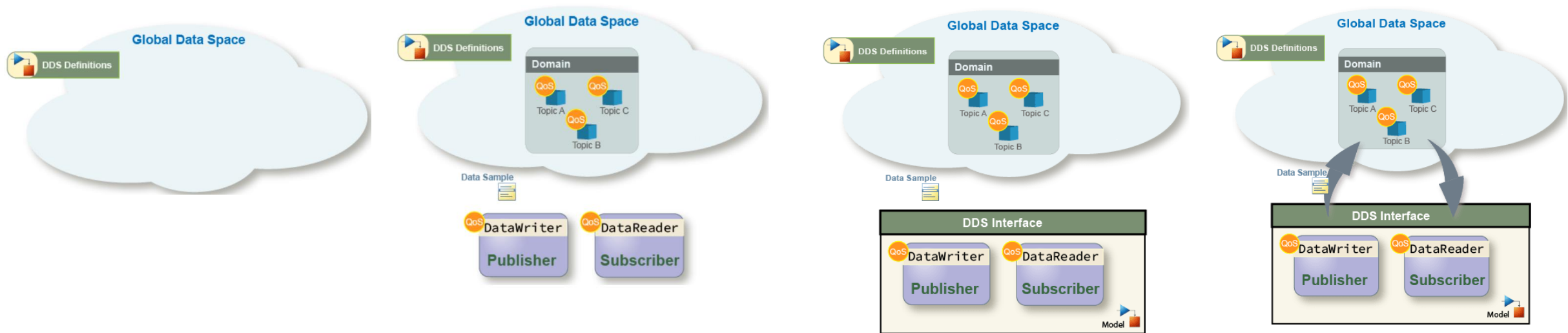
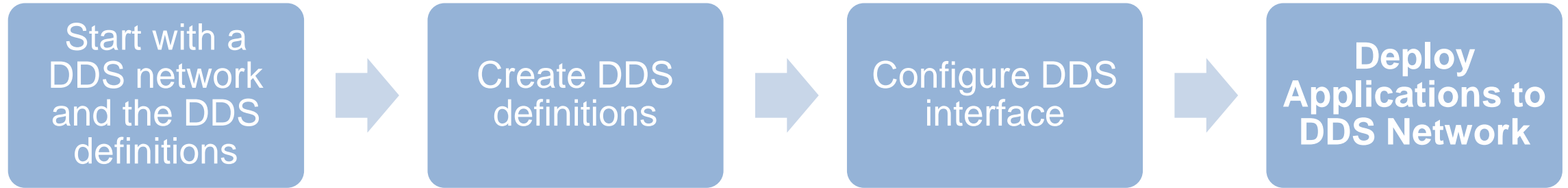
DDS Blockset

Design and simulate DDS applications

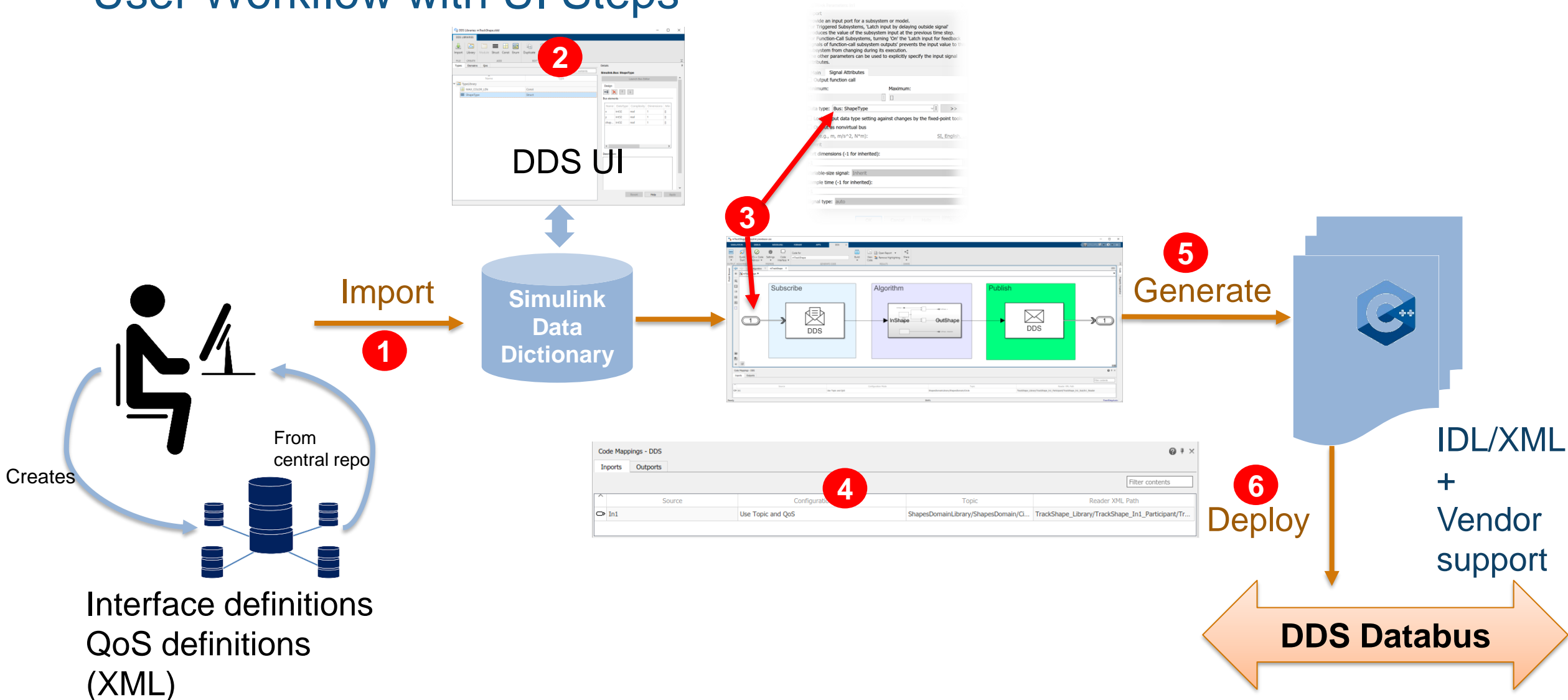
[Request a trial](#)

How does DDS Blockset Work?

Conceptual DDS Blockset Workflow

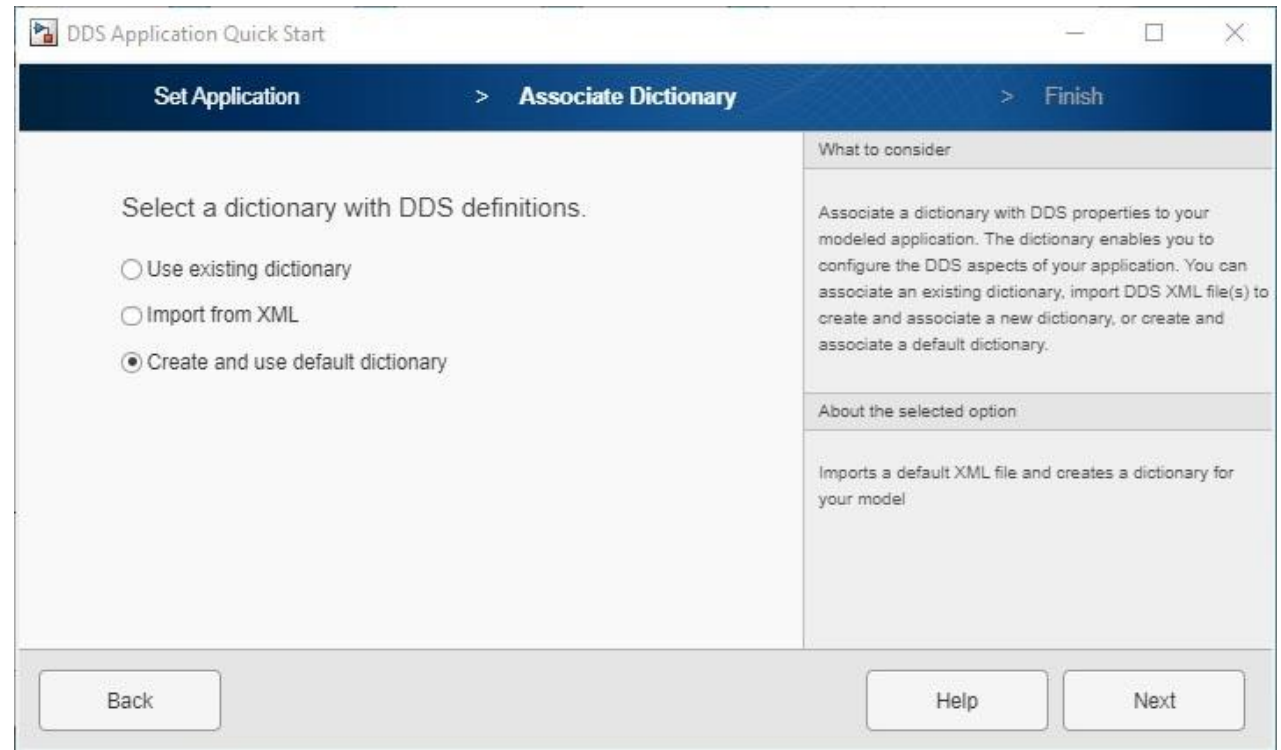


User Workflow with UI Steps



DDS Blockset in action

- Import DDS definitions from XML or create new Definitions



DDS Blockset in action

- Import DDS definitions from XML or create new Definitions
- Define/Modify DDS definitions in DDS Dictionary
 - Topic Types
 - Domains
 - QoS

The screenshot displays the 'DDS LIBRARIES' tool window for 'ShapesApp.sidd'. The interface includes a toolbar with icons for Import, Library, Domain, Topic, Duplicate, Delete, and Help. Below the toolbar are tabs for 'Types', 'Domains', and 'QoS'. A 'Filter contents' search box is present above the main table.

Name	Domain ID	Topic Type
ShapesDomainLibrary		
ShapesDomain	0	
Circle1		ShapeType1
Square1		ShapeType1
Triangle1		ShapeType1

The 'Details' panel on the right shows the 'Domain: ShapesDomain' with 'Domain ID: 0'. It includes sections for 'Registered Types' and 'Topics'. The 'Registered Types' section contains a table:

Name	TypeRef
ShapeType1	ShapeType1

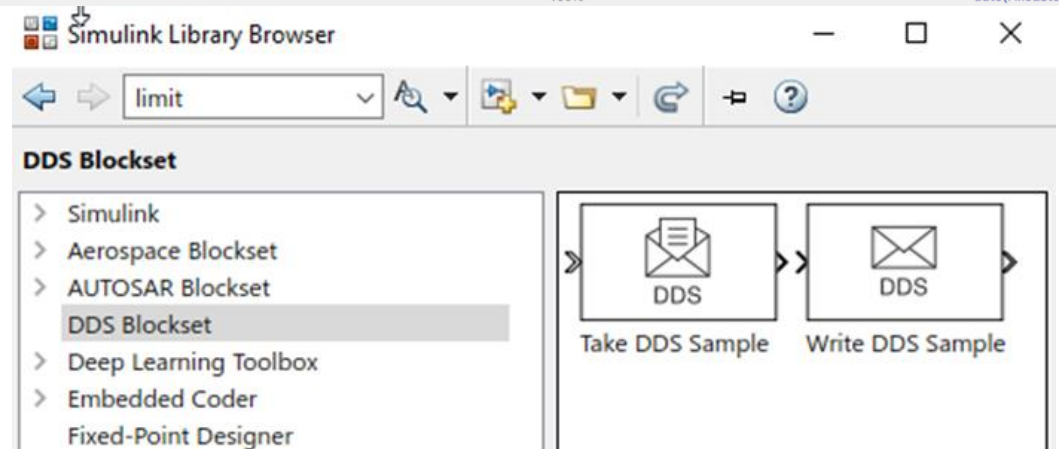
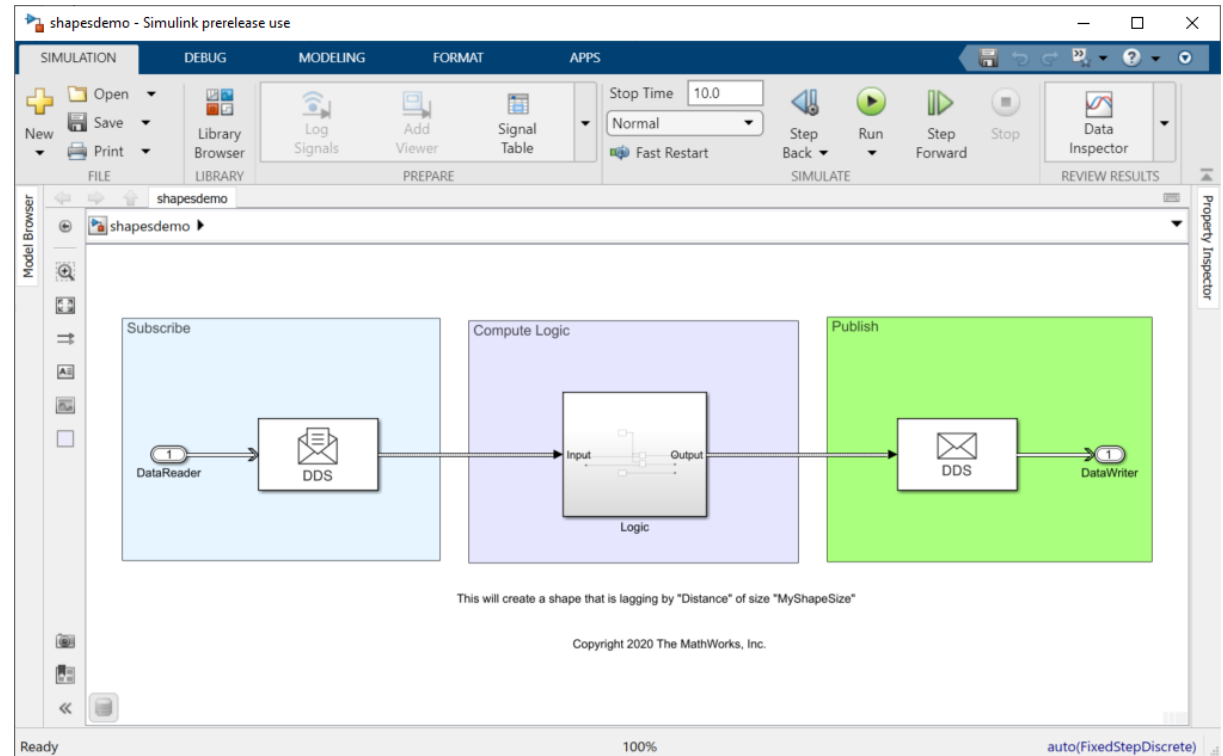
The 'Topics' section contains a table:

Name	RegisterTypeRef
Circle1	ShapeType1
Square1	ShapeType1
Triangle1	ShapeType1

DDS Blockset in action

- Import DDS definitions from XML or create new Definitions
- Define/Modify DDS definitions in DDS Dictionary
- Model applications

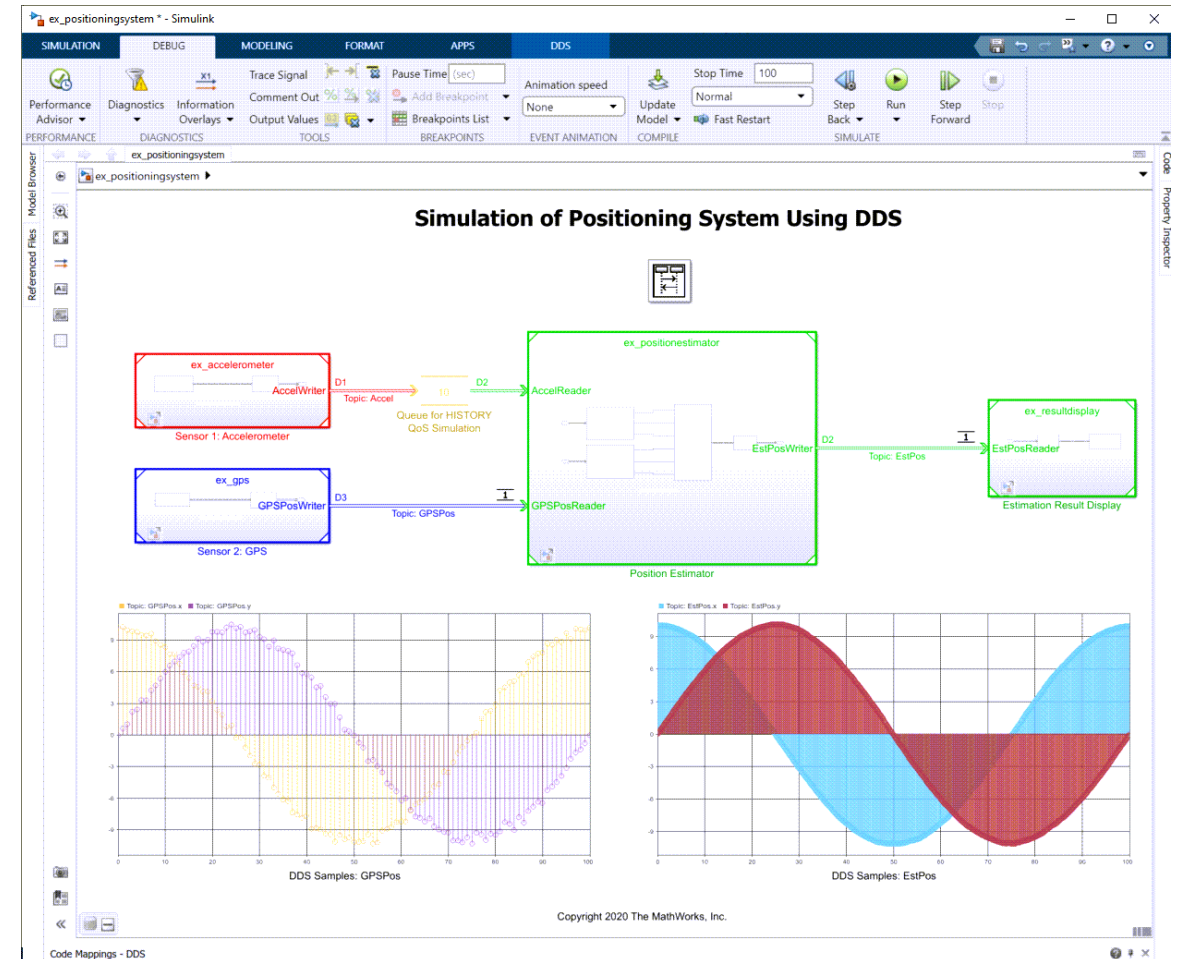
Use DDS Blocks to model a Publisher or Subscriber



DDS Blockset in action

- Import DDS definitions from XML or create new Definitions
- Define/Modify DDS definitions in DDS Dictionary
- Model applications
- Simulate DDS models including QoS

Use Simulink to model and simulation Quality of Services (QoS) policies including **history** to verify the runtime behavior.



DDS Blockset in action

- Import DDS definitions from XML or create new Definitions
- Define/Modify DDS definitions in DDS Dictionary
- Model applications
- Simulate DDS models including QoS
- Generate DDS executables and deploy on a DDS network

```
bool writeWithWriter(const PosType* data, std::string participantName, std::string writerName) {
    DDS_DataWriter* writer = getWriter(writerName, participantName);
    PosTypeDataWriter* foowriter = PosTypeDataWriter_narrow(writer);
    if(!foowriter) {
        return false;
    }
    const DDS_ReturnCode_t ret = PosTypeDataWriter_write((PosTypeDataWriter*)writer, data);
    return (ret == DDS_ReturnCode_t::DDS_RETCODE_OK);
};

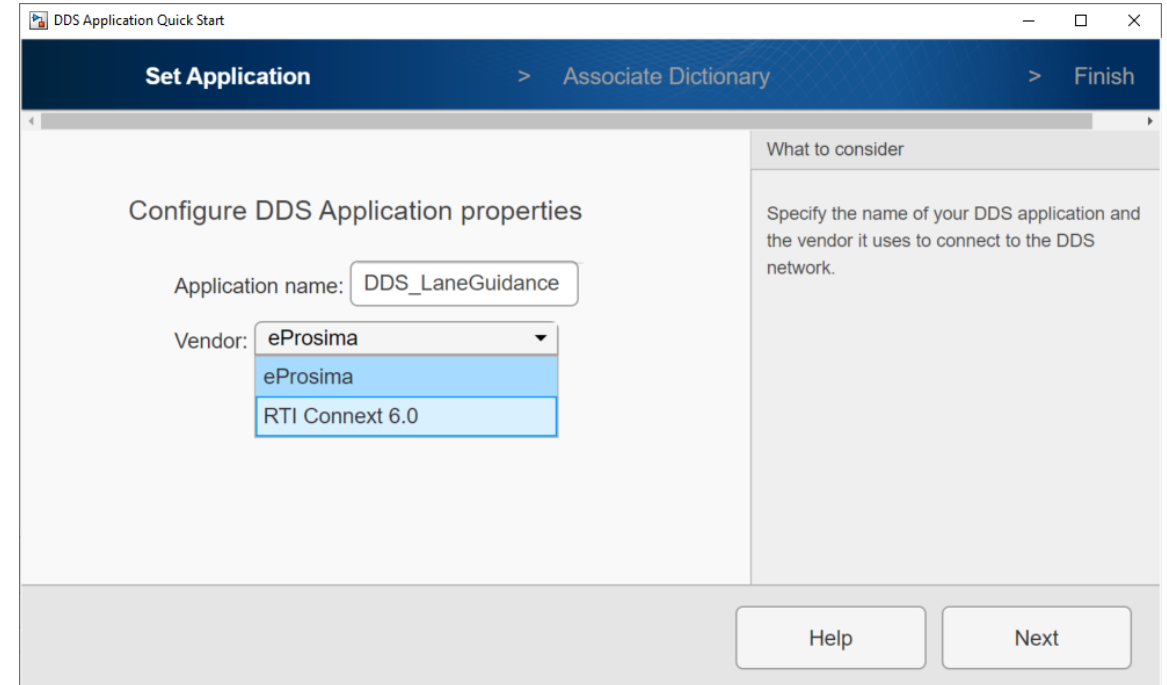
bool createParticipant(std::string participantName) {
    if (participants.find(participantName) == participants.end()) {
        DDS_DomainParticipant* participant =
            DDS_DomainParticipantFactory_create_participant_from_config(
                DDS_TheParticipantFactory, participantName.c_str());
        if(!participant) {
            return false;
        }
        participants[participantName] = participant;
    }
    return true;
};
```

With Embedded coder, generate

- C++ production code with DDS APIs
- XML or IDL files from Simulink models to deploy

DDS Blockset in action

- Import DDS definitions from XML or create new Definitions
- Define/Modify DDS definitions in DDS Dictionary
- Model applications
- Simulate DDS models including QoS
- Generate DDS executables and deploy on a DDS network



Full integration with third-party DDS stacks including RTI Connnext and eProsima Fast DDS

Poll Question #3

Are you working on DDS based applications? (Check all that apply)

Yes, I'm working on a production or prototyping project

Yes, I am generating C code from Simulink model configured for DDS

Yes, I am generating C++ code from Simulink model configured for DDS

No, but I'm going to work on DDS in near future

No, I'm not planning to use DDS

Conclusions and Key takeaways

- **Automotive E/E and SW architecture are evolving**, pushed by need for advanced, complex functions
- New, **service-oriented architectures** are required to **master complexity** and enable **frequent updates**
- You can **design, simulate and generate** code to deploy service-oriented applications in **Simulink**
- You can **reuse your existing expertise and models** to mitigate the risk of migration to SOA applications

To learn more, visit the SOA, AUTOSAR & DDS Blockset pages

What Is SOA? Search MathWorks.com

Trial software Contact sales


Model service-oriented architectures (SOA) in Simulink

Service-oriented architecture (SOA) is a software architecture based on the concept that a system consists of a set of services in which one service may use another, and applications use one or more of the services based on their need. SOA promotes a loosely coupled component-based approach using middleware for service-oriented communication.

SOA is used in multiple industry standards, including:

- AUTOSAR: Engineers in the automotive industry have been increasingly using SOA when designing systems for highly autonomous driving applications. The AUTOSAR Adaptive Platform was developed by the AUTOSAR organization and is based on SOA. The AUTOSAR Adaptive Platform provides flexibility and scalability in processing distribution and compute resource allocations. Therefore, you can securely update and upgrade adaptive ECU software even after its release.
- ROS: Many robotics applications use Robot Operating System (ROS), a robotics middleware that follows SOA methodology. It serves as a framework for communication between the components necessary to run the software.
- DDS: Data Distribution Services (DDS) uses SOA methodology, and directly addresses publish and subscribe communications for real-time and embedded systems. DDS addresses the needs of applications that require real-time data exchange in industries like aerospace and defense, automotive, and robotics.

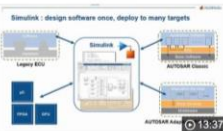
You can use Simulink to model and simulate software based on SOA that runs in different applications.




Free Power Electronics Control Design white paper

Hardware-in-the-Loop Testing for Power Electronics Control Design

[Download now](#)




Model Service-oriented Architectures in Simulink



How to Model Software Services with Simulink Functions

www.mathworks.com/discovery/soa.html



AUTOSAR Blockset

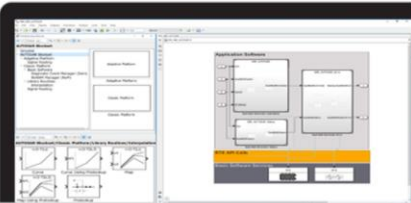
Design and Simulate AUTOSAR Software

[Download a free trial](#)

AUTOSAR Blockset provides an AUTOSAR dictionary and blocks for developing Classic and Adaptive AUTOSAR software using Simulink® models. You can define AUTOSAR software component properties, interfaces, and datatypes, and map them to existing Simulink models using the AUTOSAR editor. Alternatively, the blockset provides an application interface that lets you automatically generate new Simulink models for AUTOSAR by importing software component and composition descriptions from AUTOSAR XML files.

AUTOSAR Blockset provides blocks and constructs for AUTOSAR library routines and Basic Software (BSW) services, including NVRAM and Diagnostics. By simulating the BSW services together with your application software model, you can verify your AUTOSAR ECU software without leaving Simulink.

AUTOSAR Blockset supports C and C++ production code generation and AUTOSAR XML file export (with Embedded Coder®). It is qualified for use with the ISO 26262 standard (with IEC Certification Kit).



www.mathworks.com/products/autosar.html



DDS Blockset

Design and simulate DDS applications

[Request a trial](#)

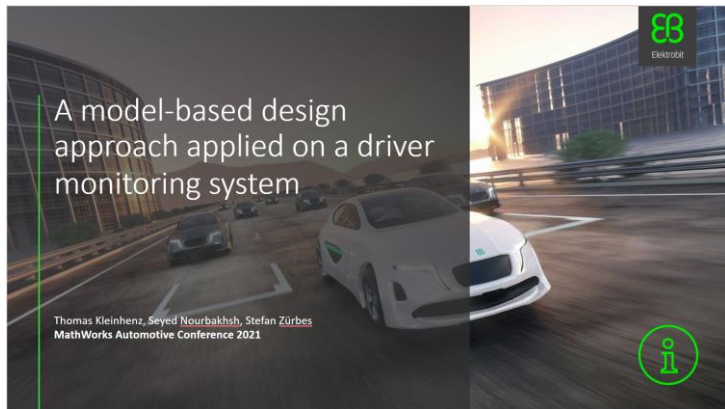
DDS Blockset provides apps and blocks for modeling and simulating software applications that publish or subscribe to Data Distribution Service (DDS) middleware. The blockset includes a DDS dictionary that lets you manage, create, and edit your DDS definitions in Simulink®. You can import DDS specifications as XML files to create a skeleton Simulink model as a starting point for developing algorithms for DDS applications.

DDS Blockset provides blocks for publishing and subscribing samples to DDS, including their corresponding Quality of Service (QoS). It fully

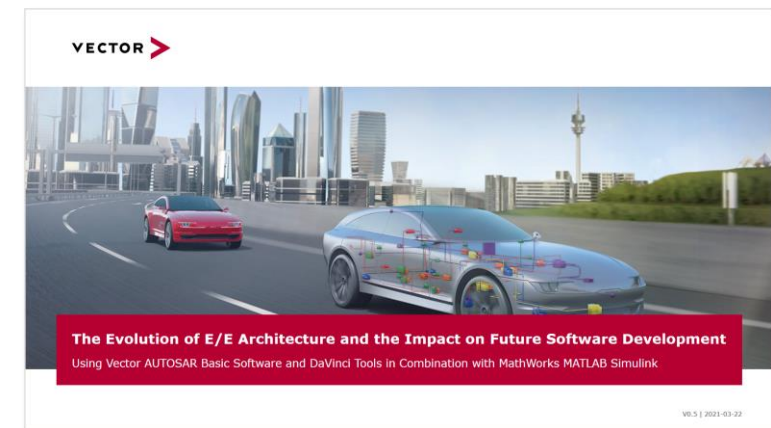


www.mathworks.com/products/dds.html

Learn more by visiting demo showcase and partner talks at the MATLAB Expo



Thomas Kleinhenz - Elektrobit Automotive - Developing a Driver Monitoring System Using Model-Based Design



Francisco González - Vector Informatik - Developing a Driver Monitoring System Using Model-Based Design

Get in touch!

Please, reach out to us! We'll be happy to work with you and embrace your **SOA project** together.

Poll - Are you interested in more information?

- a. Yes, I have some questions and would like to talk
- b. Yes, I would like to schedule follow up session on this topic with my team
- c. Not at this time

MATLAB EXPO

2021

Thank you



© 2021 The MathWorks, Inc. MATLAB and Simulink are registered trademarks of The MathWorks, Inc. See [mathworks.com/trademarks](https://www.mathworks.com/trademarks) for a list of additional trademarks. Other product or brand names may be trademarks or registered trademarks of their respective holders.