MATLAB EXPO 2021

使用Simulink设计和部署面向服务的架构

龚小平， MathWorks

MathWorks

内容

- 软件定义汽车时代的行业挑战
- Simulink开发面向服务的架构(SOA)
- Simulink对SOA行业标准的支持
  - Simulink对AUTOSAR Adaptive的支持
  - Simulink对数据分发服务 (DDS)的支持
- 总结及如何获取更多相关信息

内容

- 软件定义汽车时代的行业挑战
- Simulink开发面向服务的架构(SOA)
- Simulink对SOA行业标准的支持
  - Simulink对AUTOSAR Adaptive的支持
  - Simulink对数据分发服务 (DDS)的支持
- 总结及如何获取更多相关信息

软件时代来临

From the news...

Adds 3,000 New Tech Jobs to Boost Virtual Testing and Software Expertise

Continental and Amazon Web Services Create Platform for Automotive Software

Boss Warns the Troops: We Don't Want to End Up Like Nokia

BOSCH Consolidates All Automotive Software And Electronics Into New Division

DAIMLER to Cut Out Suppliers to Fund Software Hiring Spree

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

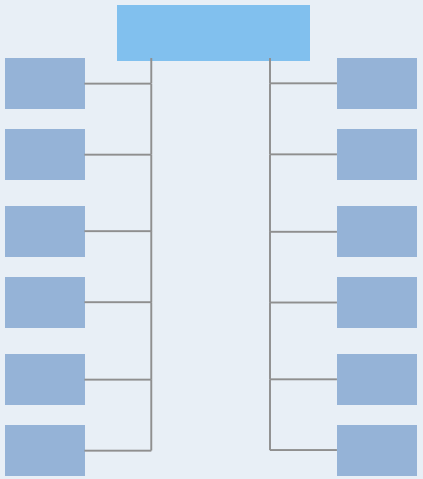
汽车行业趋势

软件定义汽车

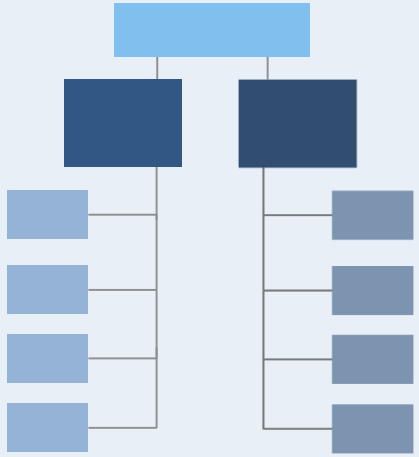
关键赋能因素:

1. 算力向高性能车载计算机集中
2. 新的电子电气和软件架构设计理念

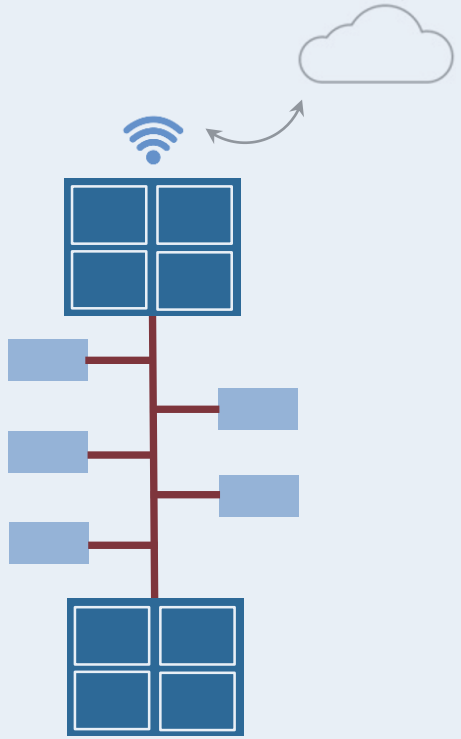
电子电气架构进化



- ~100 ECUs
- 1个功能对应1个 ECU
- 繁复的线束



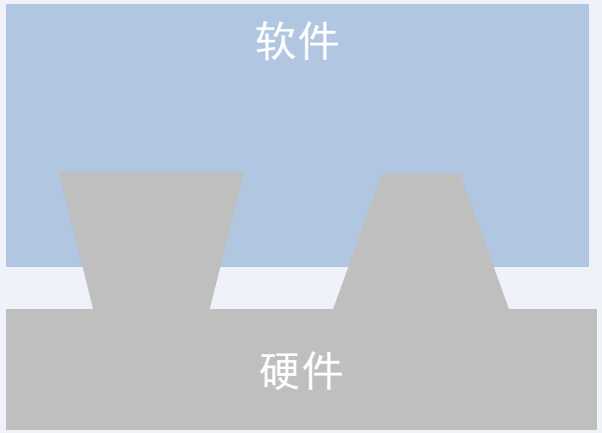
- 域控制器
- 多核 ECUs
- 网关和网络



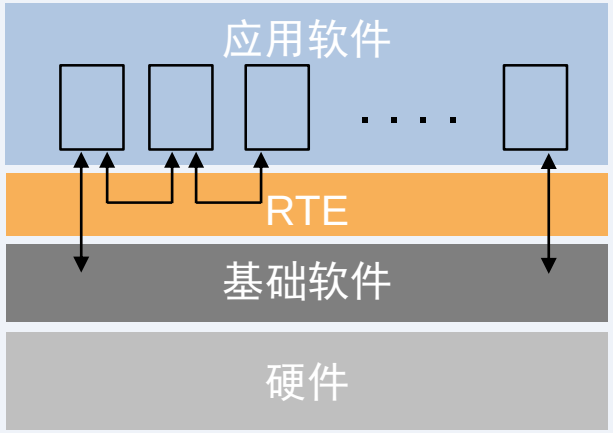
- 车辆/区域控制器
- 多核异构（CPU/GPU/FPGA）
- 高速以太网

分布 → 集中 → 高性能高连接

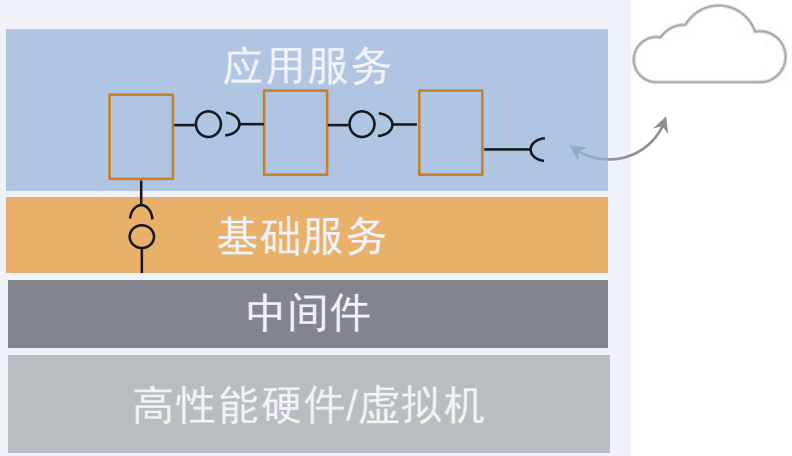
软件架构演变



- 软硬件高度耦合
- 没有标准 APIs
- 软件复用程度低



- 静态软件组件分配
- 基于信号通讯
- 硬件抽象
- 整体升级



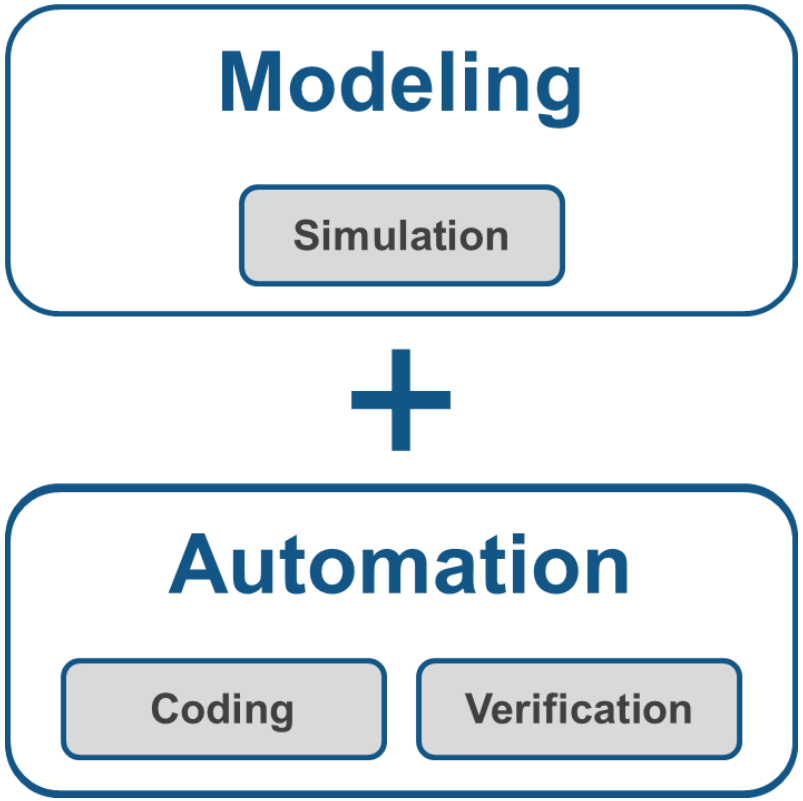
- 动态服务发现
- 面向服务的通讯
- 更高级别的硬件抽象
- 选择性升级(OTA)

软硬耦合 → 基于组件 → 面向服务

关键挑战

- 面向服务的应用对于汽车行业仍是个新事物
- 已有架构和面向服务的架构将长期共存
- 知识经验、工作流程和软件资产的重用

解决方案 – 面向SOA的基于模型设计



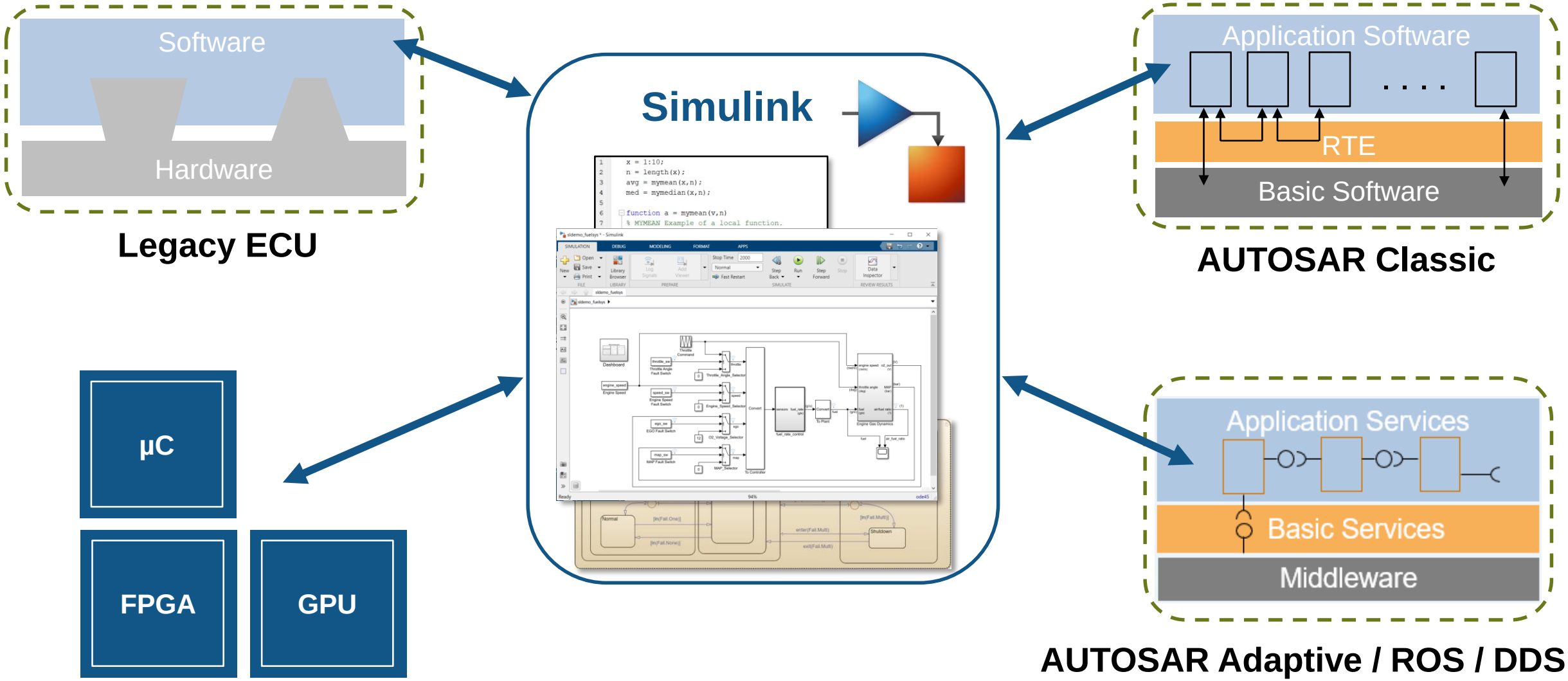
设计和仿真SOA应用

- 图形化的直观协同软件架构设计环境
- Simulink提供语义支持建模和仿真SOA应用

加强自动代码生成和验证支持

- 行业标准支持
- 异构目标硬件平台
- 集成人工智能算法

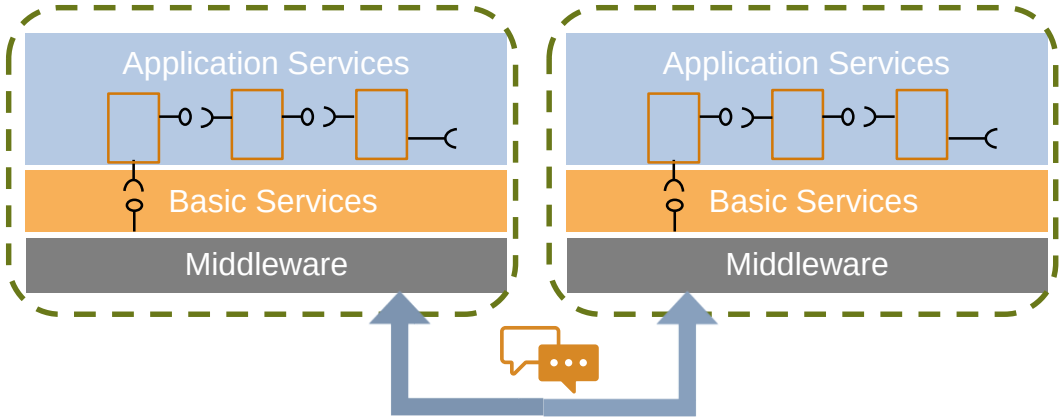
关键概念 – 以算法为中心


Software
Hardware
Legacy ECU
Simulink
Application Software
RTE
Basic Software
AUTOSAR Classic
µC
FPGA
GPU
Application Services
Basic Services
Middleware
AUTOSAR Adaptive / ROS / DDS

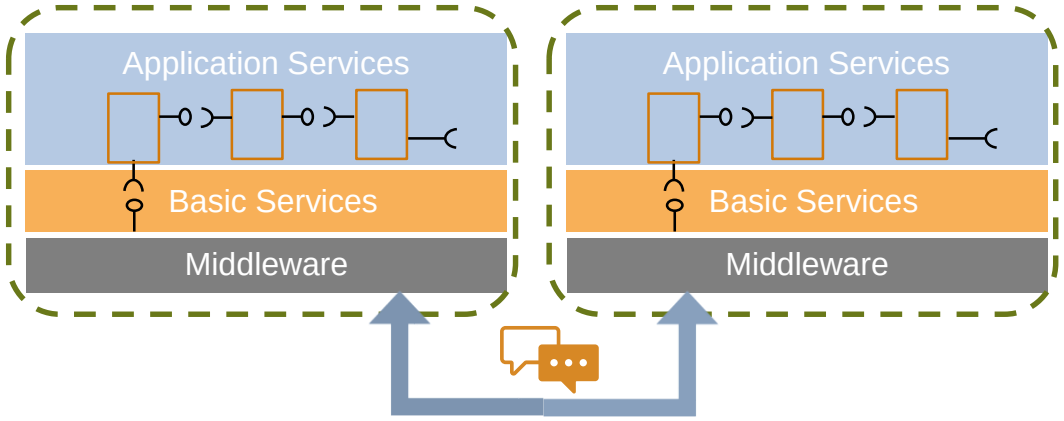
内容

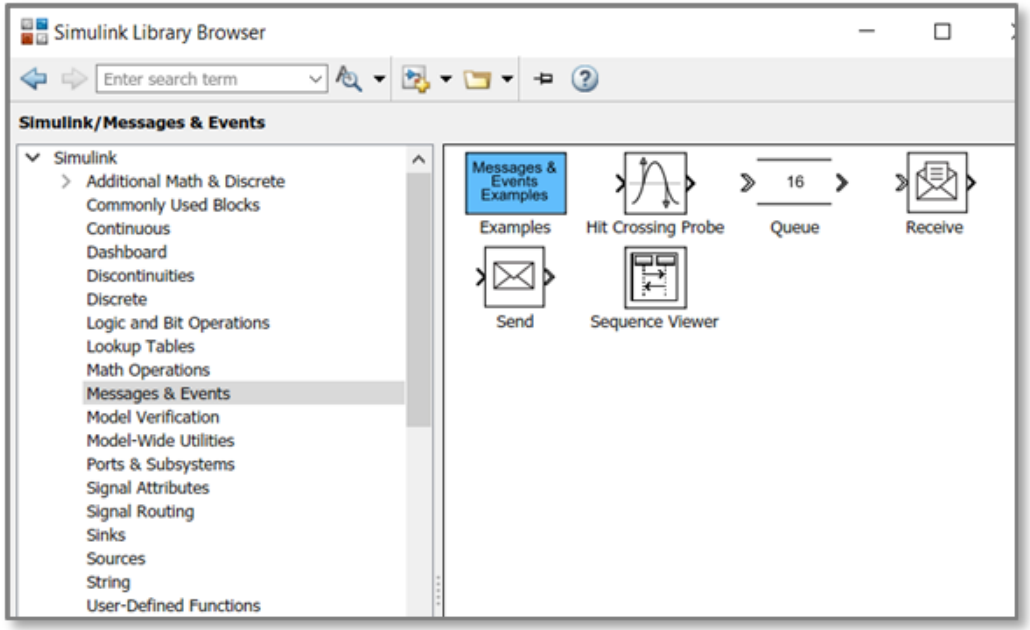
- 软件定义汽车时代的行业挑战
- Simulink开发面向服务的架构(SOA)
- Simulink对SOA行业标准的支持
  - Simulink对AUTOSAR Adaptive的支持
  - Simulink对数据分发服务 (DDS)的支持
- 总结及如何获取更多相关信息

面向服务的通讯建模支持 – Simulink Messages

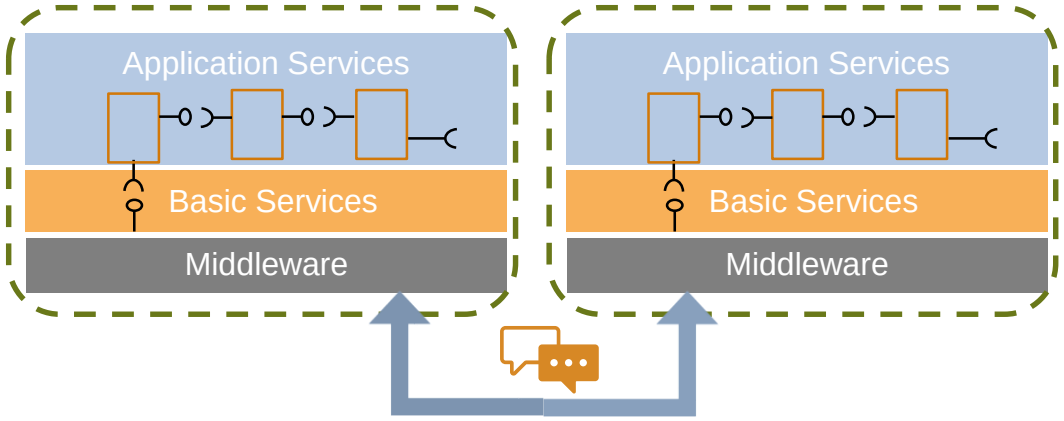


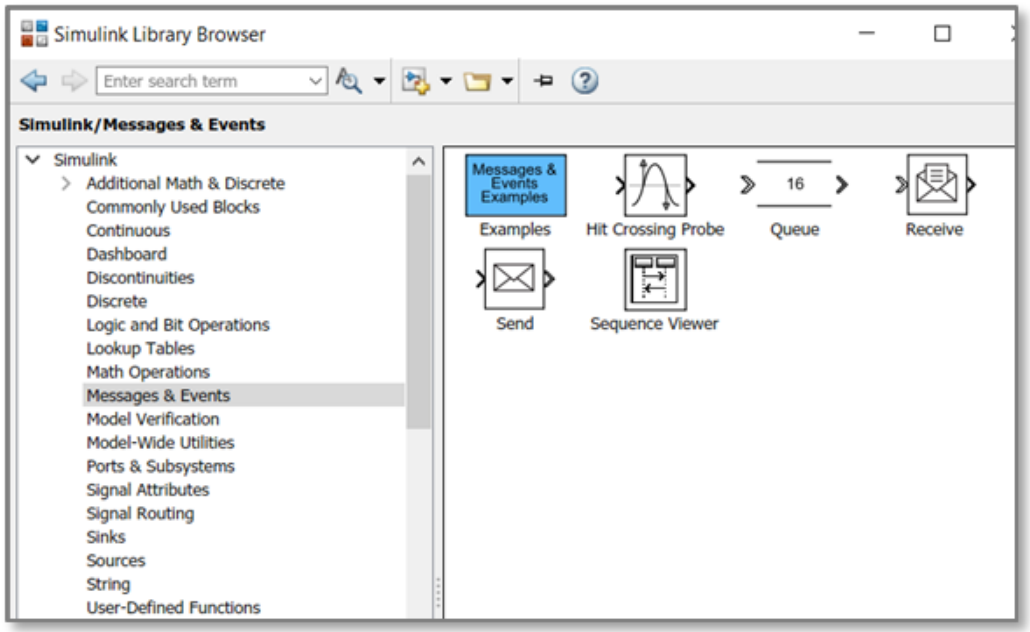
面向服务的通讯建模支持 – Simulink Messages

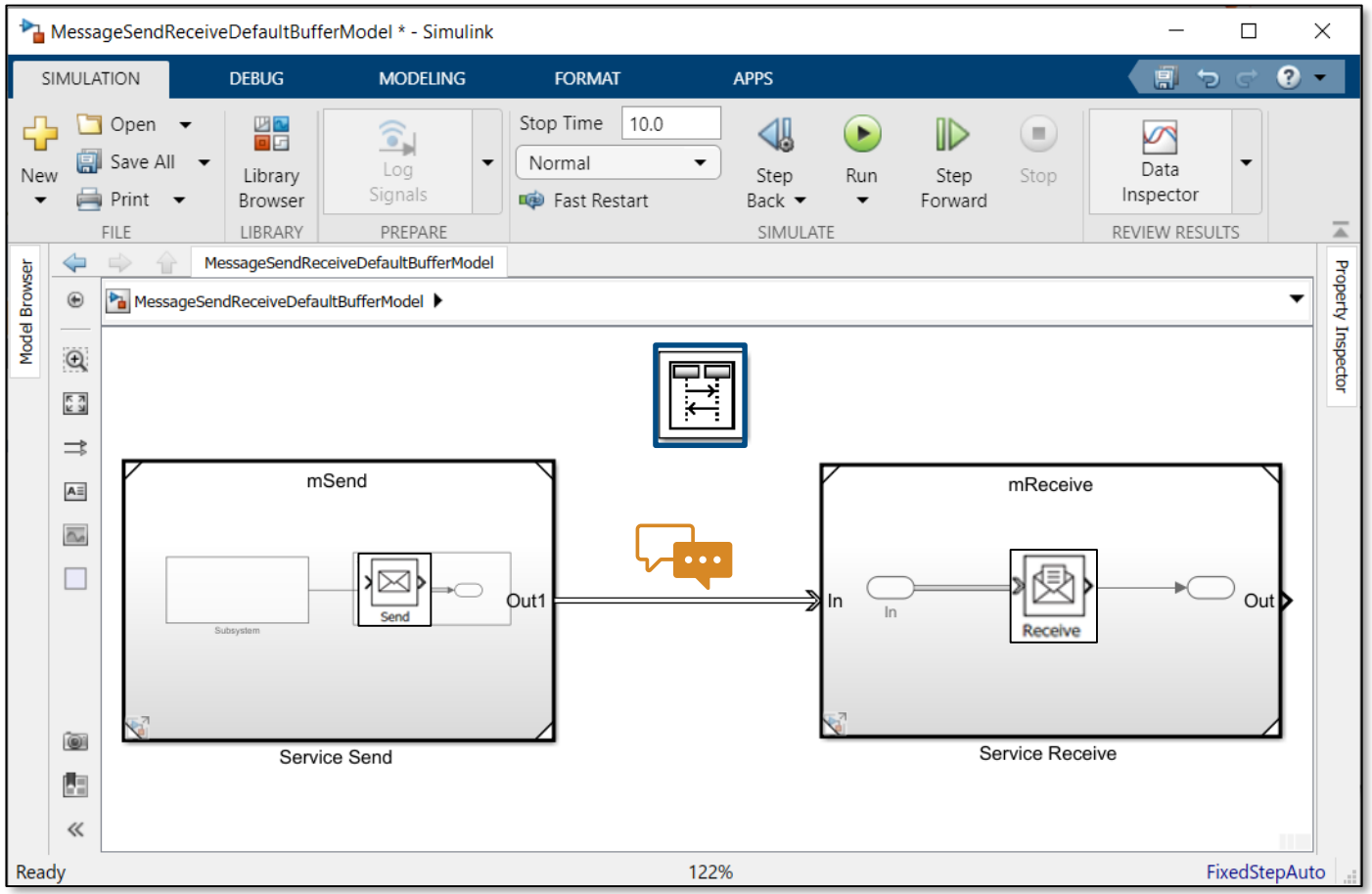




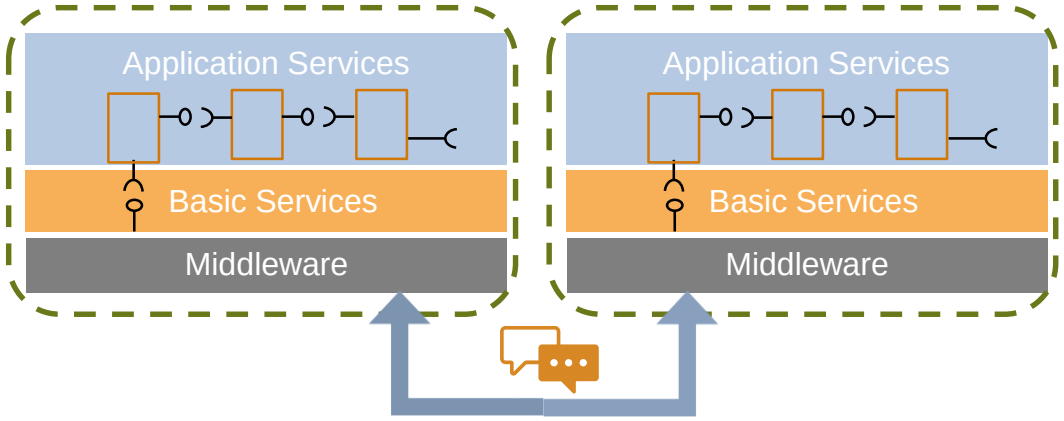
面向服务的通讯建模支持 – Simulink Messages

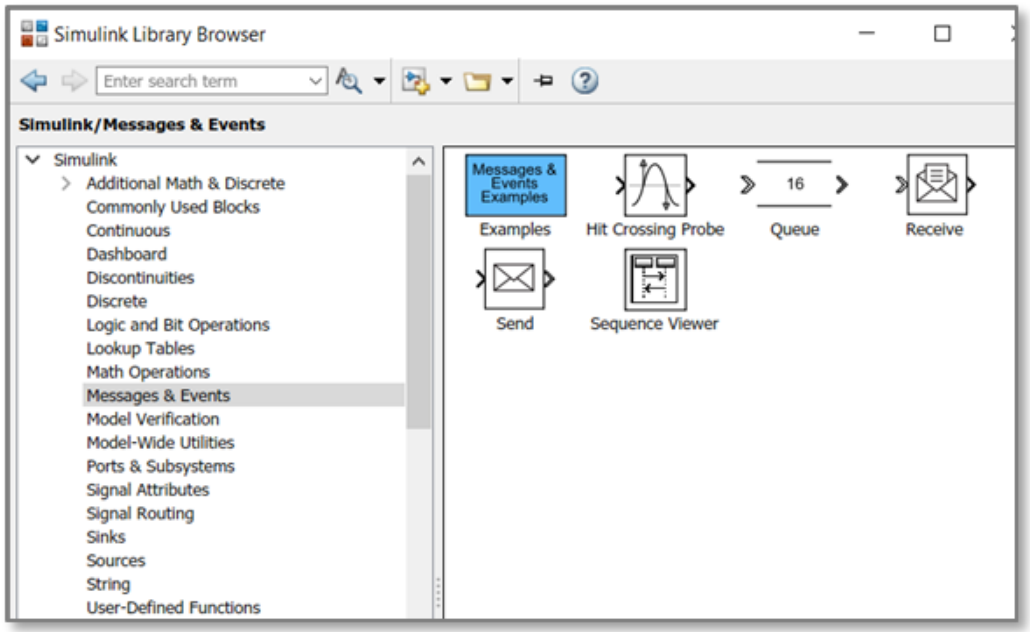


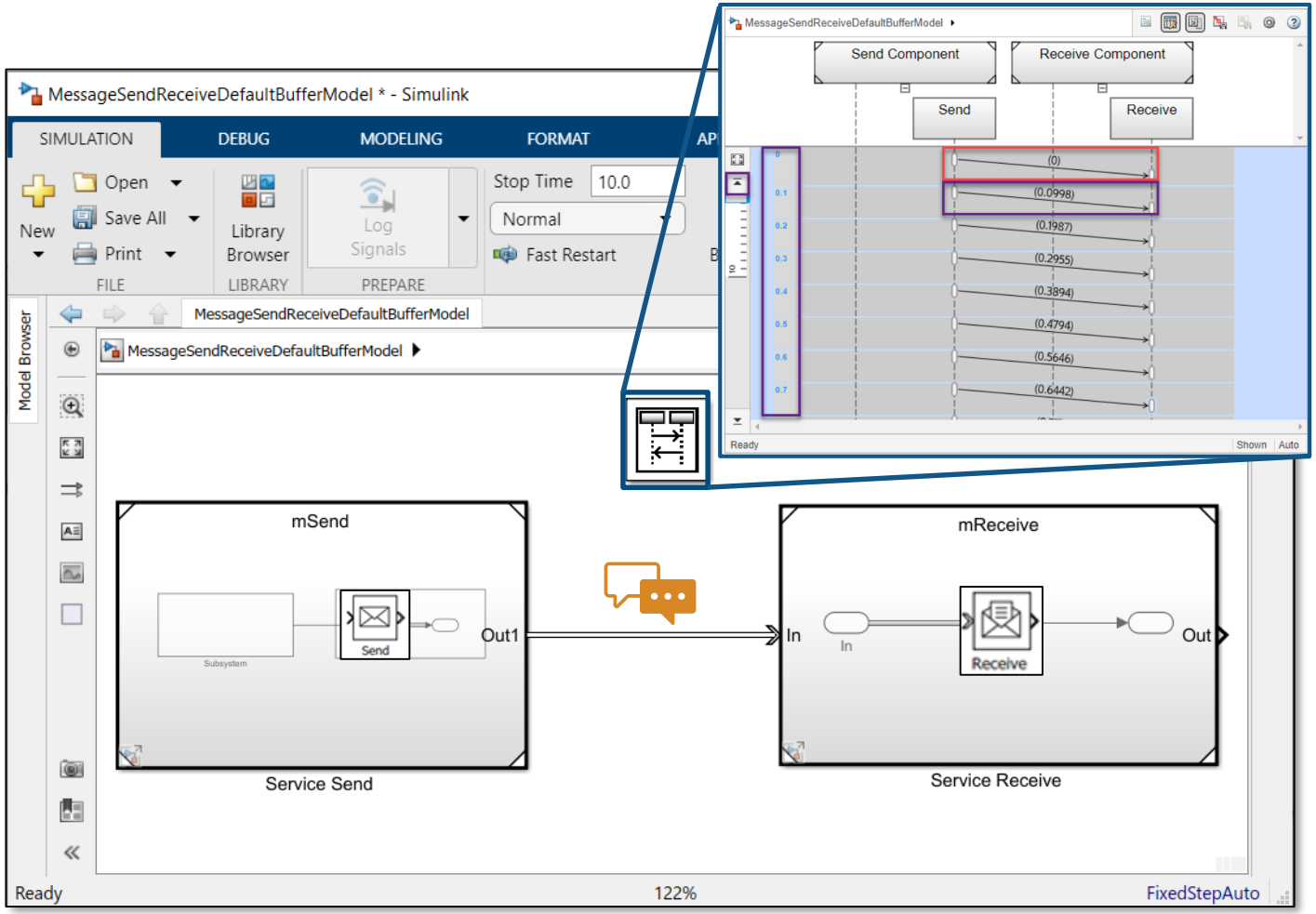




面向服务的通讯建模支持 – Simulink Messages



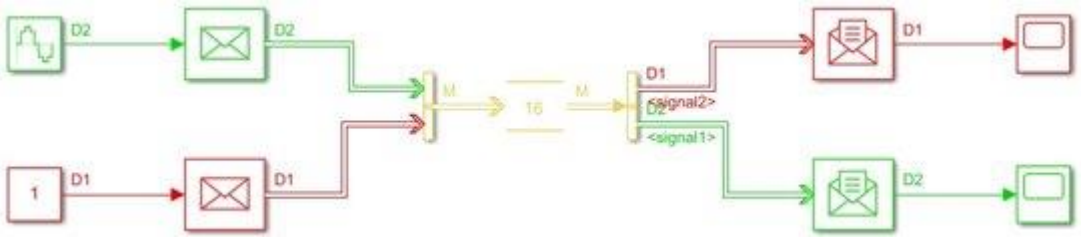




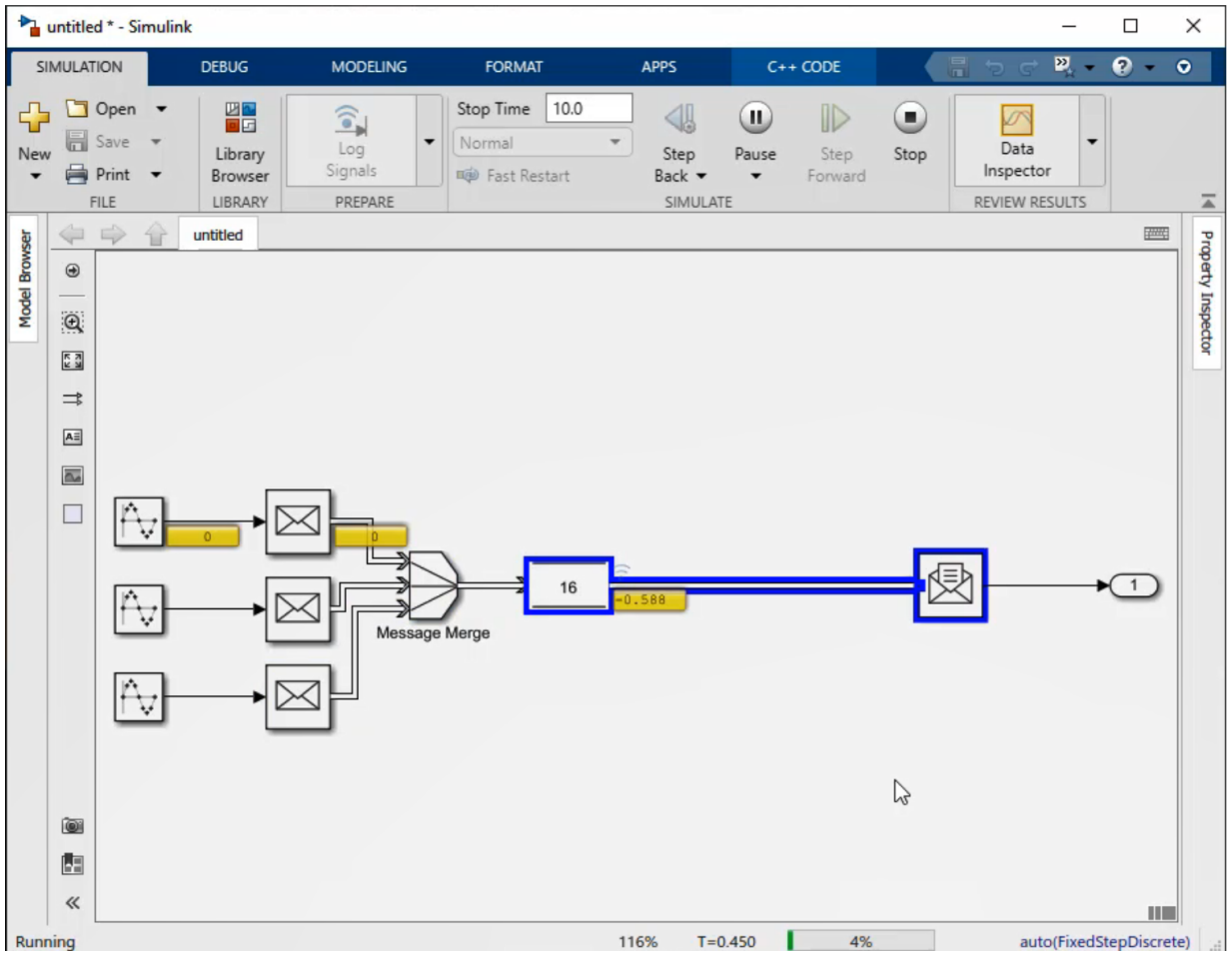
使用消息发送和接收模块建模面向服务的通讯

消息建模的新功能

- 消息线可以合并为非虚拟或虚拟总线
- 将多个消息线合并为一个消息线



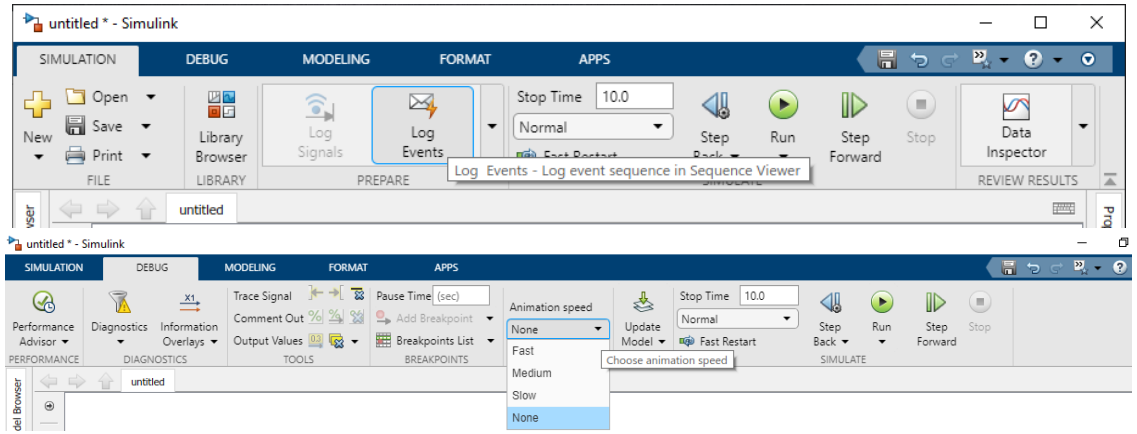
R2020b

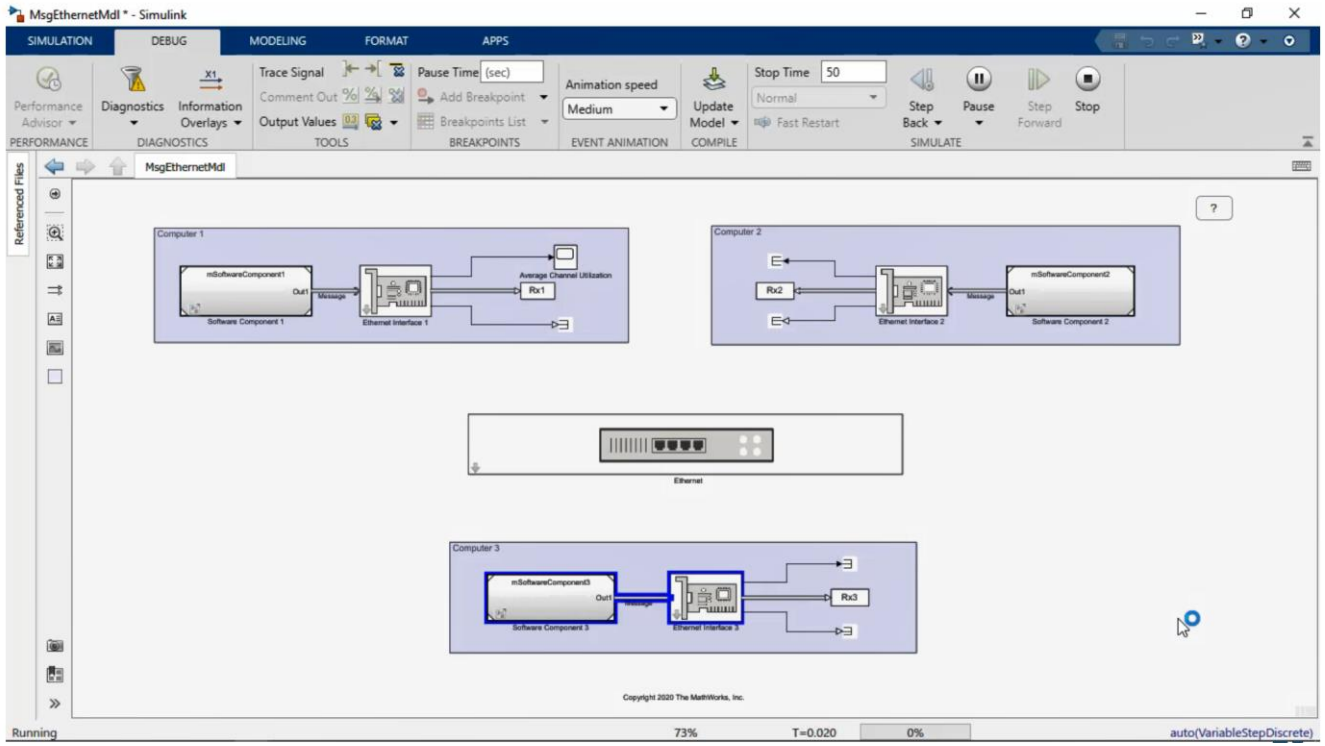


R2021a

理解复杂系统 – 基于事件的记录和动画

- 查看模型中涉及到以下模块的事件
  - Function-Call 子系统
  - Simulink 消息
  - Simulink 函数
- 用于
  - 事件记录
  - 事件动画
  - 时序观测工具





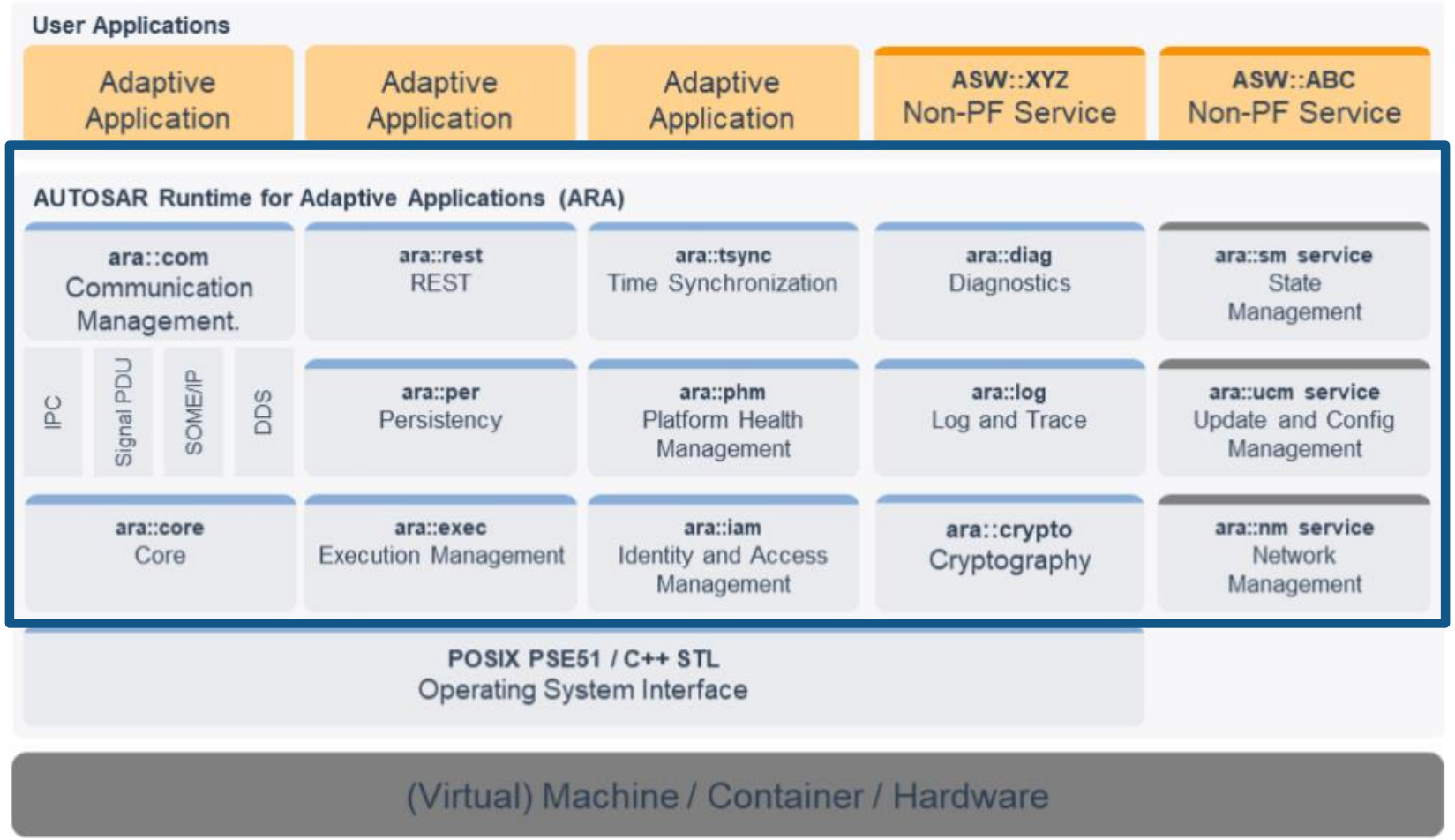
内容

- 软件定义汽车时代的行业挑战
- Simulink开发面向服务的架构(SOA)
- Simulink对SOA行业标准的支持
  - Simulink对AUTOSAR Adaptive的支持
  - Simulink对数据分发服务 (DDS)的支持
- 总结及如何获取更多相关信息

AUTOSAR Adaptive 概述

AUTOSAR Adaptive

- 服务和接口(ARA)

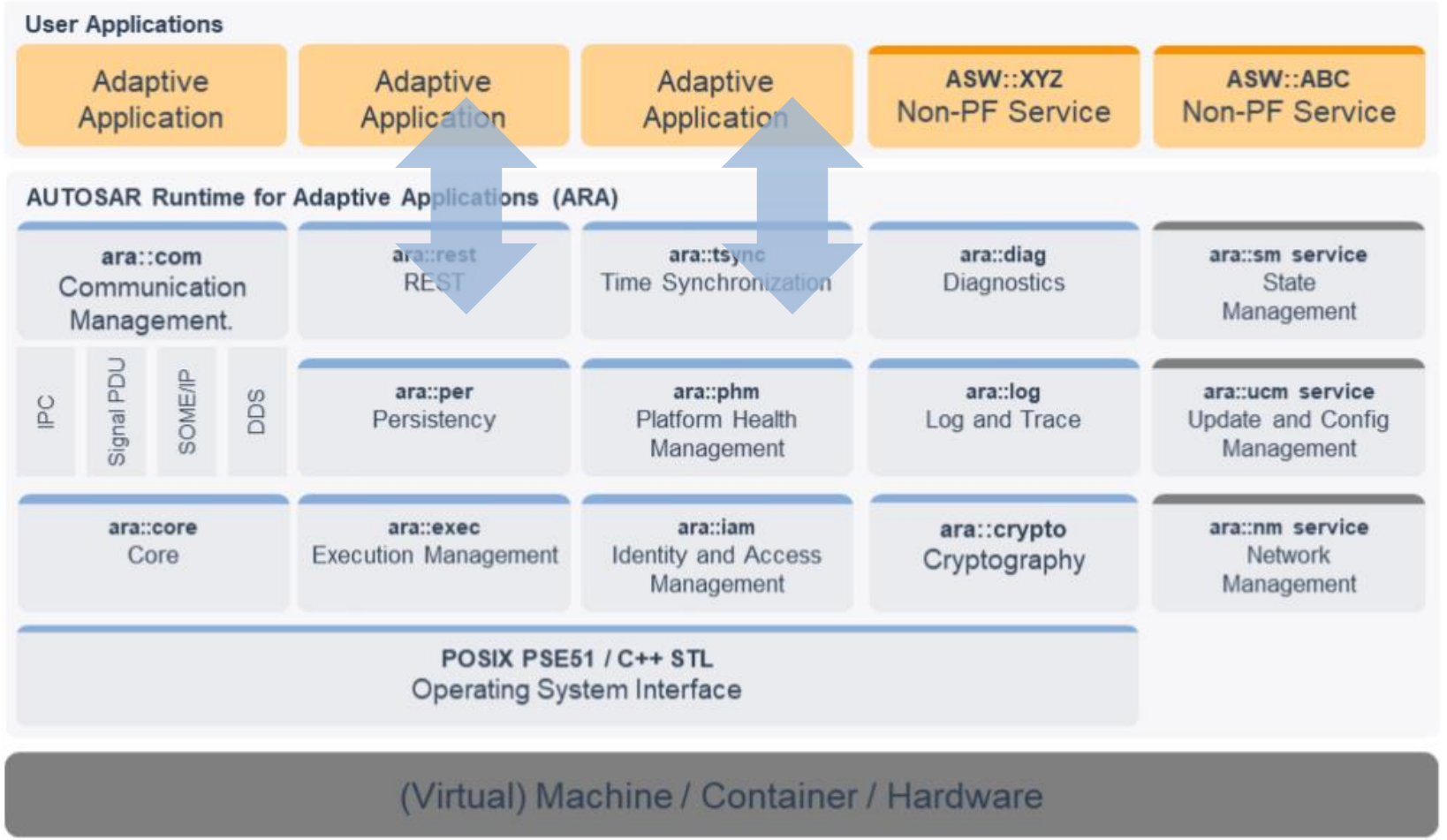


Source: www.autosar.org

AUTOSAR Adaptive 概述

AUTOSAR Adaptive

- 服务和接口(ARA)
- 客户端（应用）和服务器的运行时动态链接

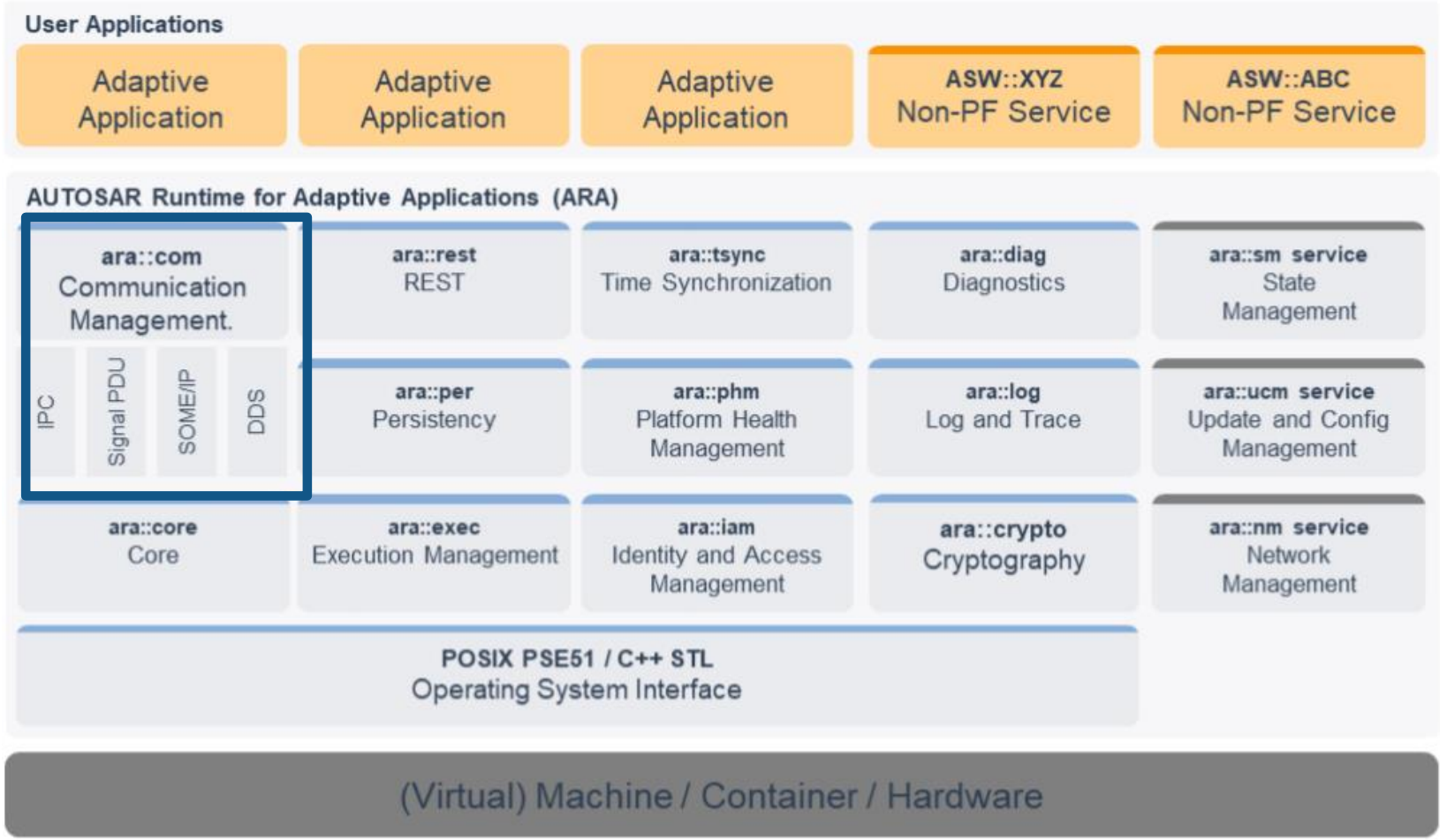


Source: www.autosar.org

AUTOSAR Adaptive 概述

AUTOSAR Adaptive

- 服务和接口(ARA)
- 客户端（应用）和服务器的运行时动态链接
- 通讯中间件：SOME/IP 和 DDS



Source: www.autosar.org

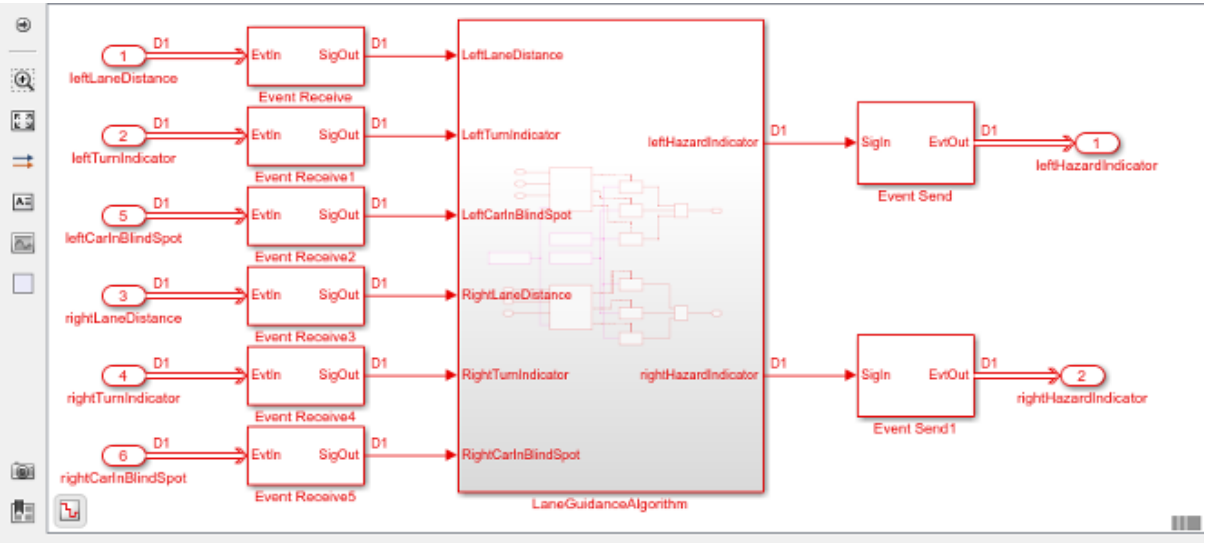
AUTOSAR Adaptive 支持 – AUTOSAR Blockset

AUTOSAR Adaptive平台实现了汽车行业Adaptive应用的AUTOSAR Runtime

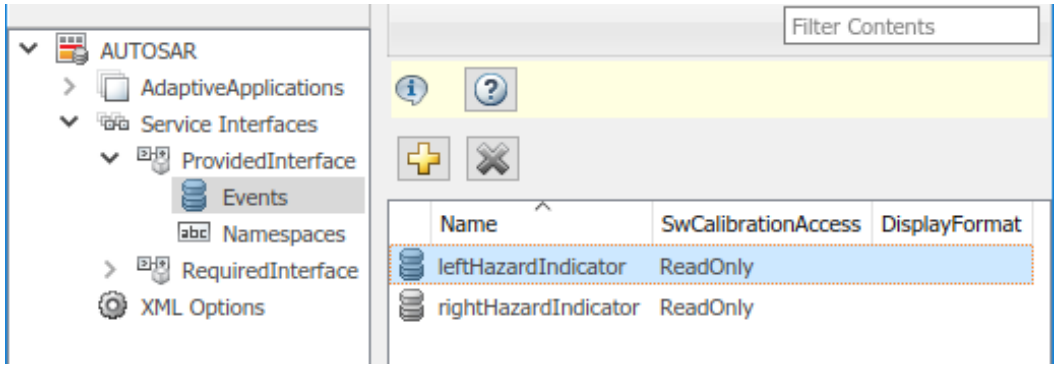
在Simulink中对AUTOSAR Adaptive应用进行建模、仿真、测试和生成代码

使用Simulink开发AUTOSAR Adaptive

- AUTOSAR Adaptive的服务通讯机制:
  - Events
  - Methods
  - Fields



- Simulink中Events可以采用消息建模及配置生成代码



AUTOSAR Adaptive 软件架构

Radar Service Interface

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

Hazard Service Interface

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

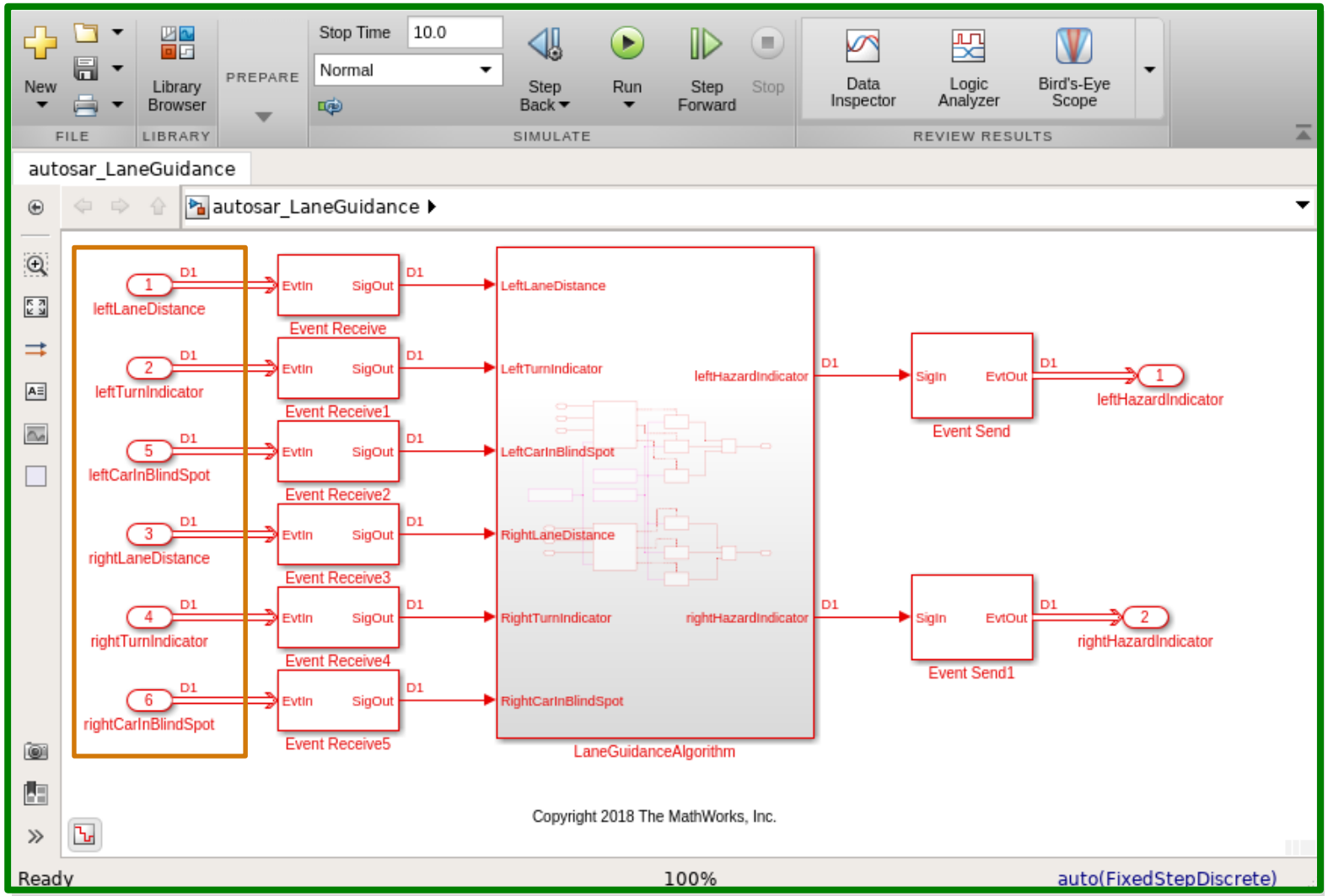
AUTOSAR Adaptive 应用建模

AUTOSAR

Adaptive Application

RequiredPort

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



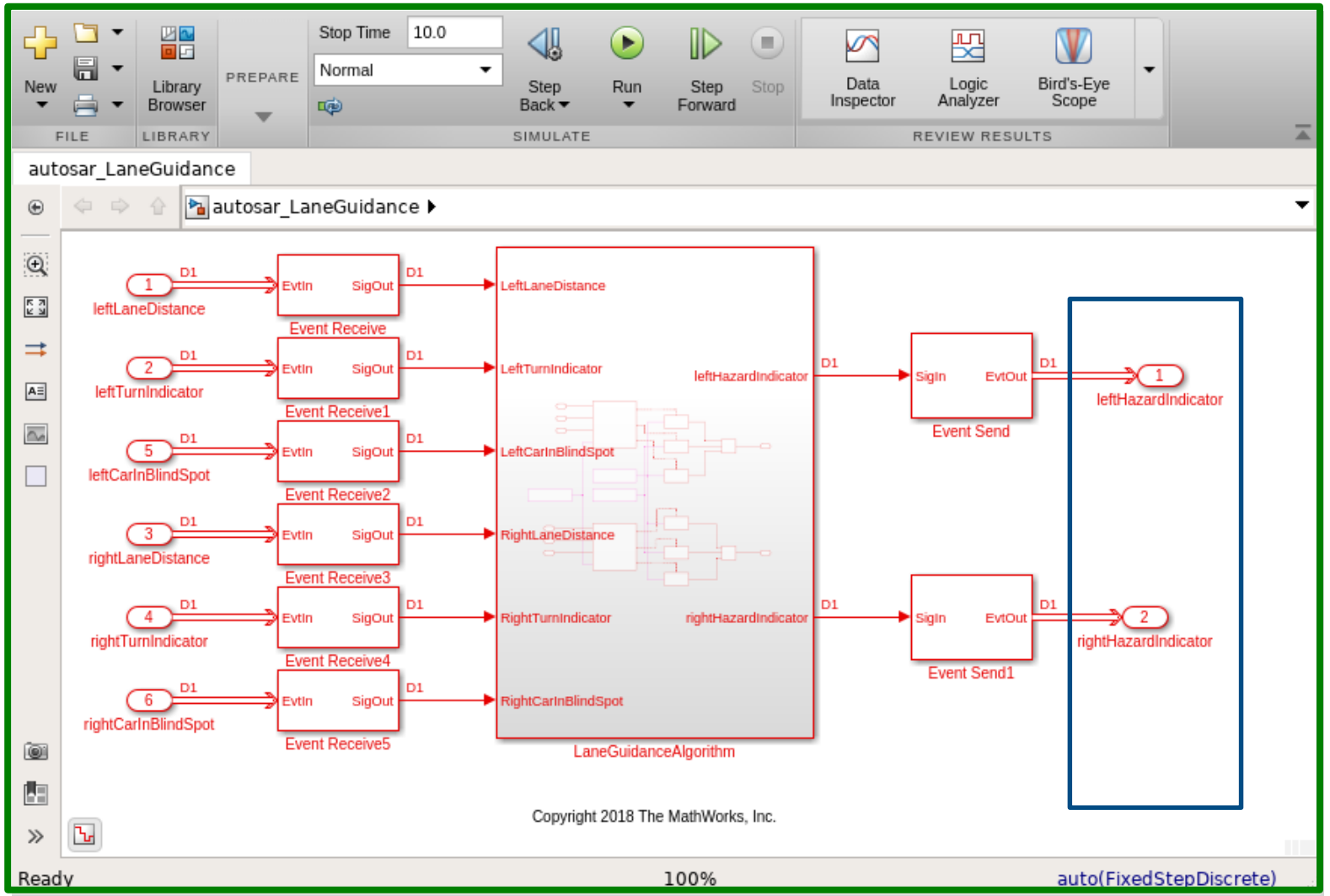
AUTOSAR Adaptive 应用建模

AUTOSAR

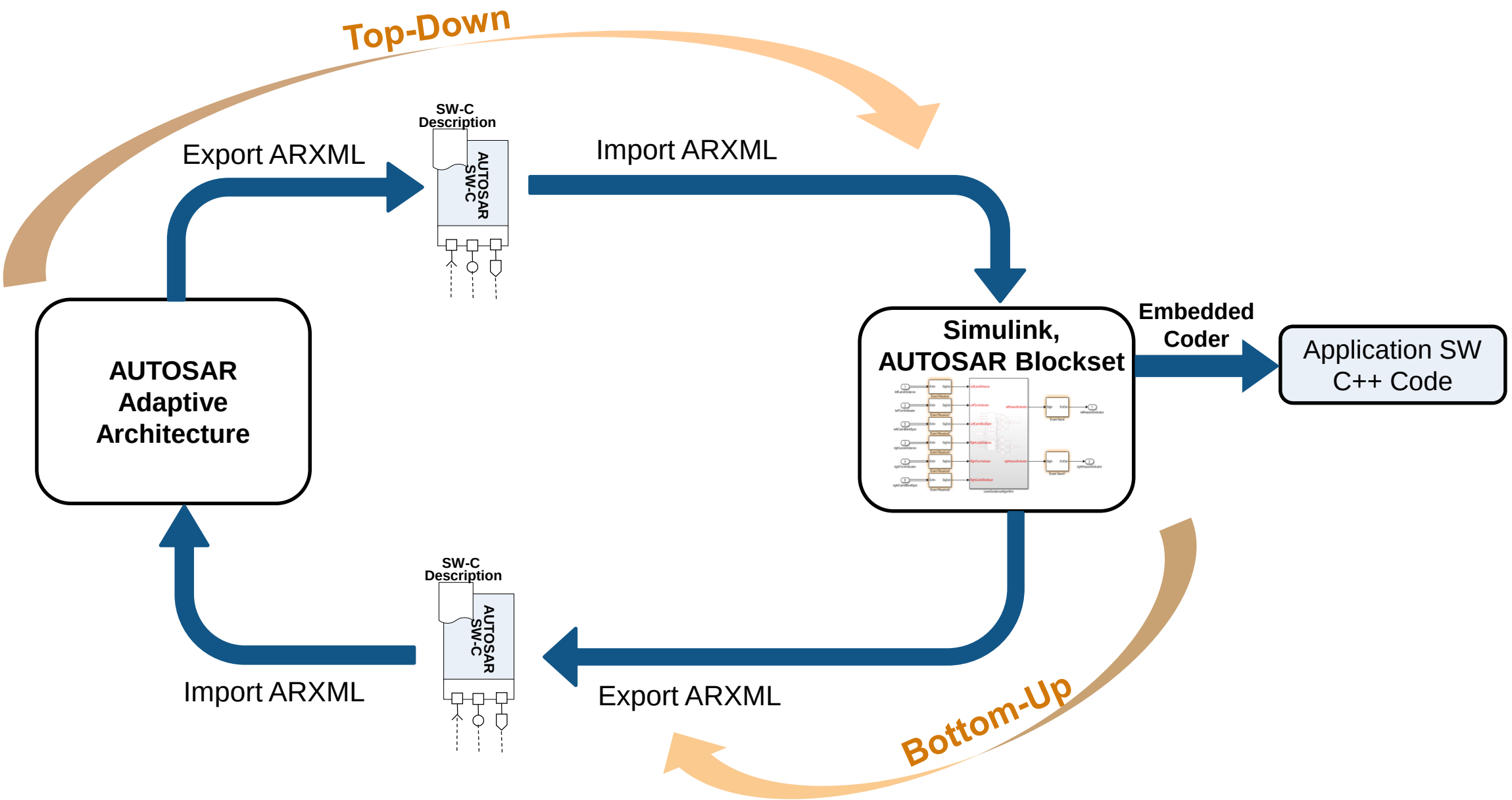
Adaptive Application

ProvidedPort

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

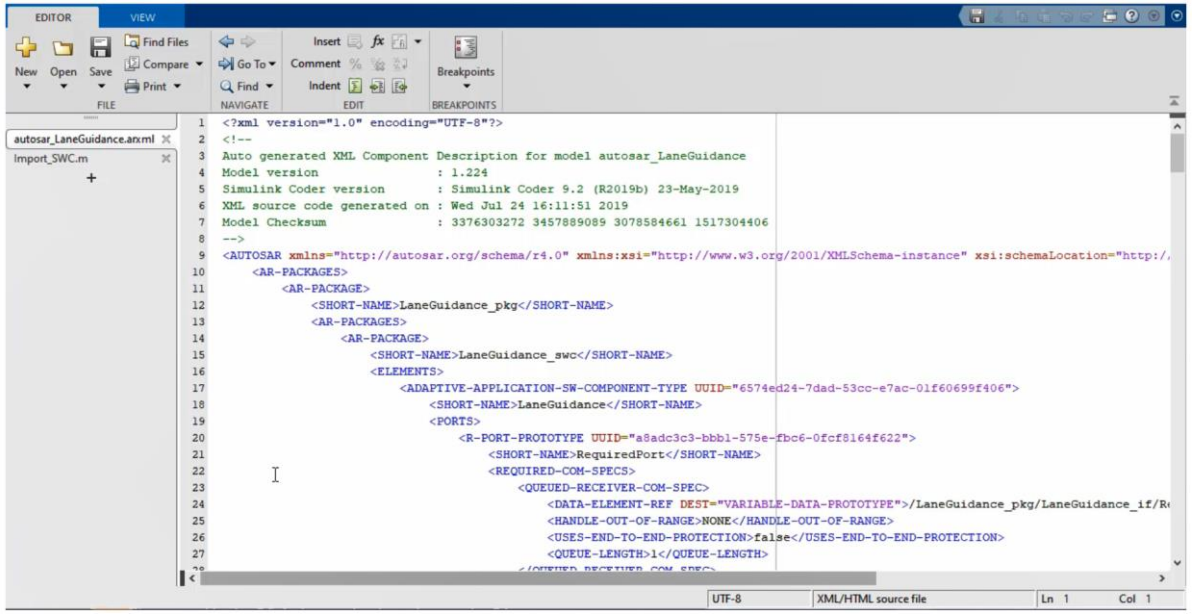


AUTOSAR Adaptive 工作流程



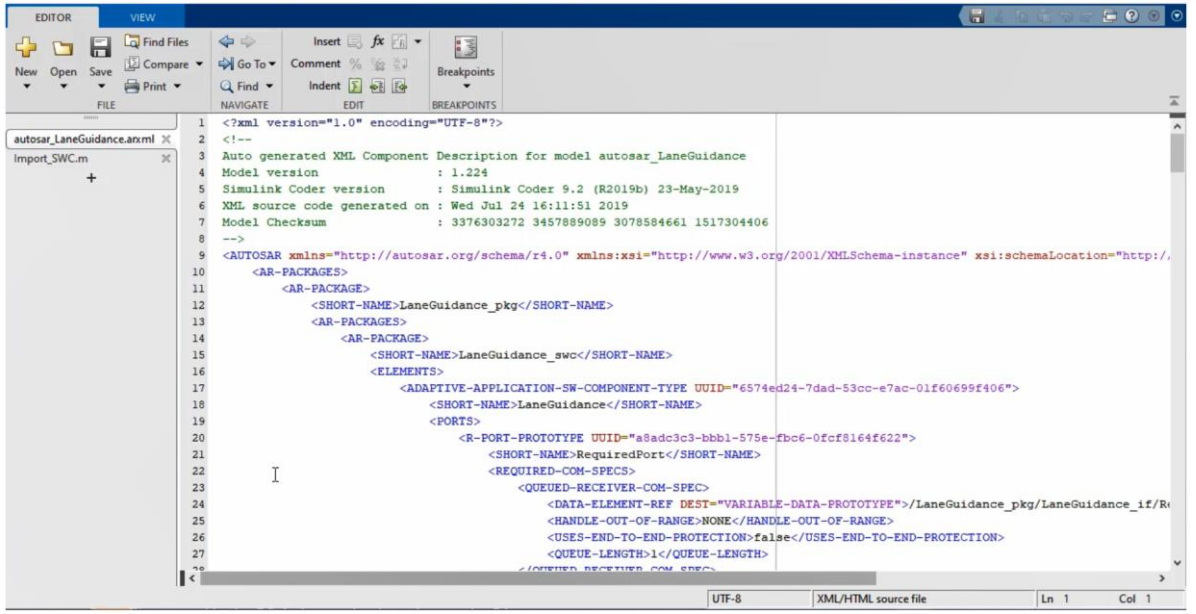
AUTOSAR Adaptive 开发步骤

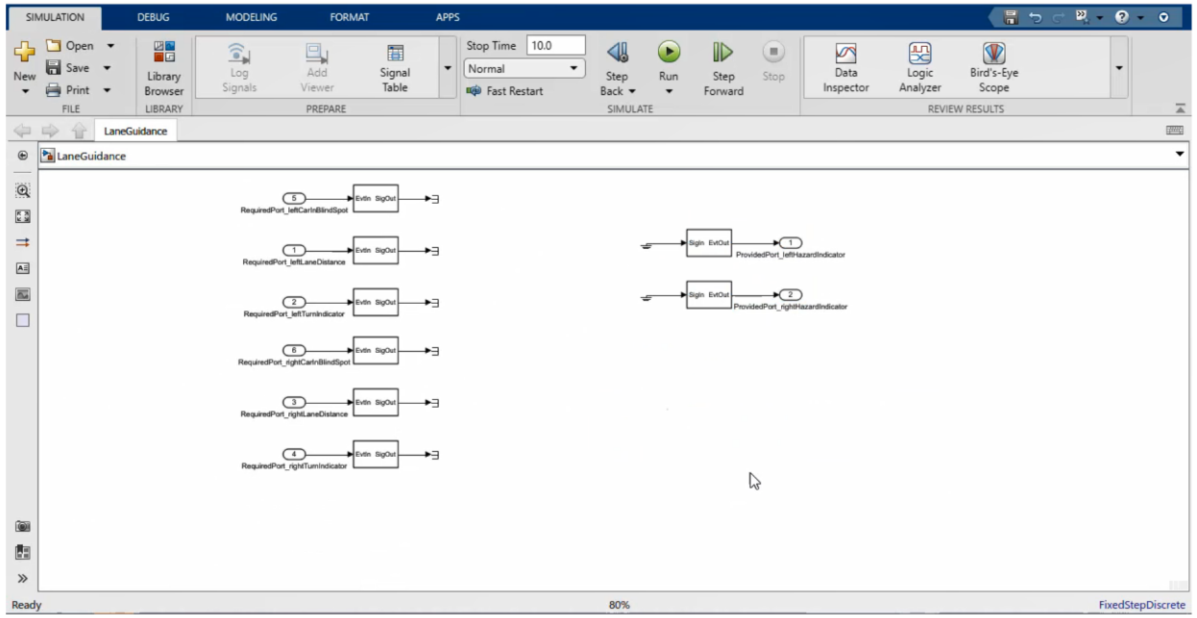
- 导入ARXML创建模型框架



AUTOSAR Adaptive 开发步骤

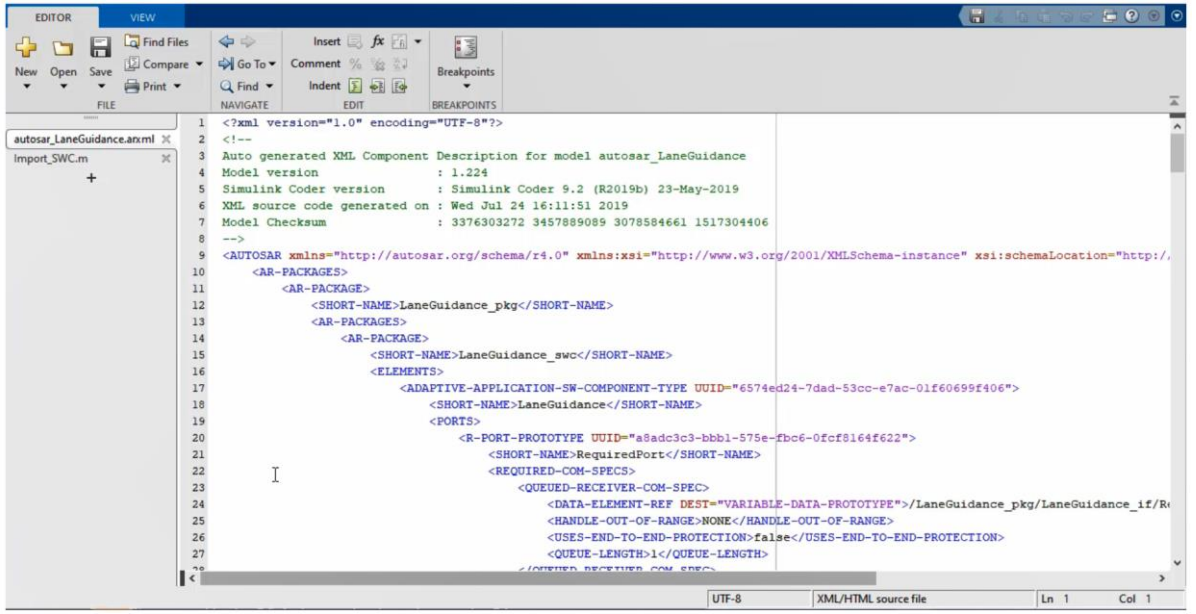
- 导入ARXML创建模型框架

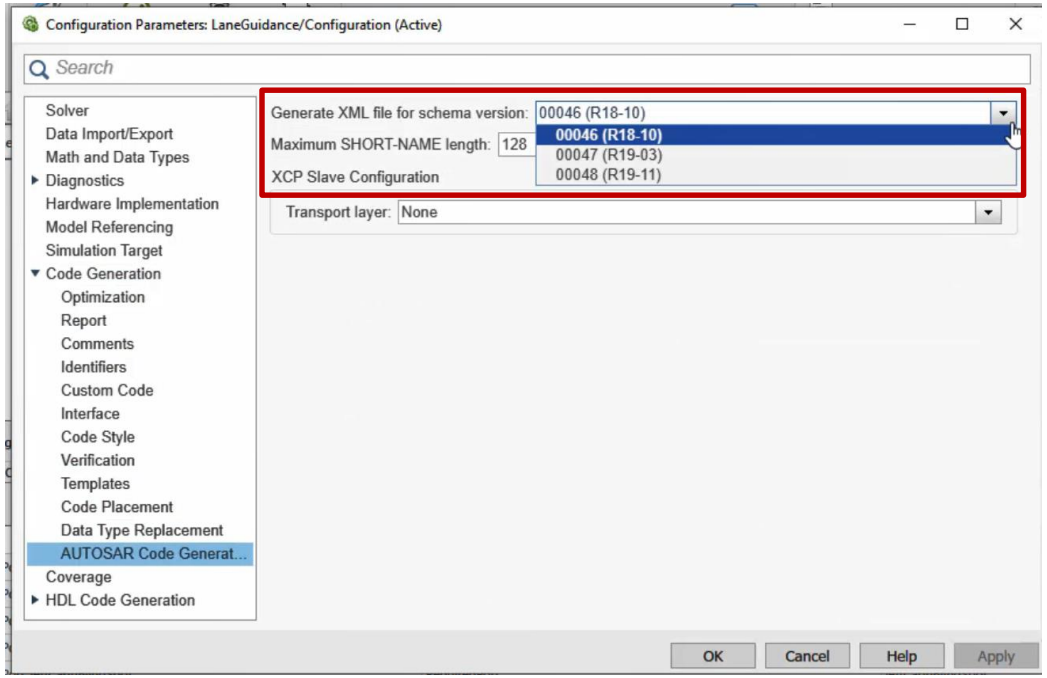


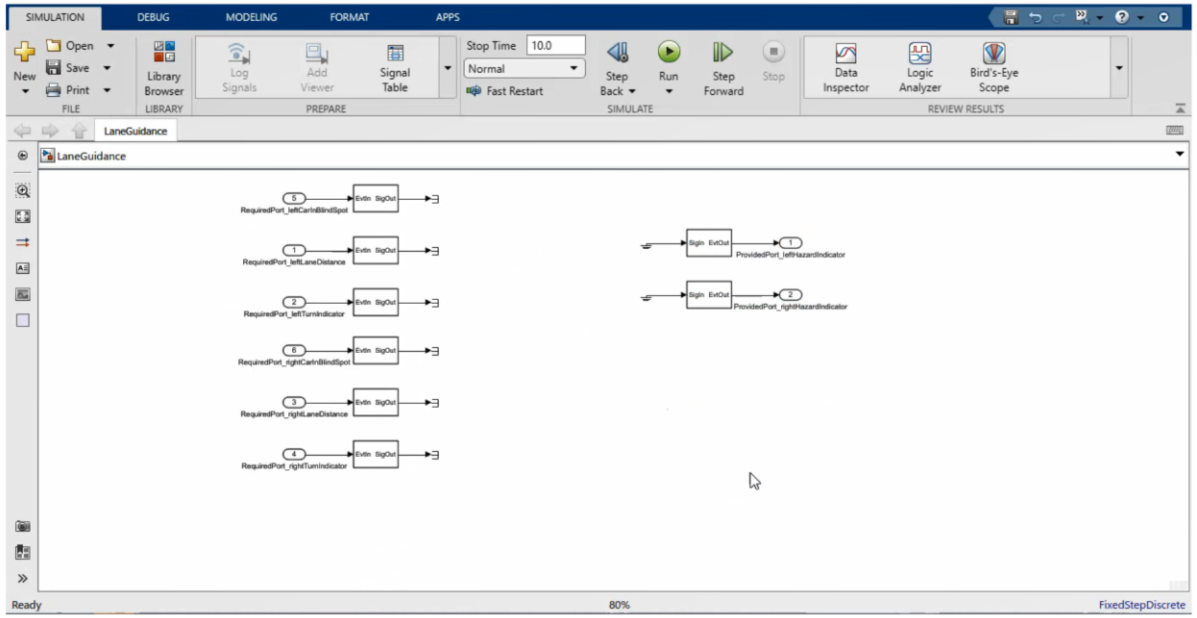


AUTOSAR Adaptive 开发步骤

- 导入ARXML创建模型框架

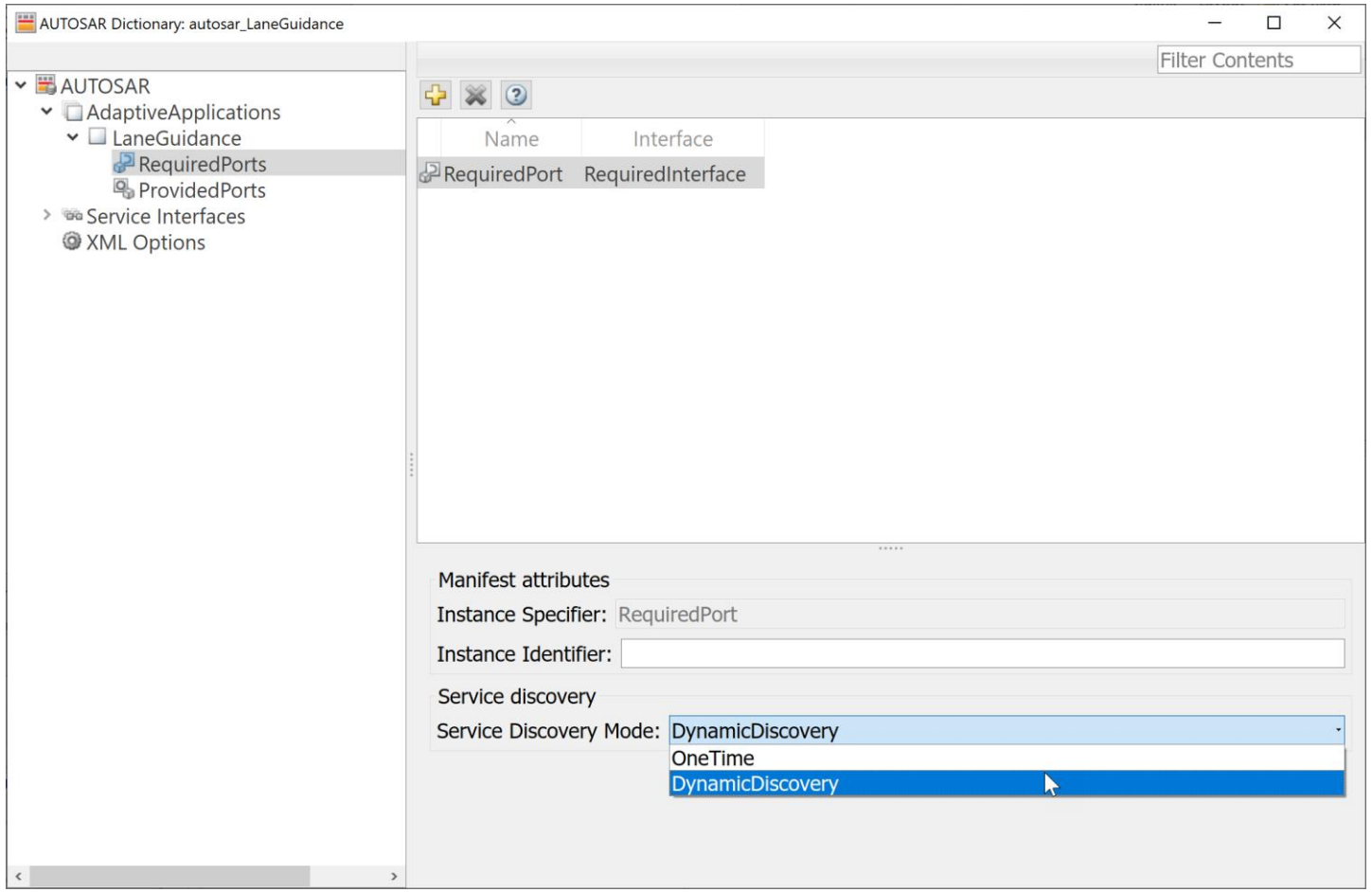






AUTOSAR Adaptive 开发步骤

- 导入ARXML创建模型框架
- 配置服务发现机制

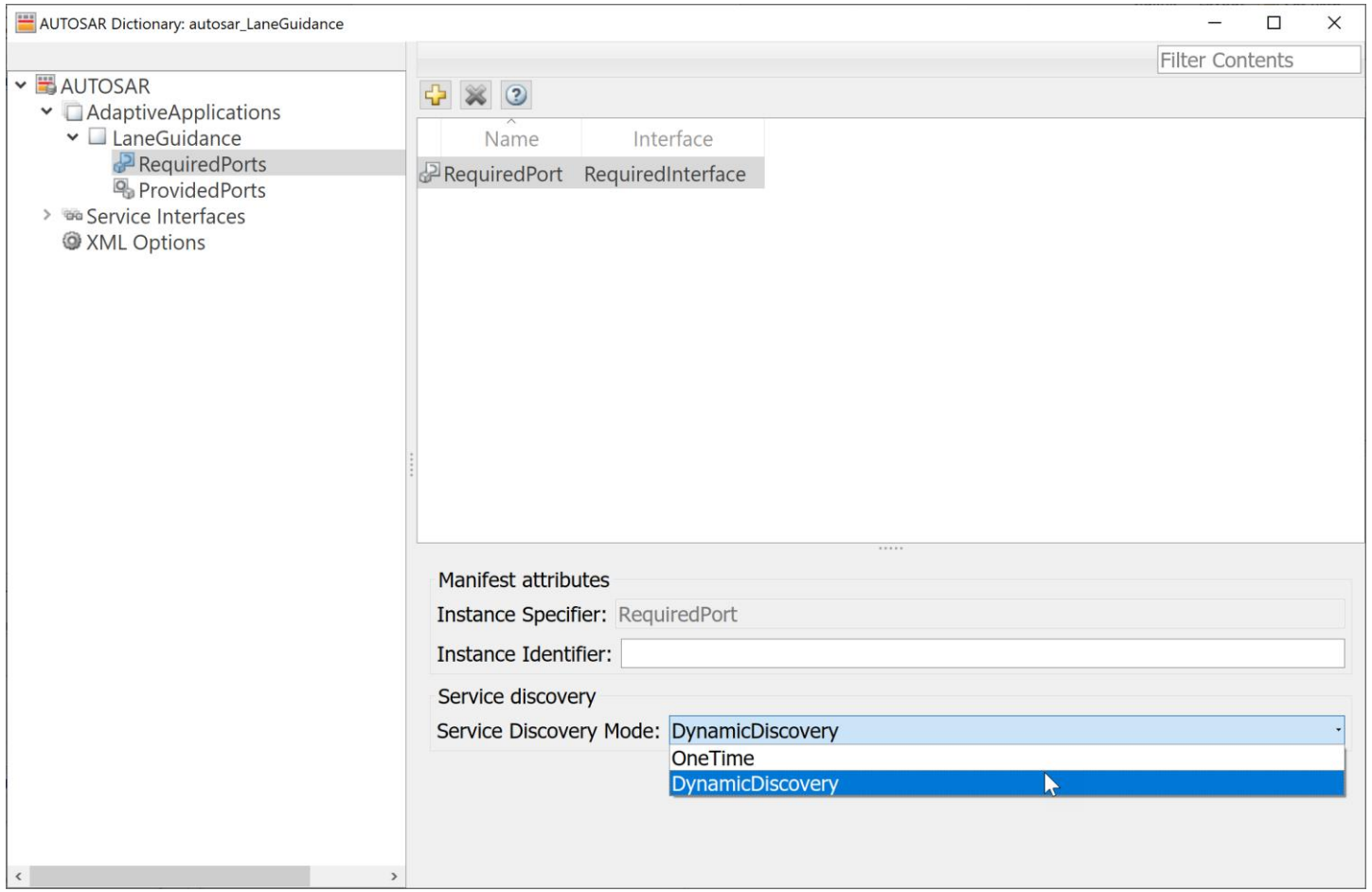


AUTOSAR Adaptive 开发步骤

- 导入ARXML创建模型框架

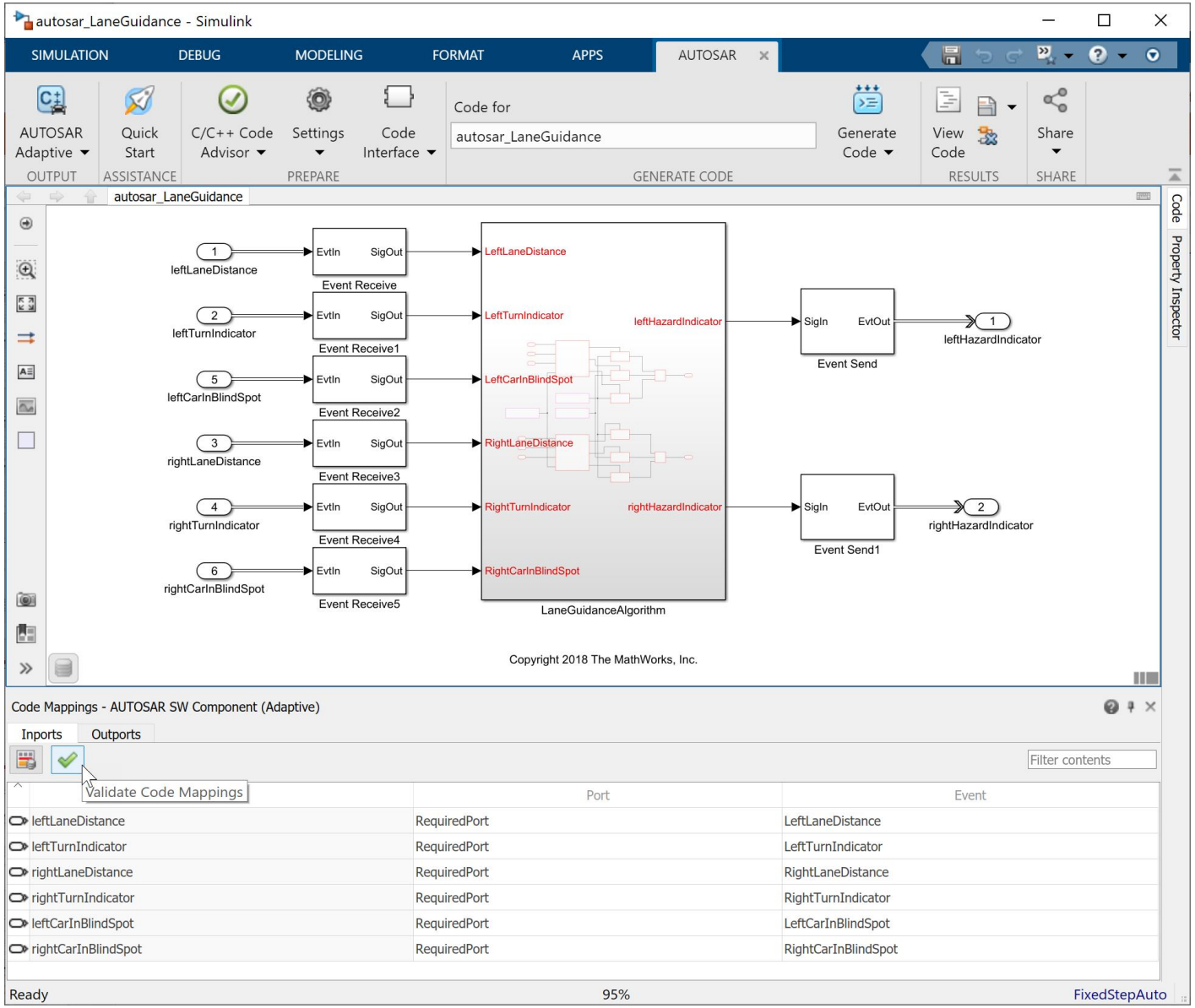
- 配置服务发现机制

  订阅自适应服务
  - 只在启动时, 或
  - 在其可用时动态发现



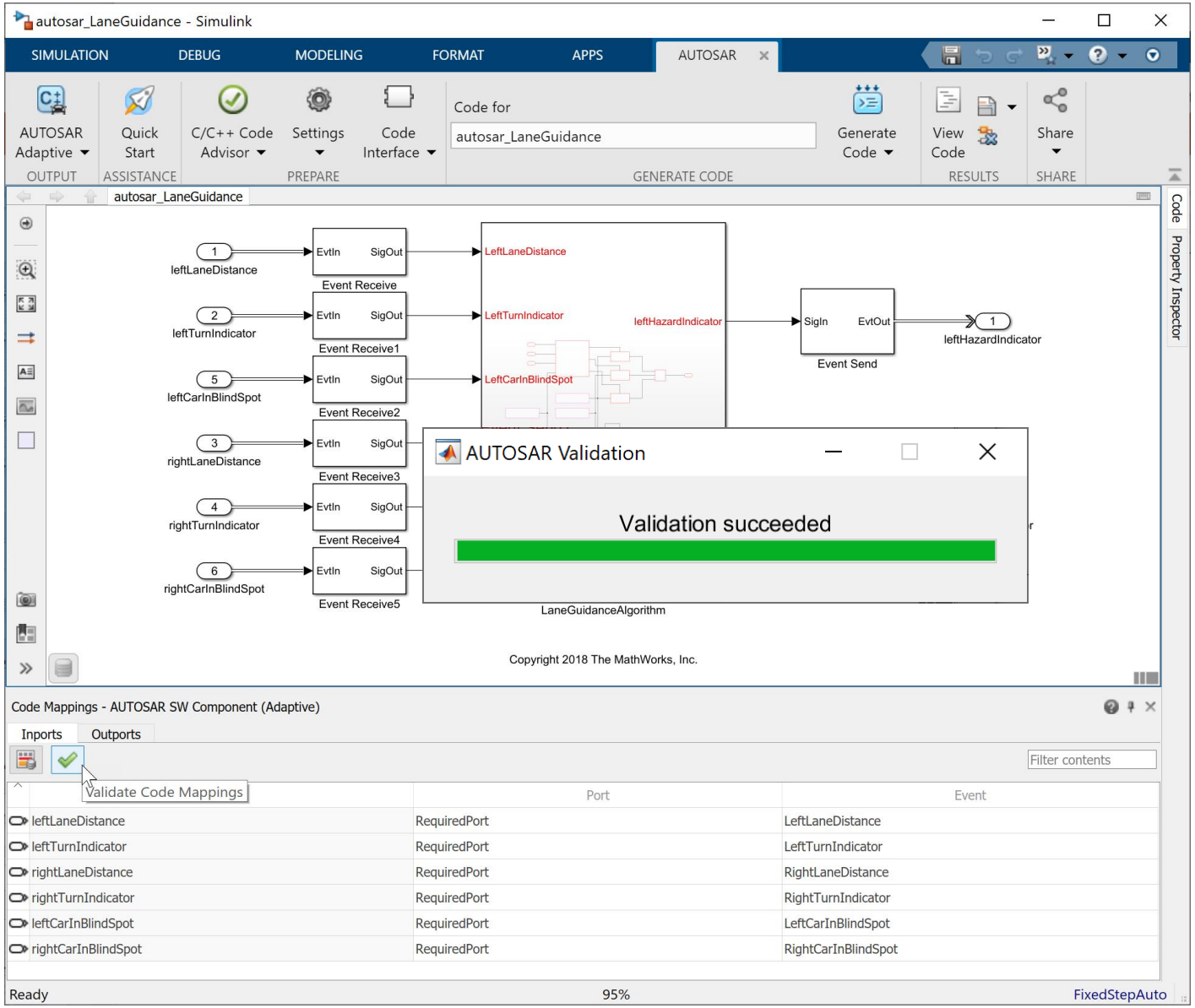
AUTOSAR Adaptive 开发步骤

- 导入ARXML创建模型框架
- 配置服务发现机制
- 确认AUTOSAR属性



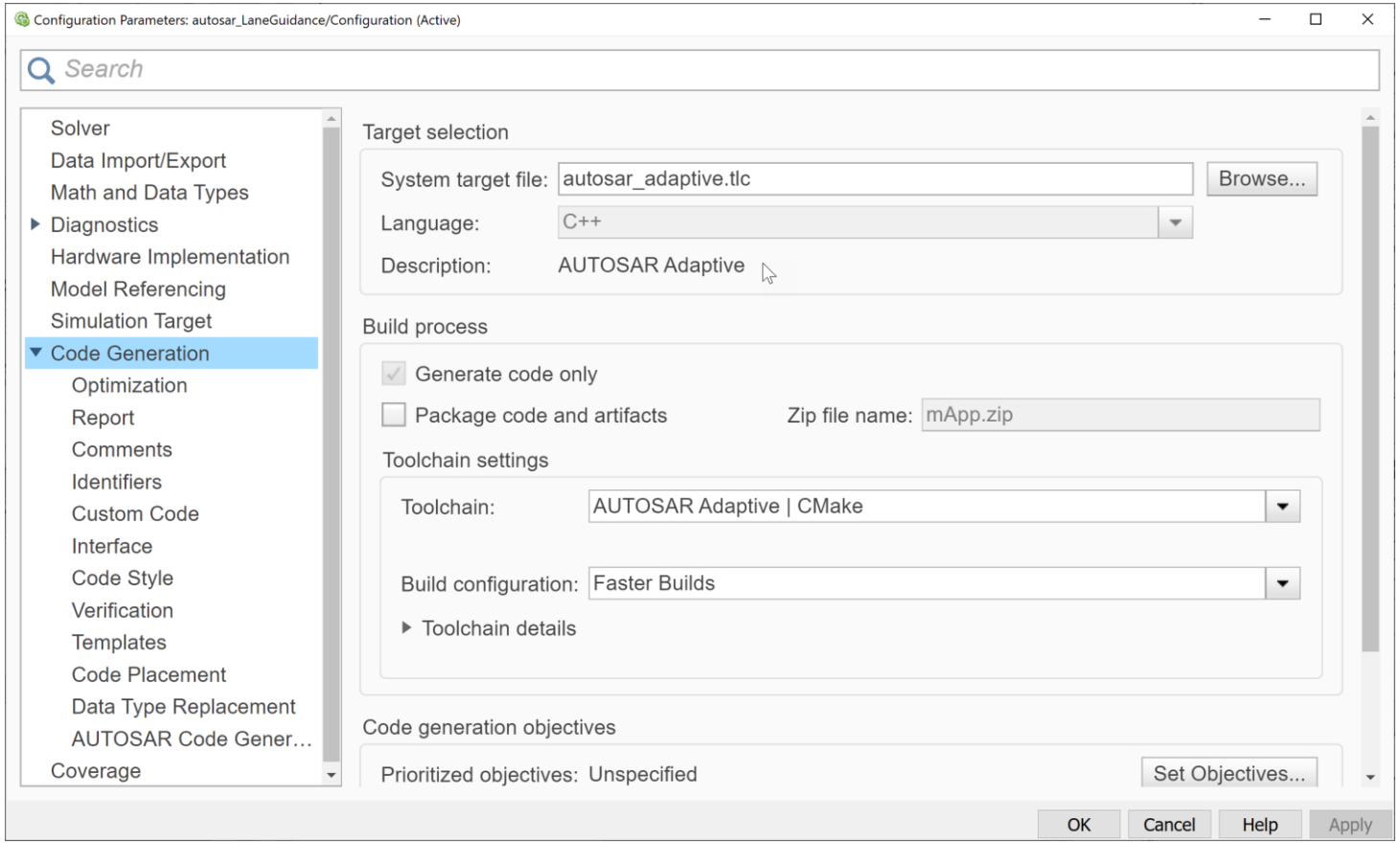
AUTOSAR Adaptive 开发步骤

- 导入ARXML创建模型框架
- 配置服务发现机制
- 确认AUTOSAR属性



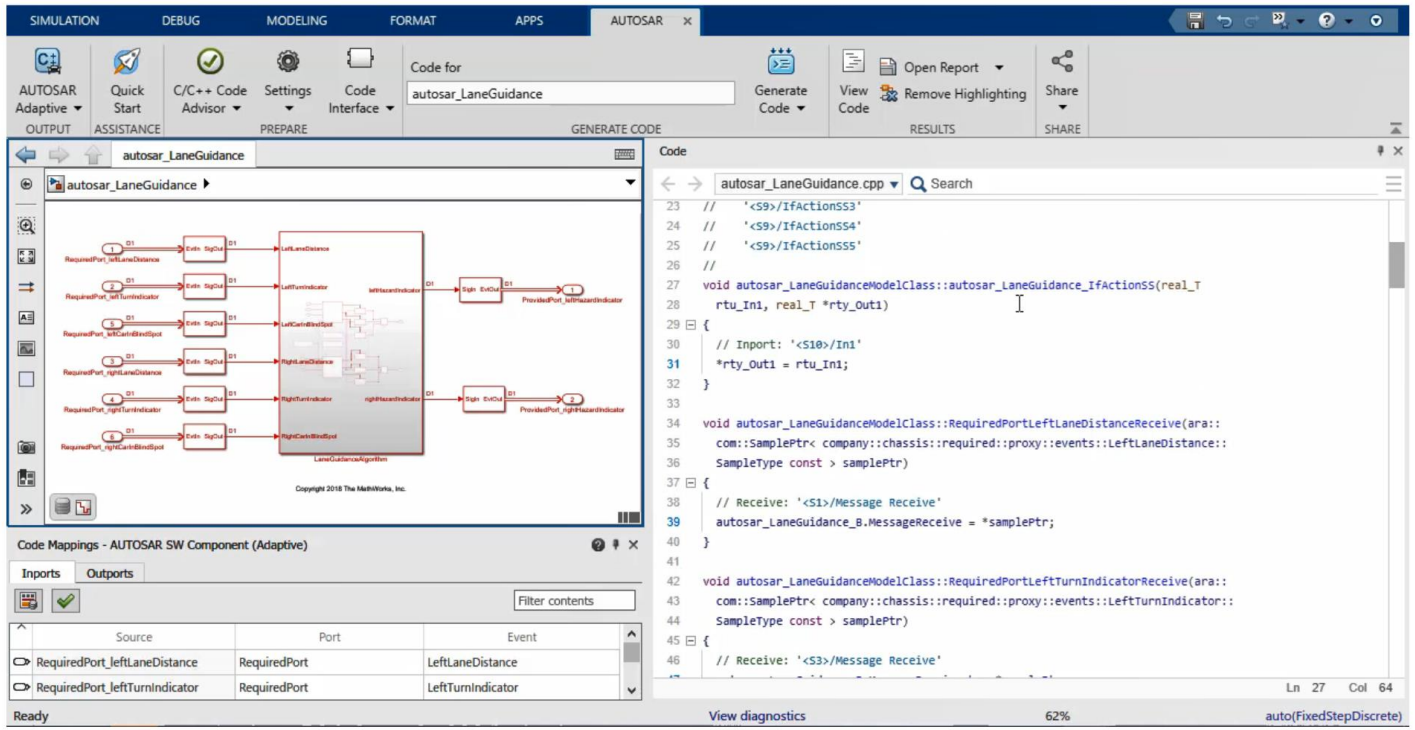
AUTOSAR Adaptive 开发步骤

- 导入ARXML创建模型框架
- 配置服务发现机制
- 确认AUTOSAR属性
- 生成代码



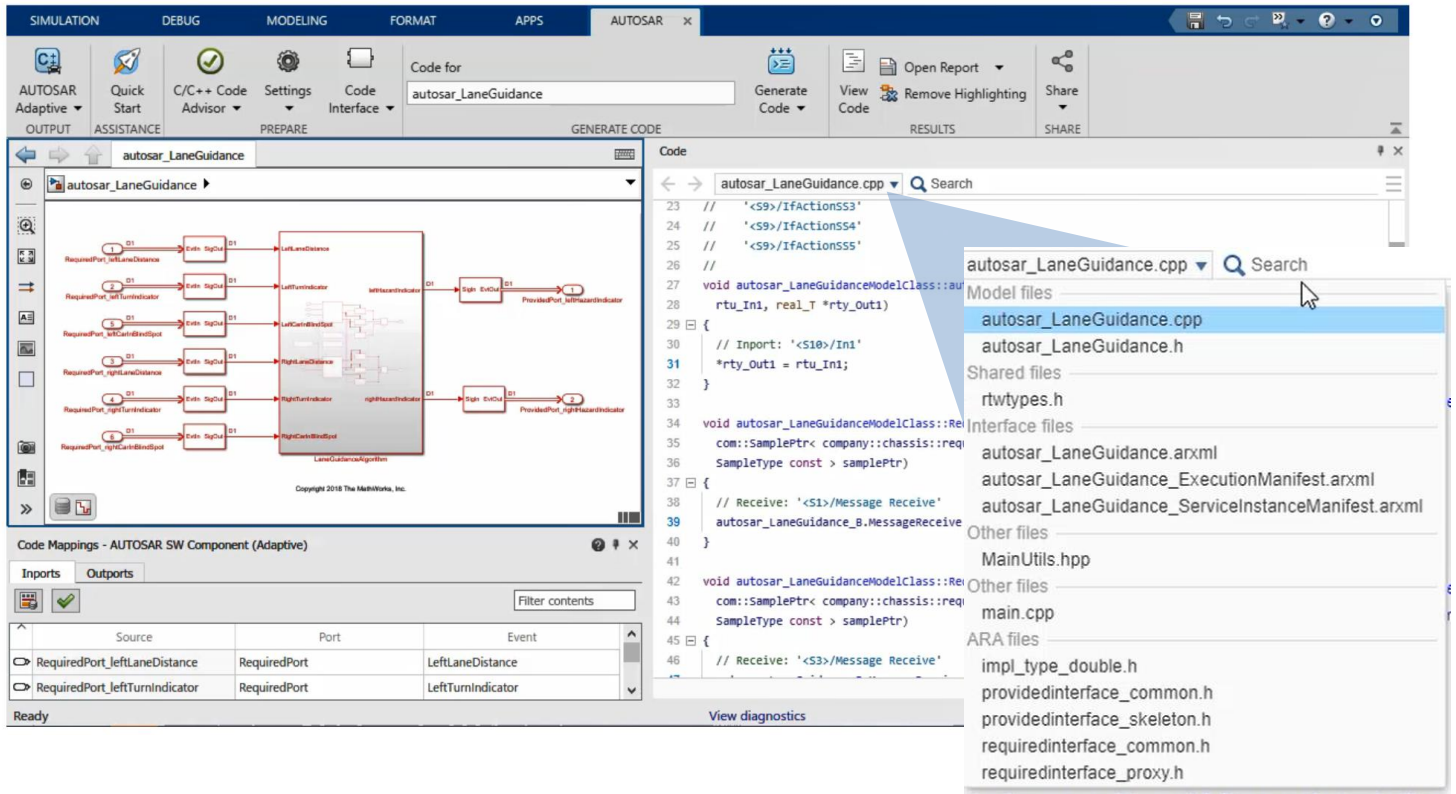
AUTOSAR Adaptive 开发步骤

- 导入ARXML创建模型框架
- 配置服务发现机制
- 确认AUTOSAR属性
- 生成代码



AUTOSAR Adaptive 开发步骤

- 导入ARXML创建模型框架
- 配置服务发现机制
- 确认AUTOSAR属性
- 生成代码



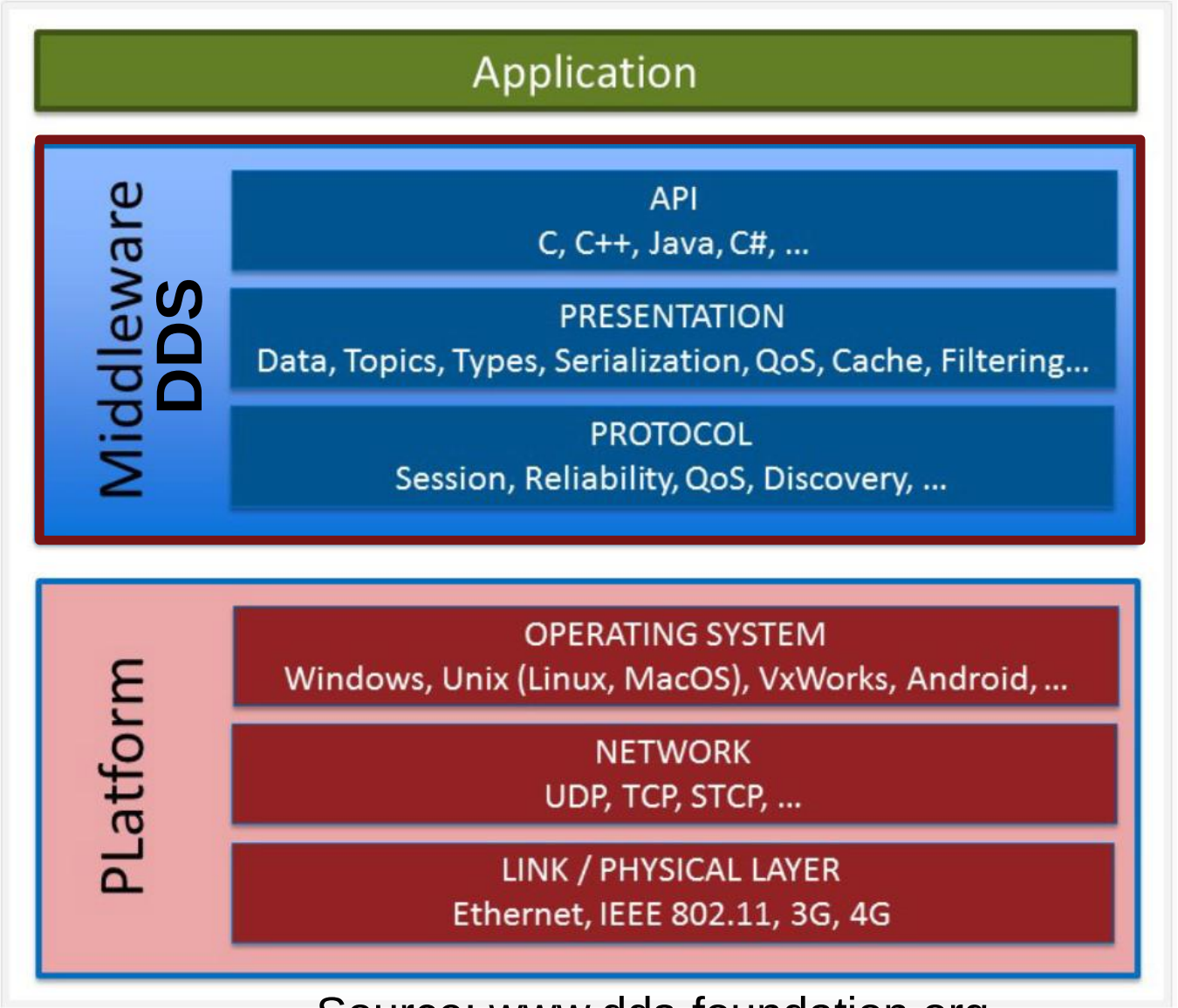
内容

- 软件定义汽车时代的行业挑战
- Simulink开发面向服务的架构(SOA)
- Simulink对SOA行业标准的支持
  - Simulink对AUTOSAR Adaptive的支持
  - Simulink对数据分发服务 (DDS)的支持
- 总结及如何获取更多相关信息

DDS概述

数据分发服务 (DDS)

- 从网络、数据格式和操作系统的细节中将应用抽象出来
- 数据中心通讯（发布-订阅）
- 服务质量和动态发现



Source: www.dds-foundation.org

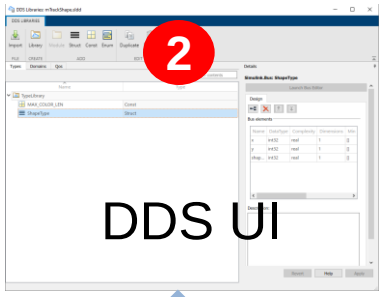
DDS支持 – DDS Blockset

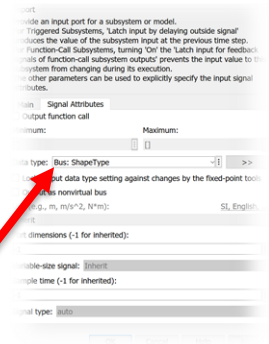
数据分发服务(DDS)采用SOA方法论解决实时和嵌入式系统的发布/订阅的通讯问题

DDS面向航空、国防、汽车和机器人等行业的实时数据交换应用需求

R2021a

DDS开发流程

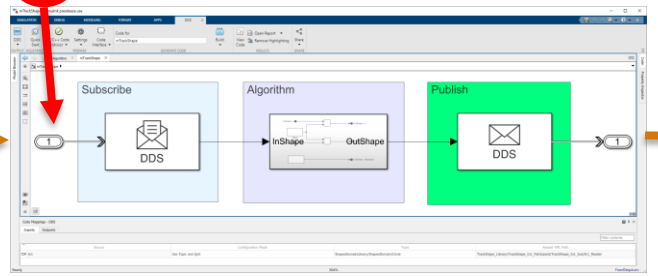




Import

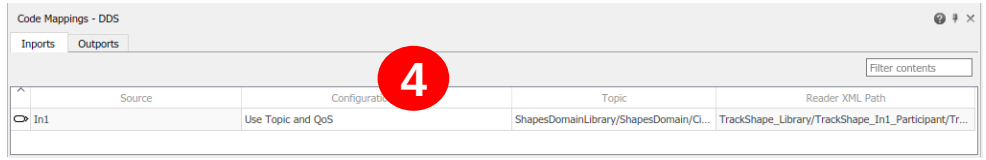
1

Simulink Data Dictionary



5

Generate

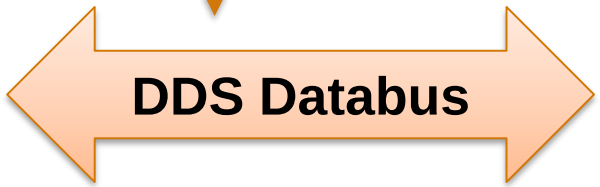


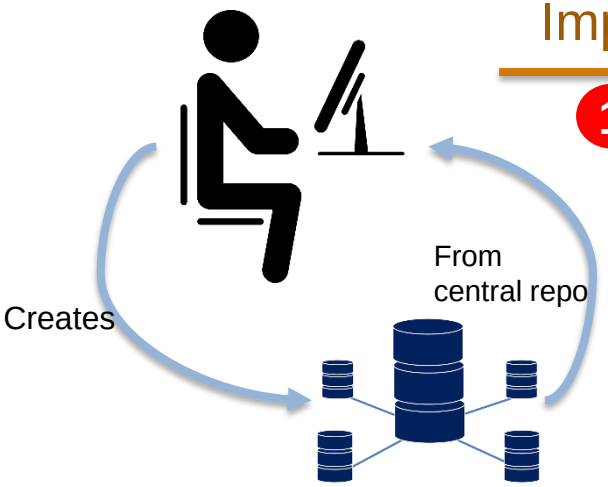
4

6

Deploy

IDL/XML
+
第三方
工具支持

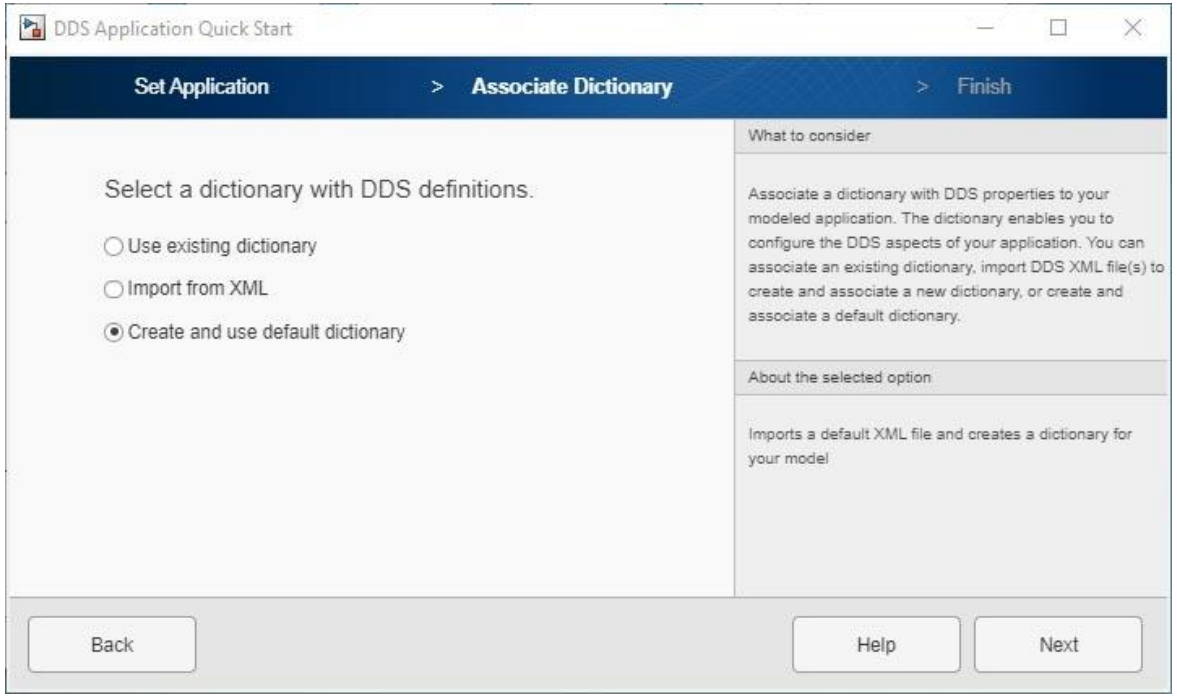




接口定义
QoS定义
(XML)

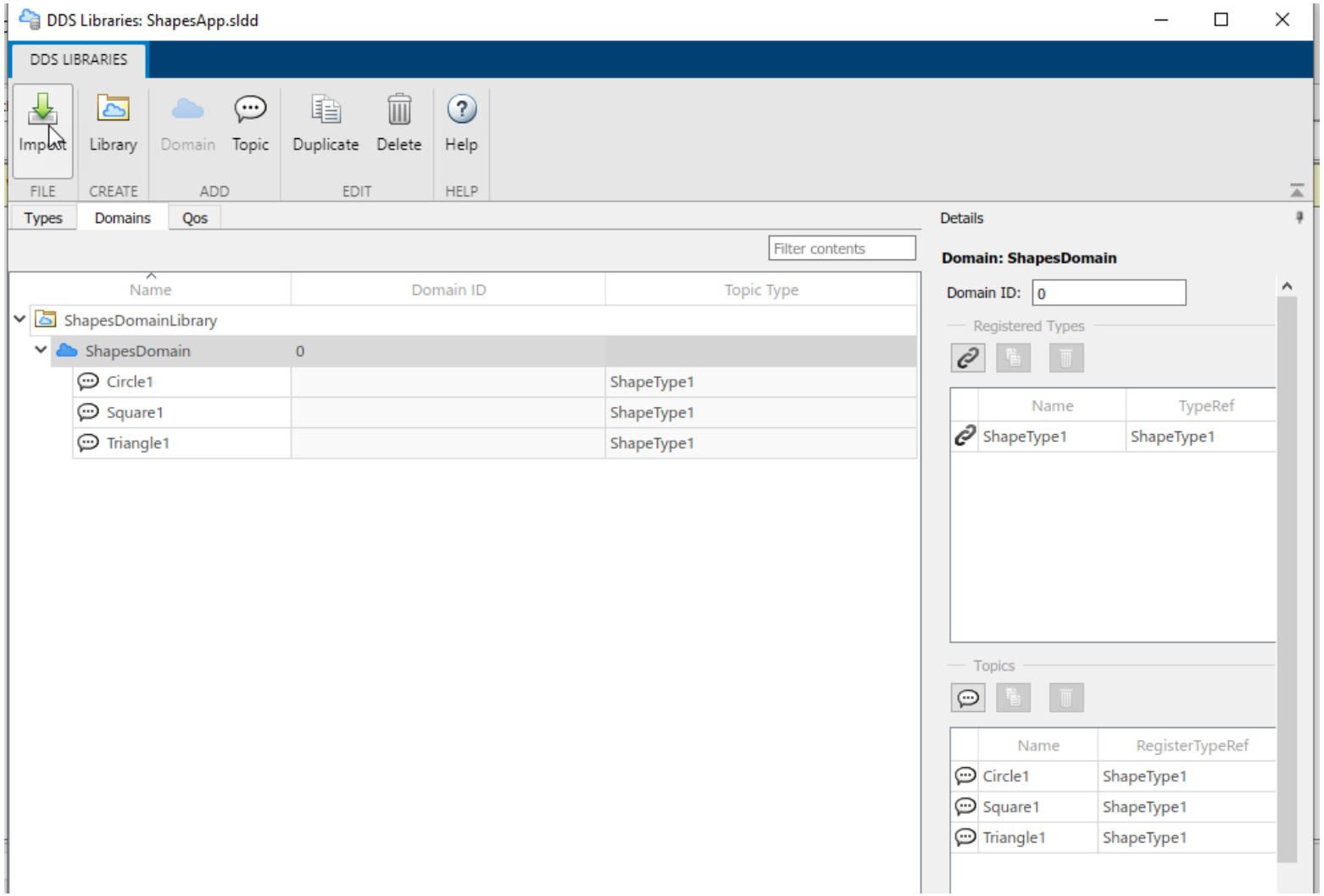
DDS 应用快速上手

- DDS 应用快速上手功能可以
  - 创建默认的DDS定义
  - 从XML文件导入DDS定义
  - 使用在模型中创建好的预定义数据



DDS数据词典

- 导入XML
- 定义/修改
  - 话题类型
  - 域
  - 服务质量策略

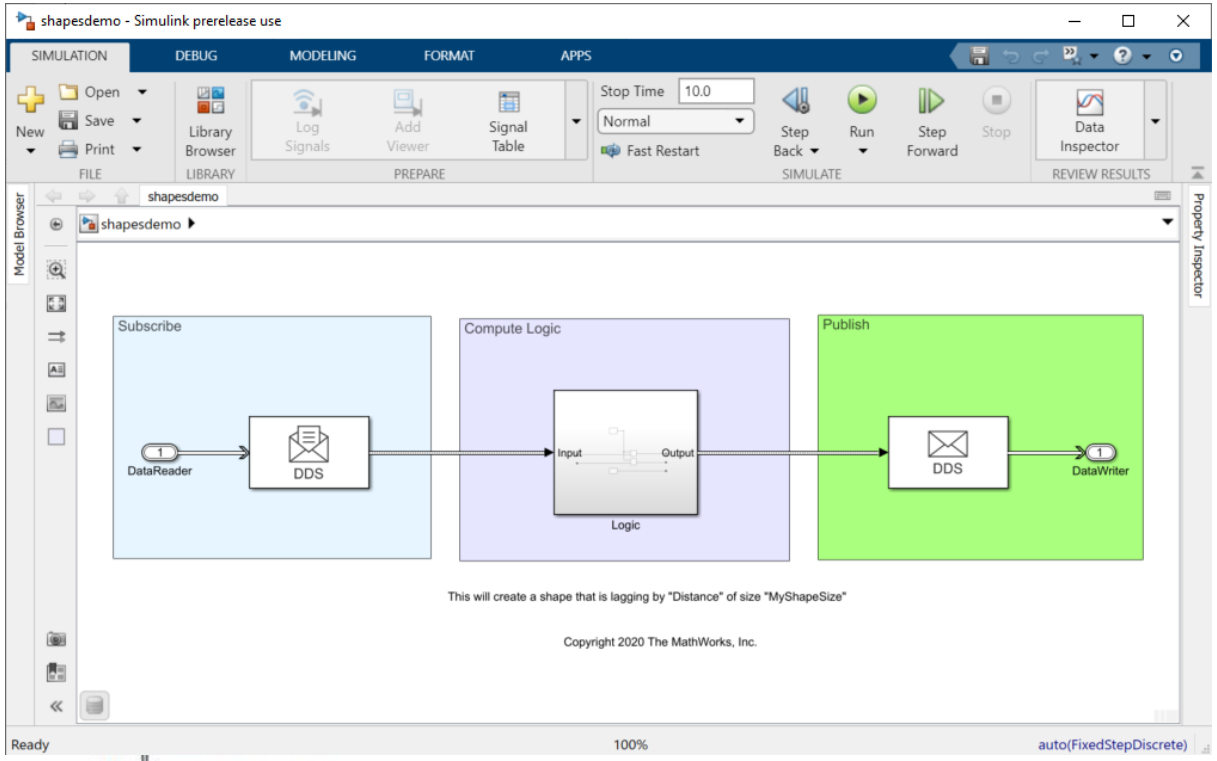


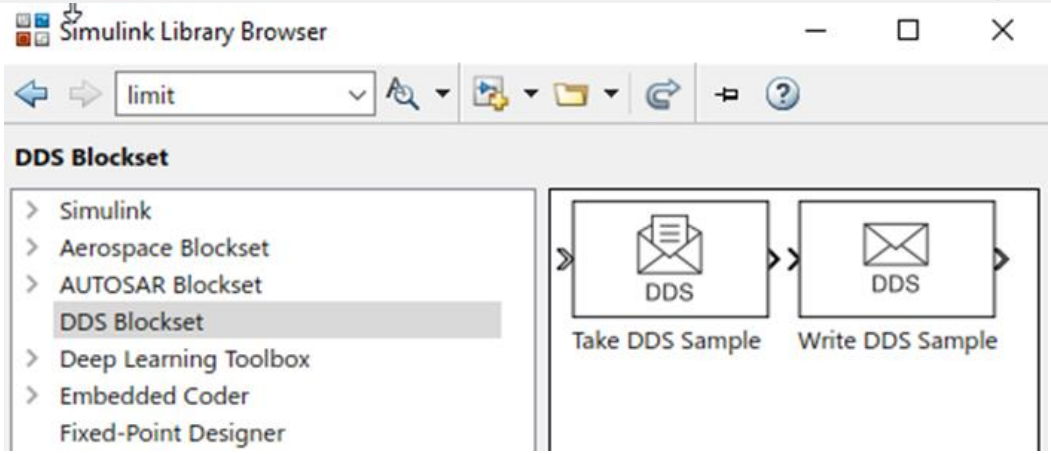
DDS建模模块

使用DDS模块建模发布或订阅机制

DDS Blockset 提供基本模块用于建模DDS应用

- 接收 DDS 采样
- 写入 DDS 采样



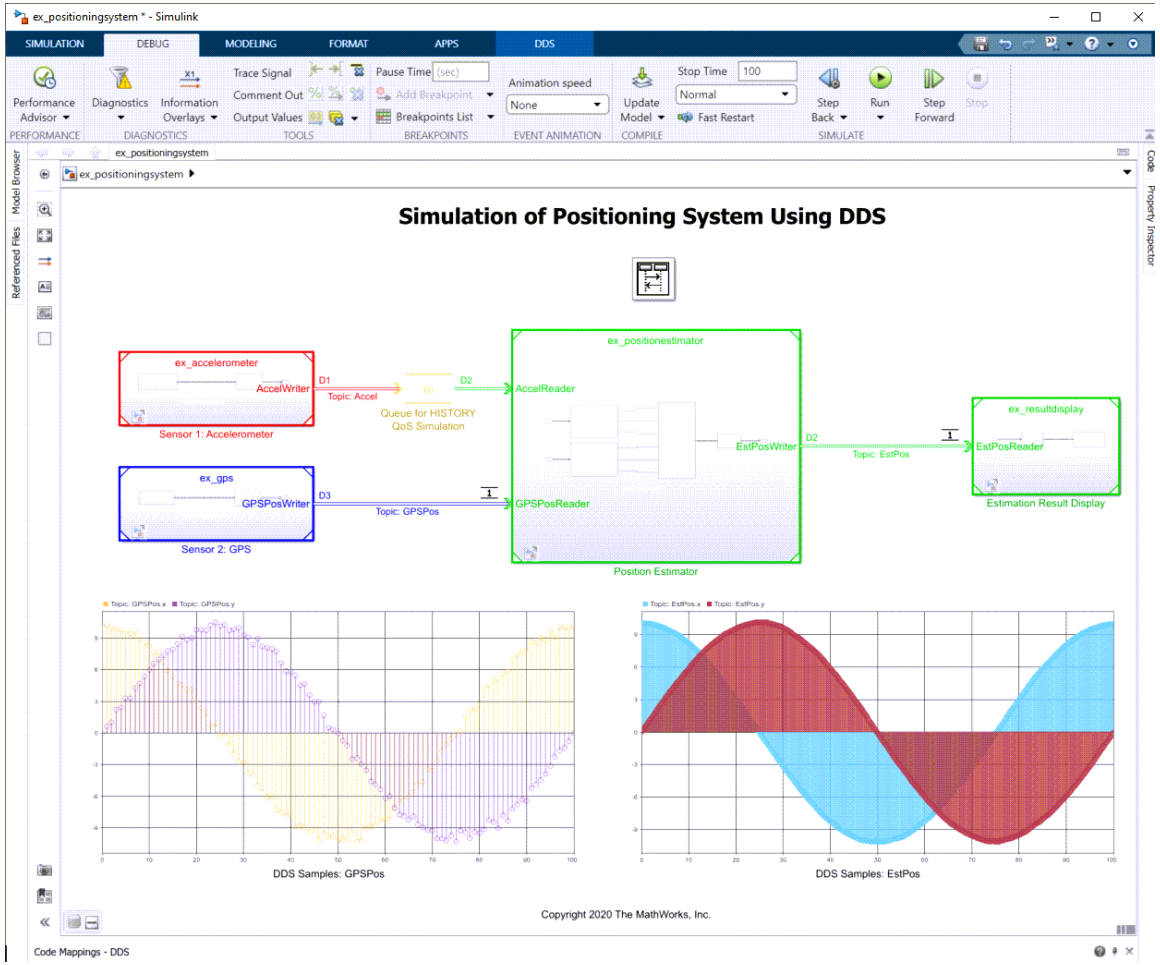



```
» open_system('shapesdemo');
```

DDS应用仿真

仿真DDS Simulink模型及其服务质量

使用Simulink建模和仿真包括历史数据在内的服务质量策略以验证运行时行为



» Live script demo

DDS代码生成和部署

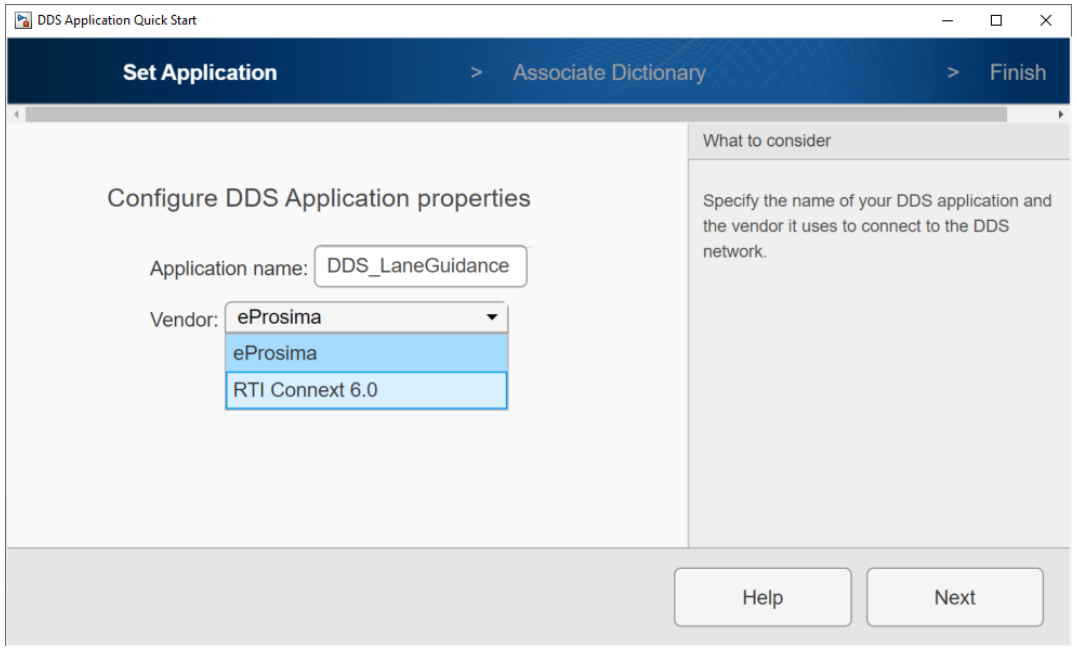
生成DDS可执行文件并部署到DDS网络

- 使用Embedded coder生成
  - 带DDS API的C++产品级代码
  - 用于部署的XML或IDL文件

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

与第三方DDS栈集成

DDS Blockset 与第三方DDS栈 (如RTI Connext 和 eProsima Fast DDS)完全集成



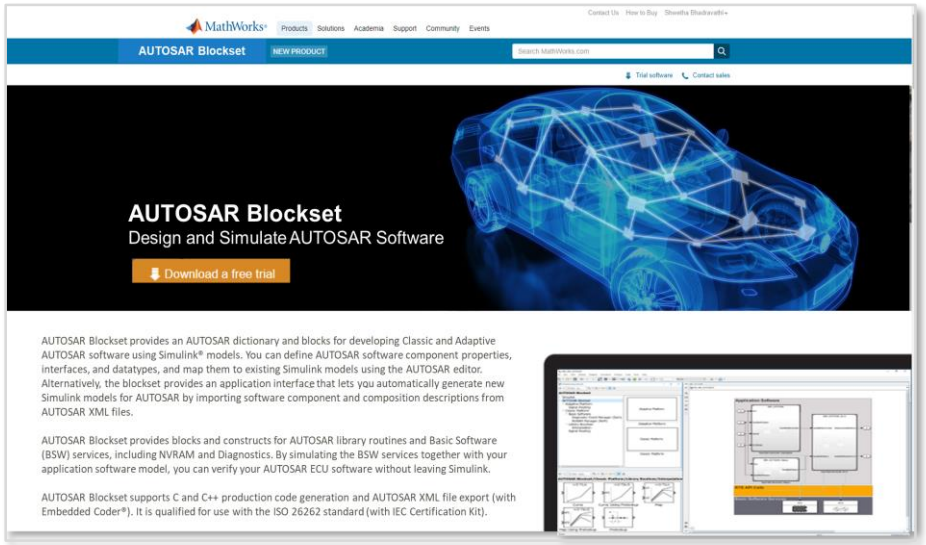
内容

- 软件定义汽车时代的行业挑战
- Simulink开发面向服务的架构(SOA)
- Simulink对SOA行业标准的支持
  - Simulink对AUTOSAR Adaptive的支持
  - Simulink对数据分发服务 (DDS)的支持
- 总结及如何获取更多相关信息

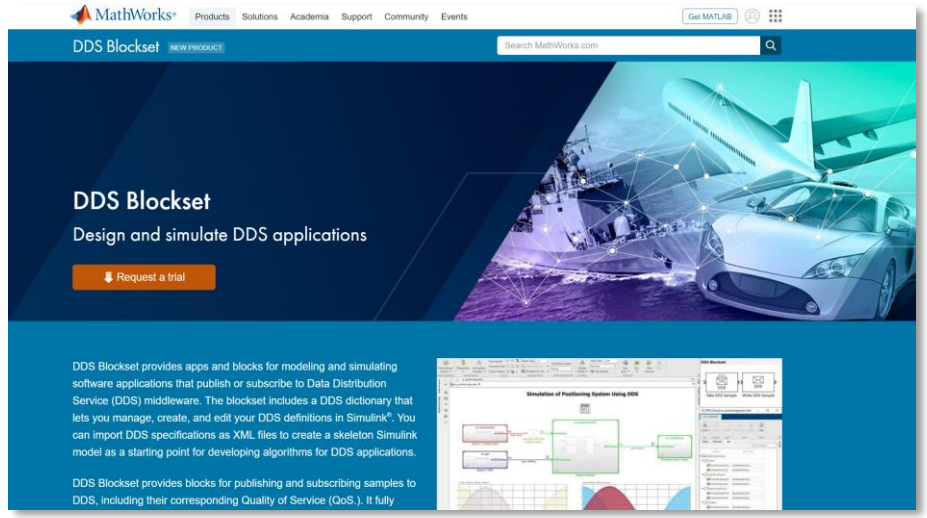
总结

- 高级的功能需求推动了汽车行业电子电气和软件架构的进化
- 软件的复杂管理和频繁升级要求引入了面向服务的架构
- Simulink支持设计、仿真、生成和部署面向服务的应用
- 最大化的复用已有模型和经验有助于降低迁移至SOA应用的风险

更多相关信息 – 产品主页和演示

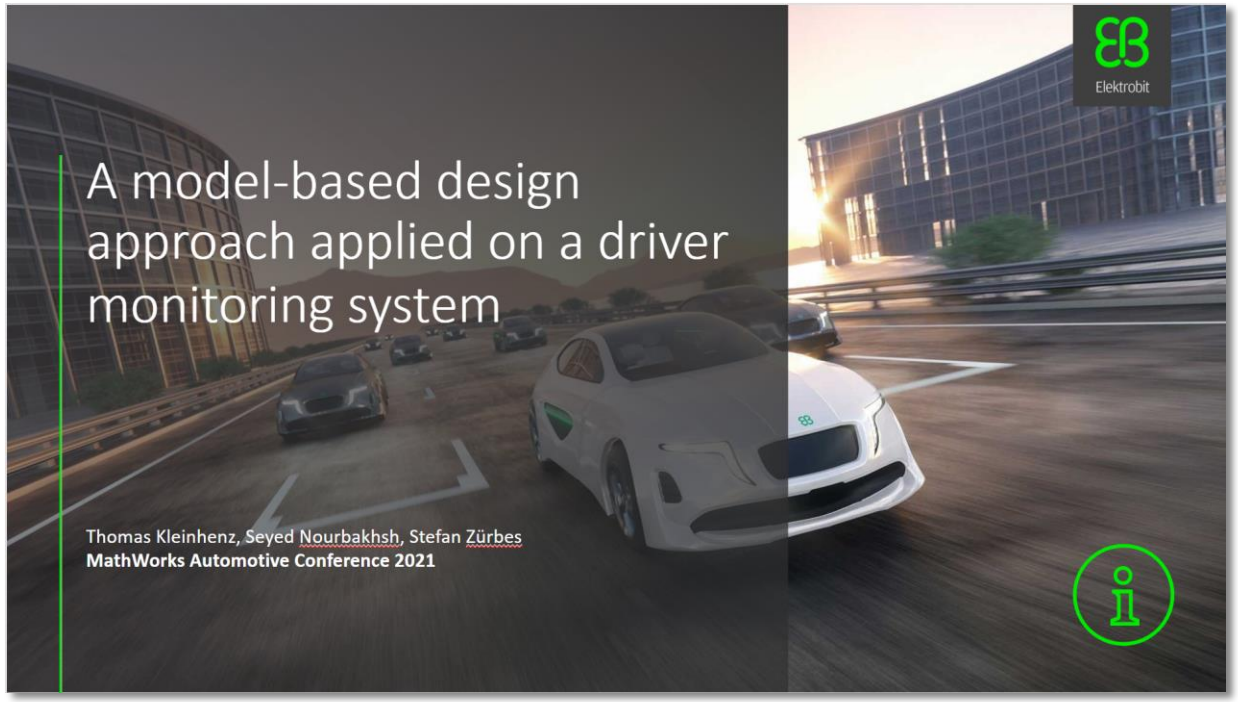


www.mathworks.com/products/autosar.html



www.mathworks.com/products/dds.html

更多相关信息 – 与第三方栈的集成



Elektrobit – A model-based design approach applied on a driver monitoring system (using Adaptive AUTOSAR)

Vector – The evolution of E/E Architecture and the Impact on Future Software Development.

MATLAB EXPO 2021

感谢参会！

MathWorks®