

# 使用 Simulink 部署 AUTOSAR 的 10个最佳实践

作者:David Jaffry 和 Holly Keener

## 目录

术语表.....	2
简介 .....	3
Simulink 为 AUTOSAR 提供哪些支持 .....	5
实施 AUTOSAR 的最佳实践.....	7
了解更多 .....	18

## 术语表

关于 AUTOSAR 部署的讨论涉及许多缩写。以下是本指南所用的主要术语。

缩写	含义
AAT	AUTOSAR 编写工具
API	应用程序编程接口
ARXML	AUTOSAR XML
AUTOSAR	汽车开放系统架构
BSW	基础软件
CRL	代码替换库 (Embedded Coder 的功能)
ECU	电子控制单元
IDE	集成开发环境
OEM	整车厂
PIL	处理器在环
RTE	运行时环境
SIL	软件在环
SWC	软件组件
V&V	验证和确认
XML	可扩展标记语言

## 简介

### AUTOSAR

AUTOSAR (汽车开放系统架构) 是由汽车制造商、供应商以及电子、半导体和软件行业的其他公司组成的全球开发合作伙伴关系。AUTOSAR 标准旨在实现软件标准化、重用和互操作性。

AUTOSAR 标准提供了两个平台, 为当前和下一代汽车 ECU 提供支持。一个是 Classic 平台, 用于动力总成系统、底盘、车身及内饰电子元件等传统应用领域。另一个是 Adaptive 平台, 用于新兴应用领域, 例如高度自动驾驶、车联网 (Car-to-X)、车载软件空中升级 (OTA) 或车辆纳入物联网。此外还引入了 Foundation AUTOSAR 标准, 用于加强这两种 AUTOSAR 平台之间的互操作性。

### MathWorks 支持

MathWorks 是 AUTOSAR 高级成员, 积极参与该标准的制定, 重点关注如何在汽车电子控制单元 (ECU) 的 AUTOSAR 开发流程中运用基于模型的设计。使用 Simulink® 和 AUTOSAR Blockset (图 1), 您可以:

- 对 AUTOSAR Classic 和 Adaptive 软件组件建模, 可以在 Simulink 中创建, 也可以从 ARXML 组件描述导入。
- 细化 AUTOSAR 组件配置并创建算法模型内容。
- 在 Simulink 中创建 AUTOSAR 架构模型 (需要 System Composer™)。
- 使用 Simulink 库模块 (包括基础软件服务, 如 NVRAM Manager 和 Diagnostic Event Manager) 进行 AUTOSAR 组件交互的系统级仿真。
- 生成 ARXML 组件描述和算法 C/C++ 代码, 以便在 Simulink 中测试, 或集成到 AUTOSAR 运行时环境 (RTE) (需要 Embedded Coder®)。

本指南依托采用基于模型的设计部署 AUTOSAR 的丰富经验, 总结了使用 Simulink 部署 AUTOSAR 的最佳实践。最终目的是形成一个系统化的企业级 AUTOSAR 开发流程, 能够最大限度地带来收益, 同时最大限度地降低与收益实现相关联的成本、风险和冲击。这些建议均源于指导整车厂和零部件供应商客户使用 MATLAB® 和 Simulink 进行基于模型的 AUTOSAR 部署的实践经验。

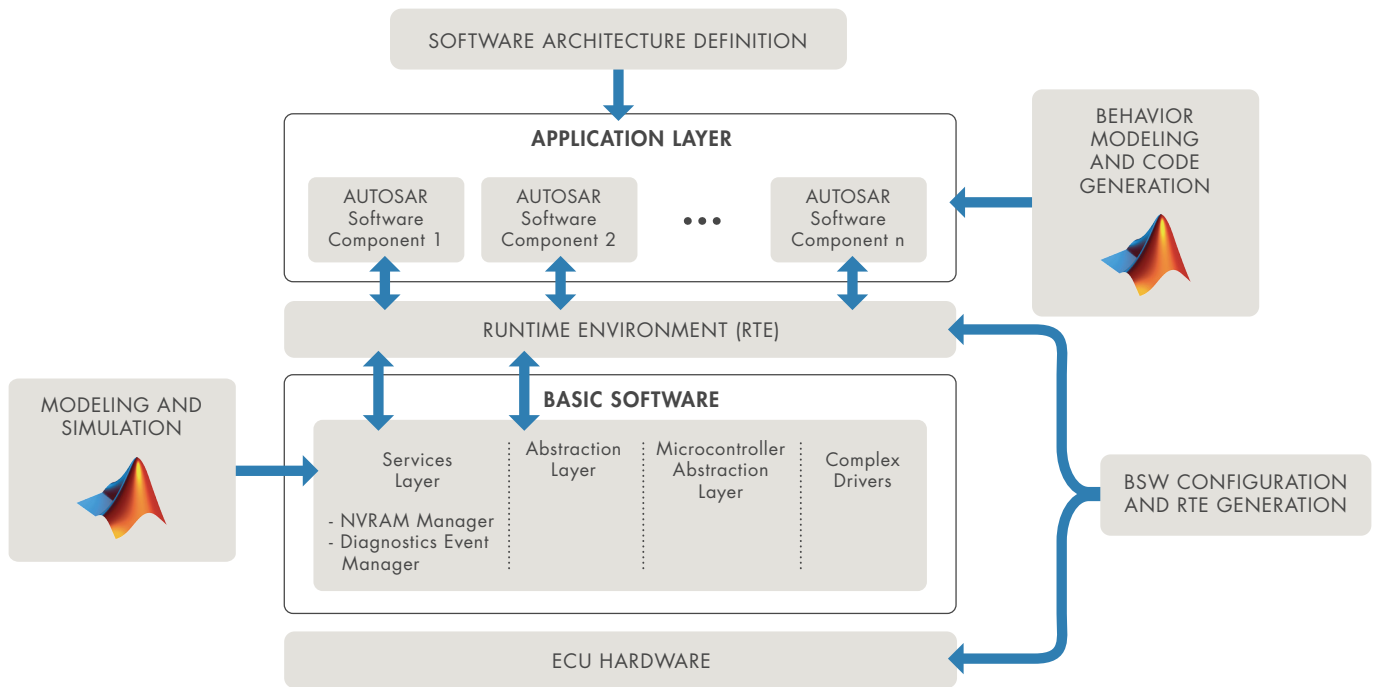


图 1.经典的 AUTOSAR 分层架构。

使用 Simulink 高效部署 AUTOSAR 的十大最佳实践包括：

1. 确定将现有 Simulink 模型迁移到 AUTOSAR 的策略
2. 只使用一种 AUTOSAR 工作流程
3. 选定数据管理策略
4. 建立建模标准
5. 先仿真，再生成代码
6. 使用产品级代码生成
7. 使用 Simulink 将现有代码迁移到 AUTOSAR
8. 贯彻自动化
9. 针对 ISO 26262 提前做好规划
10. 积极规划迁移

## Simulink 为 AUTOSAR 提供哪些支持

首先快速回顾一下 Simulink 产品系列为 AUTOSAR 提供哪些支持。您可以使用 Simulink 和 AUTOSAR Blockset 来设计、实现和验证 AUTOSAR 原子软件组件 (SWC)。每个 AUTOSAR 软件组件可以包含一个或多个入口函数，这些函数在 AUTOSAR 中作为可运行实体实现。AUTOSAR Blockset 支持将 Simulink 模型映射到 AUTOSAR，以便 Embedded Coder 生成 AUTOSAR 合规的 C/C++ 代码和 AUTOSAR XML (ARXML) 文件。

Simulink 模型的某些方面可用于实现 AUTOSAR 概念。将 Simulink 模型配置为使用 AUTOSAR 目标后，您就可以管理 AUTOSAR 属性，并在编辑器中将 Simulink 模型映射到 AUTOSAR，如图 2 所示：

- 配置 **AUTOSAR 组件**
- 使用 **代码映射编辑器** 进行 AUTOSAR 元素的映射，以便生成代码
- 使用 **AUTOSAR 组件设计器**，为包含 AUTOSAR 模块的模型生成可读、紧凑且快速的 C 和 C++ 代码，以部署到大规模生产所用的嵌入式处理器

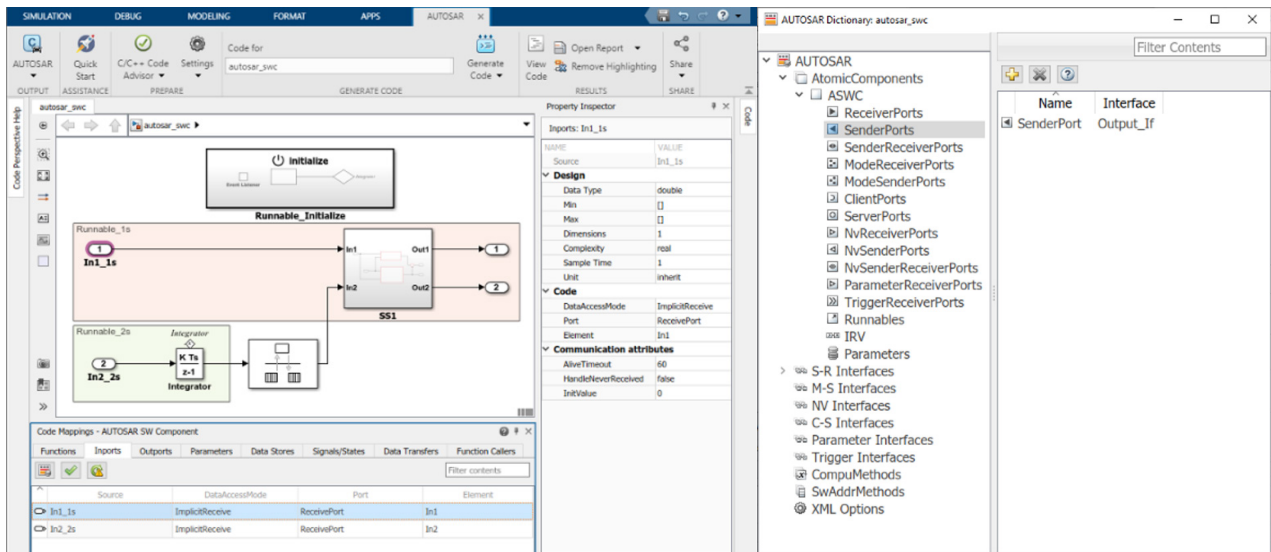


图 2. Simulink 到 AUTOSAR 的映射。

实施 AUTOSAR 时，有数种方法和工作流程可供选择。图 3 中说明了两种常见方法：自上而下，以及自下而上。

## 自上而下的方法

首先在 AUTOSAR 编写工具 (AAT) 中设计系统的架构, 然后生成 ARXML 文件并将其导入 Simulink 以生成模型的框架。然后, 您可以在 Simulink 中设计模型的内部行为。确认软件组件后, 您可以生成 C 代码。AAT 管理所有 ARXML 文件。

## 自下而上的方法

首先使用 Simulink 来创建、开发和确认您的软件组件。如果您有 System Composer, 可以创建一个软件架构画布来开发 AUTOSAR 组合和组件。您可以添加包括基础软件服务组件在内的仿真行为。在开发软件组件后, 生成 ARXML 组件描述和算法 C 或 C++ 代码, 以便在 Simulink 中测试, 或集成到 AUTOSAR 运行时环境。

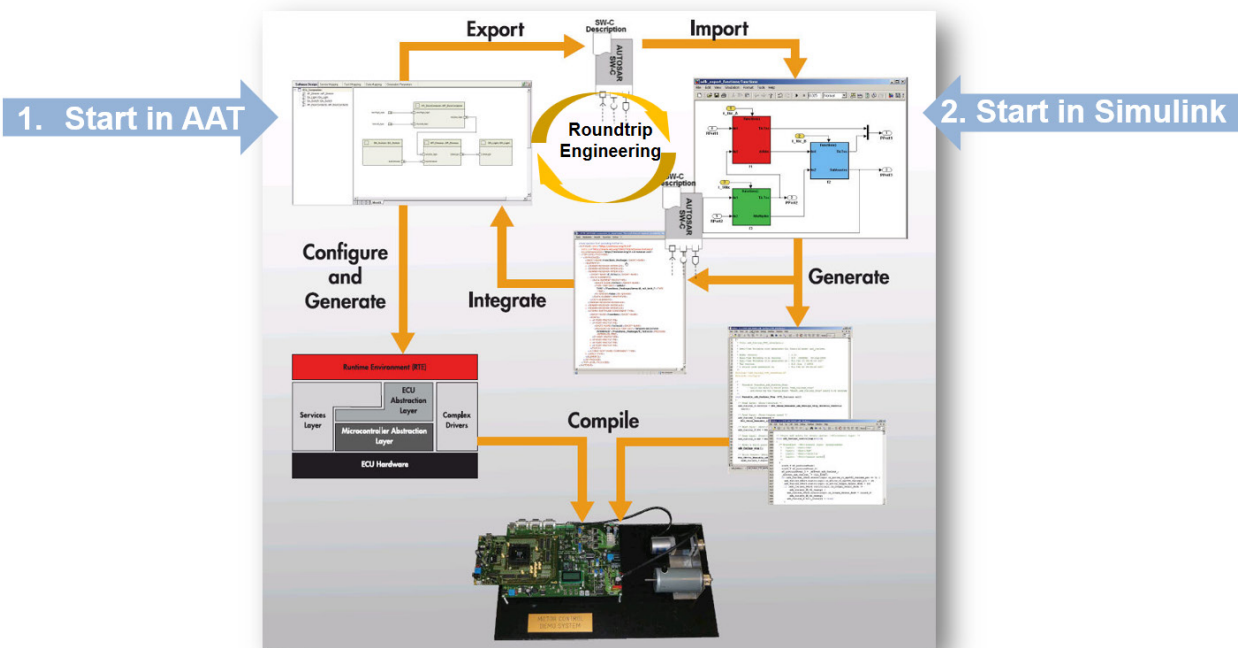


图 3.使用 Simulink 的 AUTOSAR 工作流程。自上而下的方法始于 AUTOSAR 编写工具; 自下而上的方法始于 Simulink。

# 实施 AUTOSAR 的最佳实践

## 1. 确定将现有 Simulink 模型迁移到 AUTOSAR 的策略

一种常见工作流程是先使用 Simulink 设计软件算法，再实施 AUTOSAR (图 4)。将设计从非 AUTOSAR 方法迁移到 AUTOSAR 时，有必要让整个开发团队采用统一的策略，以确保团队能够使用一致的流程和工具。要将设计迁移到 AUTOSAR，有以下三种策略：

- 从零开始
- 完全迁移到 AUTOSAR
- 一个模型同时适合 AUTOSAR 和非 AUTOSAR

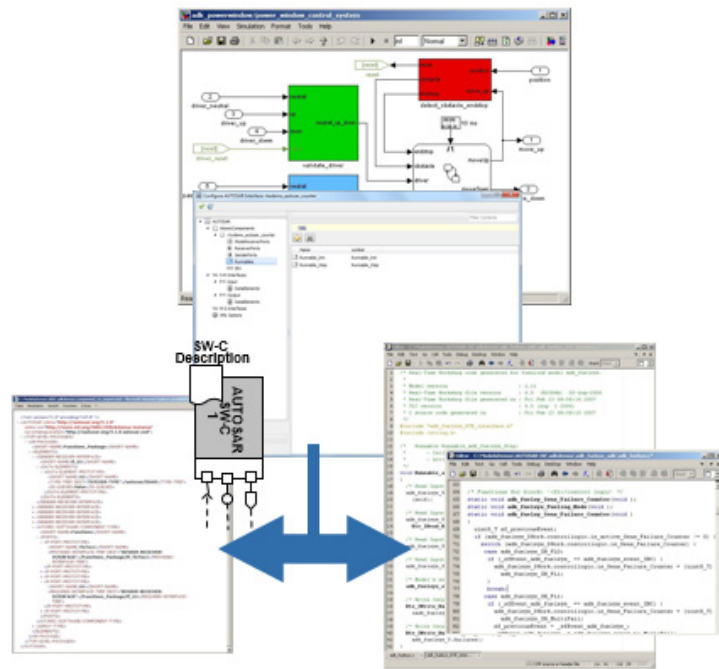


图 4. 从 Simulink 生成代码。

### 从零开始

如果算法需要大刀阔斧的修改，从“零”开始是最好的方法。设计流程始于需求，并在最初就确保新开发的 Simulink 模型适合 AUTOSAR (图 5)。由于不受原有设计的约束，此策略最为灵活，可以最大限度地利用 AUTOSAR 和基于模型的设计。然而，这一策略仅适用于极少重用原有实现的情形。

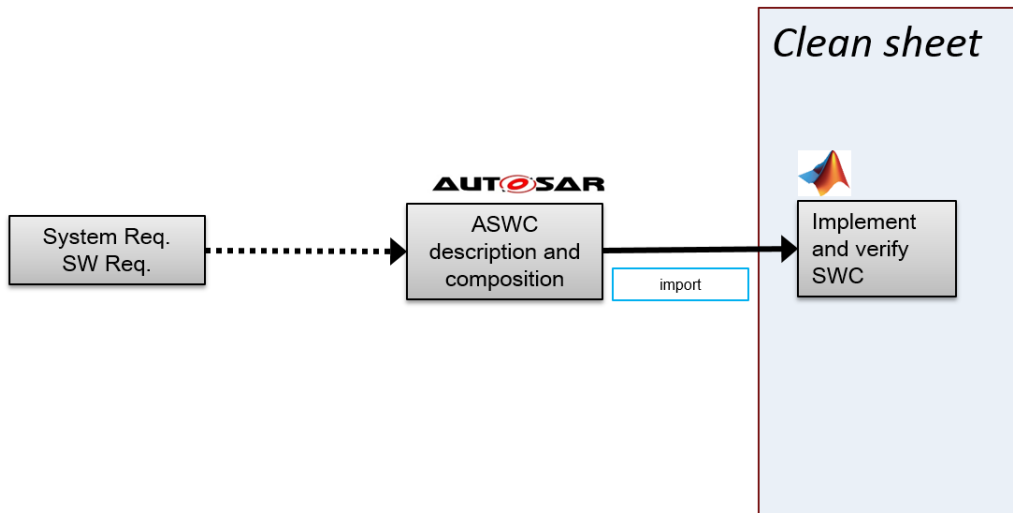


图 5.从零开始迁移。

### 完全迁移到 AUTOSAR

如果您需要重用现有 Simulink 算法，但又要放弃当前软件架构、转而使用 AUTOSAR，最佳策略是将您的 Simulink 模型完全转换或迁移到 AUTOSAR 最优建模风格。运用这一策略时，首先要选择适合 AUTOSAR 的 Simulink 建模风格和功能集，然后规划如何迁移各个模型（图 6）。必要时，您应更新 Simulink 实现以反映这种建模风格。最佳实践是在更改 Simulink 模型之前和之后使用仿真来执行回归测试，以确保您未引入任何功能更改。这是一种有效的折衷策略，既重用了原有实现，又利用了 AUTOSAR 和基于模型的设计。

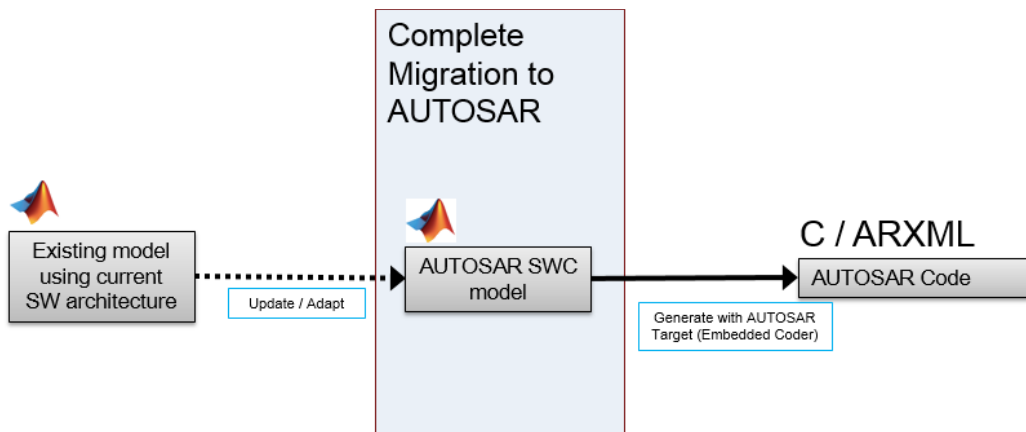


图 6.完全迁移到 AUTOSAR。



## 一个模型同时适合 AUTOSAR 和非 AUTOSAR

使用 Simulink 和 Embedded Coder 实施 AUTOSAR 有一大优势，即能够基于同一个模型为 AUTOSAR 和非 AUTOSAR 目标生成代码（图 7）。在为单个实现并行开发 AUTOSAR 和非 AUTOSAR 应用程序时，这种策略很有效。与完全迁移策略类似，您需要对建模风格和 Simulink 功能进行一次性更改，以确保设计符合 AUTOSAR 所需的 Simulink 建模风格和约束。同样，在此方案中，回归测试是迁移流程的重要一环。单实现对应多架构可能会限制 AUTOSAR 的实施程度。但是，该策略只需设计和测试一个模型，即可部署到两个目标架构中，这无疑显著提高了效率。

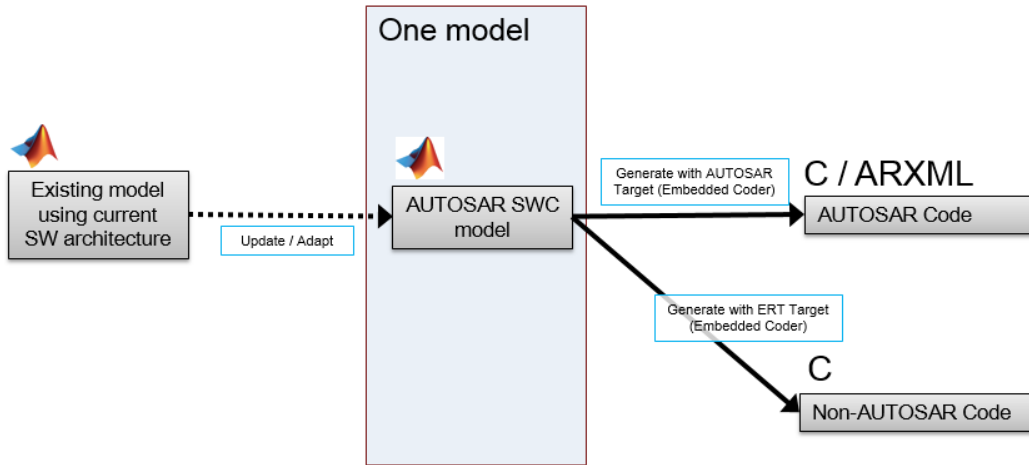


图 7.单模型迁移。

无论您采用哪种迁移策略，都有必要为迁移流程选择合适的范围，可以首先选取一个具有代表性的 Simulink 模型进行小型试点项目，然后扩展至更为完整的系统组成部分。一个典型的嵌入式系统往往涉及多个域；因此，在基于模型的设计中，必须使用相应的建模域清晰表示各个系统组件。

## 2.只使用一种 AUTOSAR 工作流程

AUTOSAR 标准包括工具互操作性的要求,这有利于高效实施自上而下和自下而上的工作流程(图 8)。您需要评估每种工作流程的利弊,然后选择合适的工作流程,以尽可能实现初始 AUTOSAR 实施的目标。如果在单个项目中引入多种工作流程,就会需要额外进行迭代和检查来确保数据和设计中的一致性,由此导致混乱、返工和浪费。

尤其是往返式工作流程,只有在数据所有者唯一且明确的情况下,才能发挥最大效用。

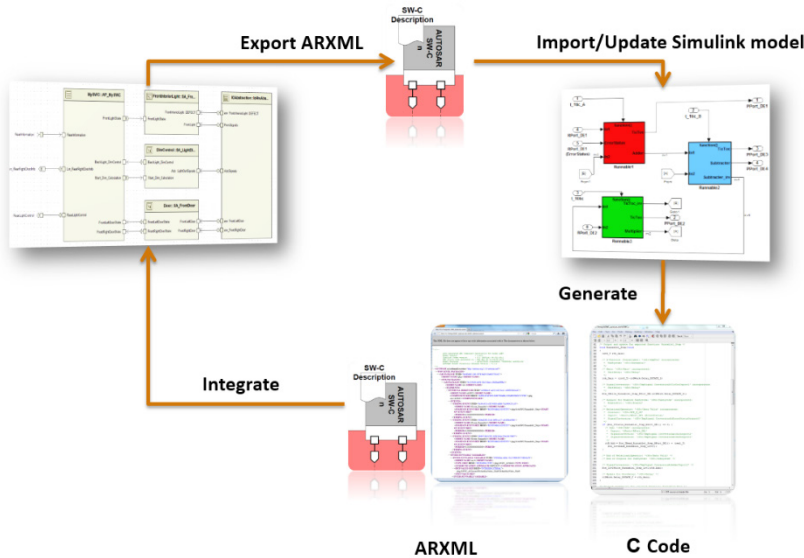


图 8.往返式 AUTOSAR 工作流程。

选定工作流程后,您需要选择一种最简单的方法,将 AUTOSAR 配置应用于 Simulink 模型。在自上而下的工作流程中,您可以使用 AUTOSAR Blockset 的内置功能,通过导入 ARXML 并创建或更新 Simulink 模型来应用 AUTOSAR 配置。在自下而上的工作流程中,您可以使用 AUTOSAR Blockset 提供的图形化方法或编程方法,对 AUTOSAR 配置进行应用和更新。您需要避免引入复杂的自定义方法来管理和配置 AUTOSAR 属性,否则日后就需要长期维护这些自定义解决方案。商业工具可为最主要的工作流程提供强有力的支持。

选择工具时,您需要考虑所选的工作流程,以及对实施最为关键的 AUTOSAR 概念。

### 3.选定数据管理策略

AUTOSAR 有多种模式可指定关键设计数据。关键设计数据管理是 AUTOSAR 实施策略的一大重点。数据可以通过 Simulink、AUTOSAR 编写工具或外部工具进行管理。数据也可以由不同所有者在不同级别上定义和管理。变更管理是数据管理策略的另一项重点。要确定数据管理策略,最佳实践是综合考虑所选的 AUTOSAR 工作流程、数据所有权策略、原有数据管理方法,以及将现有 Simulink 设计迁移到 AUTOSAR 的策略。

Simulink 中的 AUTOSAR 组件设计器 (图 9) 可用于管理 AUTOSAR 软件组件的数据。参数和变量数据的位置和作用域都限于组件内部,而系统级常量是跨组件共享的。组件设计器提供代码映射编辑器和属性检查器来管理内部组件数据。这些 Simulink 工具对 AUTOSAR 专门数据的管理形成补充。

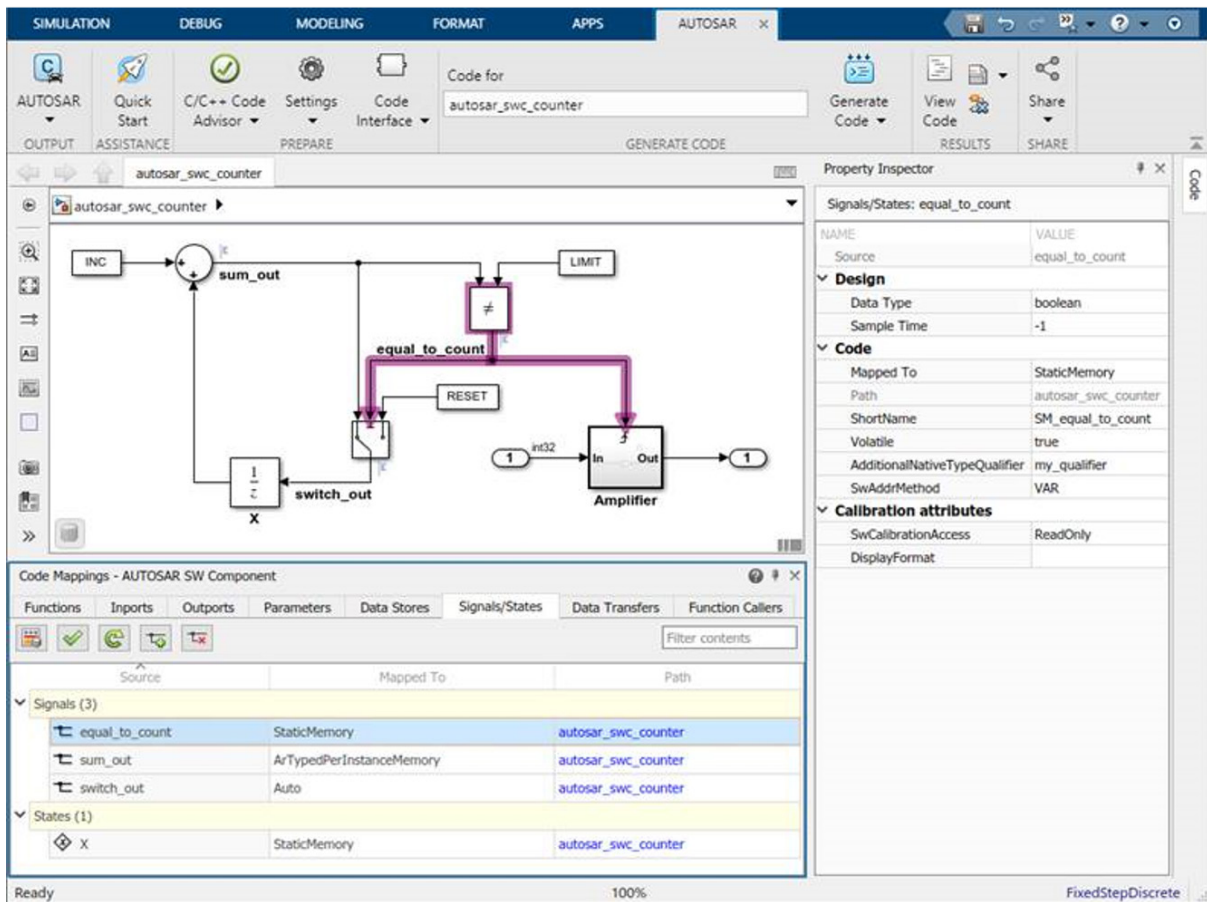


图 9.AUTOSAR 组件设计器。

此外,您可以配置组件数据来进行运行时标定和测量以满足需求。

## 4. 建立建模标准

建立一套 Simulink 和 AUTOSAR 建模标准, 确保企业上下统一实施, 从而能够跨项目重用并扩展到大型系统和团队。确立建模标准是部署基于模型的设计的最佳实践, 也是 ISO 26262 安全标准的一项要求。

建模标准应综合考虑 Simulink 模型的 AUTOSAR 实施策略、所选功能集、工作流程和数据管理策略。AUTOSAR 软件组件建模的最佳实践包括:

- 数据封装: 在每个 AUTOSAR 软件组件内, 封装参数和变量数据。使用 Simulink 参数、数据存储、信号和状态对 AUTOSAR 组件内部数据进行建模。通过数据封装, 一个组件的多个实例可以灵活地包含在组件层次结构和系统级仿真中, 或在多个项目中使用。
- 接口管理: 跨 AUTOSAR 组件和项目共享接口和类型。使用 AUTOSAR Blockset, 您可以为要导入或创建的 AUTOSAR 软件组件预定义接口和类型。例如, 企业可以预定义和共享:
  - AUTOSAR 组件端口接口及其输入/输出数据类型
  - AUTOSAR 组件内部数据类型 (IncludedDataTypeSets)
  - AUTOSAR 数据类型和相关元素 (AUTOSAR XML 包)
  - AUTOSAR 系统常量

您还可以阅读 [文档](#), 了解 AUTOSAR 软件组件行为的建模模式 (图 10)。

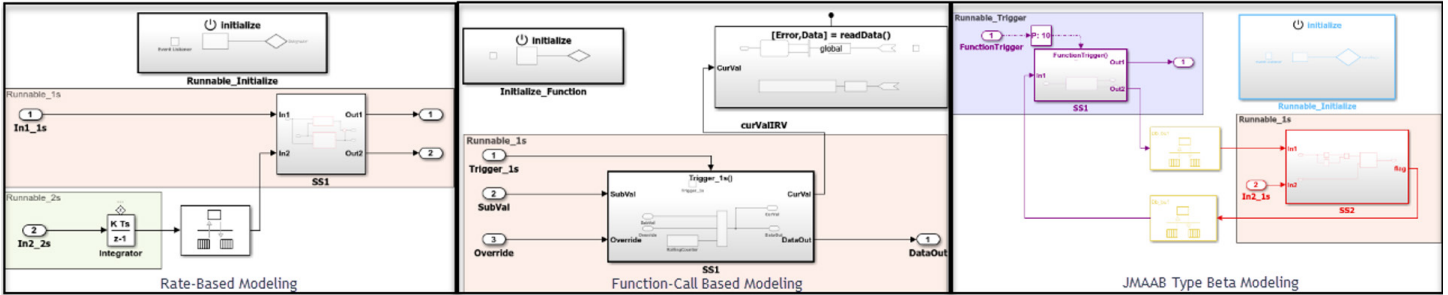


图 10. 建模模式。

## 5.先仿真,再生成代码

在引入 AUTOSAR 的过程中,新的工作流程、ARXML 文件的正确性问题以及软件所受的影响都可能带来困扰。不过,之所以下力气在 Simulink 中设计软件组件,就是为了在生成 ARXML 并与 AAT 和 RTE 生成工具集成之前,率先确保实现满足需求。Simulink 可确保为支持 AUTOSAR 标准丰富性而引入的各项功能也能得到有效的仿真(图 11)。

该仿真还有助于您:

- 在设计流程的早期确保软件组件的实现是正确的。
- 在代码集成之前,使用 Simulink 一起仿真多个 SWC (AUTOSAR 组合)。
- 在 RTE 生成之前,重用测试以进行软件在环 (SIL) 和处理器在环 (PIL) 仿真,在单元级别验证生成的代码。

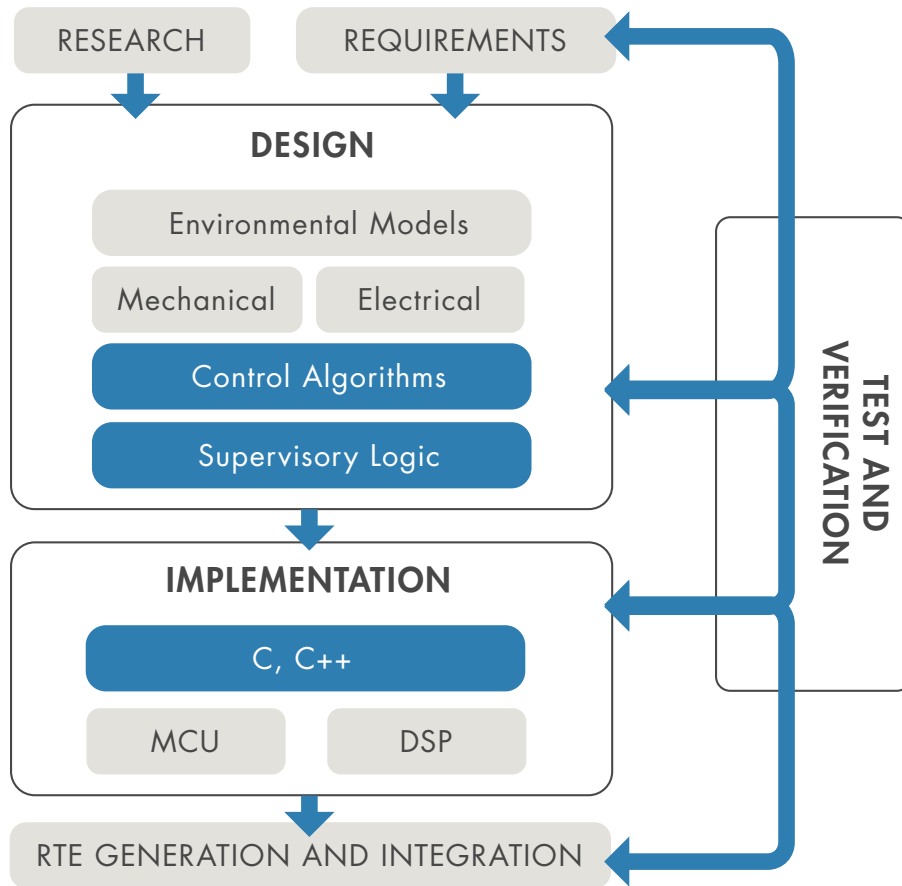


图 11.基于模型的设计中的活动。

## 6.使用产品级代码生成

基于 AUTOSAR 标准,可使用工具生成 AUTOSAR 基础软件 (BSW)、RTE、ARXML 文件等,因此也无需再困惑于是否为软件组件生成代码。最佳实践是使用产品级代码生成来实现软件组件。由于 API 相当复杂,如图 12 所示,手动进行 AUTOSAR 编码显然劳神费力。借助 Embedded Coder 和 AUTOSAR Blockset,您可以获得与 Simulink 中的设计行为极为接近的实现,从而充分发挥仿真和验证活动的效用。

```
void SeatBeltReminder_Logic(void)
{
  boolean_T TmpSignalConversionAtSeatBelt_a;
  uint16_T TmpSignalConversionAtSpeedOutpo;
  TmpSignalConversionAtSeatBelt_a =
  Rte_IrvIRead_SBR_Logic_SeatBeltfasten_debounced();
  TmpSignalConversionAtSpeedOutpo = Rte_IrvIRead_SBR_L
  if ((uint32_T)Seat_Belt_Reminder_AUTOSA_DWork.tempora
  Seat_Belt_Reminder_AUTOSA_DWork.temporalCounter_i1
  ((int32_T)Seat_Belt_Reminder_AUTOSA_DWork.tempora
  }
  switch
  (Seat_Belt_Reminder_AUTOSA_DWork.bitsForTID2.is_c2_Seat_Belt_Reminder_AUTOSA)
  {
  case Seat_Belt_Reminder_IN_KEY_CRANK:
  if ((int32_T)Rte_IRead_SBR_Logic_KEY_KEY() == 1) {
  Seat_Belt_Reminder_AUTOSA_DWork.bitsForTID2.is_c2_Seat_Belt_Reminder_AUTOSA
  = Seat_Belt_Reminder_AU_IN_KEY_ON;
  .....
  case Seat_Belt_Reminder_A_IN_KEY_OFF:
  if ((int32_T)Rte_IRead_SBR_Logic_KEY_KEY() == 1
  .....
  }
  break;
  }
  .....
  Rte_IWrite_SBR_Logic_SeatBeltIcon SeatBeltIcon
  (Seat_Belt_Reminder_AUTOSAR_B.SeatBeltIcon);
  }
}
```

</RUNNABLE-ENTITY>  
<RUNNABLE-ENTITY UUID="08f71b09-4c5d-52bb-f537-04e3170d9e5d">  
<SHORT-NAME>SeatBeltReminder\_Logic</SHORT-NAME>  
<MINIMUM-START-INTERVAL>0</MINIMUM-START-INTERVAL>  
<CAN-BE-INVOKED-CONCURRENTLY>>false</CAN-BE-INVOKED-CONCURRENTLY>

<SENDER-RECEIVER-INTERFACE UUID="a0f3e55a-368d-5ac3-825d-a92f04518baf">  
<SHORT-NAME>SeatBeltIcon</SHORT-NAME>  
<IS-SERVICE>>false</IS-SERVICE>  
<DATA-ELEMENTS>

图 12.使用 AUTOSAR API 生成的代码。

使用 Embedded Coder,您可以从 Simulink 生成产品级 C/C++ 代码。代码既可支持 AUTOSAR 合规,也可支持非 AUTOSAR 合规。

## 7.使用 Simulink 将现有代码迁移到 AUTOSAR

经过充分确认并已广泛使用的非 AUTOSAR 源代码（现有代码）通常会被重用，成为初始 AUTOSAR 实施的一部分。您需要对此代码进行一些修改，以确保其 AUTOSAR 合规性。最佳实践不但建议自动生成代码，还推荐为现有代码自动生成 AUTOSAR 合规的包装器。您可以使用 C Caller 模块的 C S-Function 模块将此代码引入 Simulink 中，以便随后将模型映射到 AUTOSAR。通过这种方法，您可以将 C 或 C++ 代码集成到 Simulink 中，用于仿真和产品级代码生成。

将代码集成到 Simulink 后，您可以：

- 创建测试框架
- 可视化和分析输出
- 实现模型覆盖率目标，包括集成的现有代码

在产品级代码生成中，Embedded Coder 将生成必要的 AUTOSAR RTE API 访问点和可运行实体，用作现有代码的包装器。图 13 展示了一段最终生成的代码。

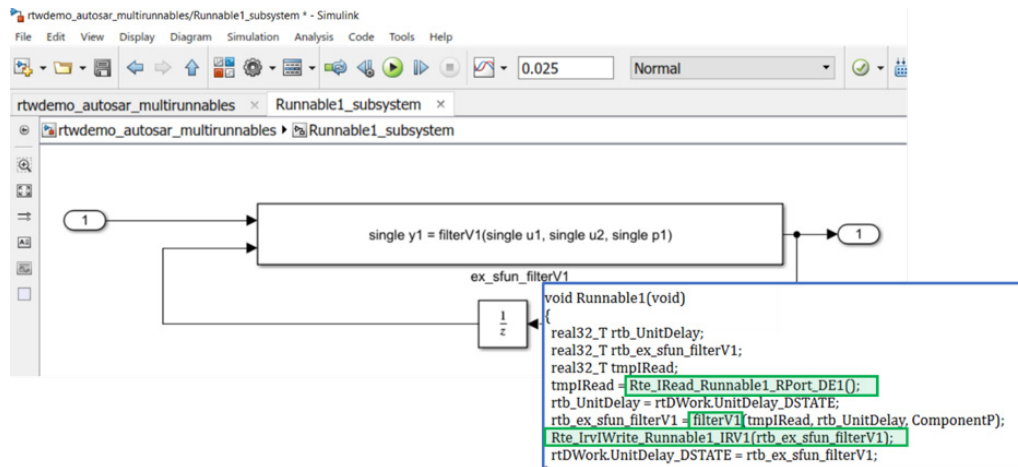


图 13.使用 AUTOSAR API 集成现有代码。

## 8.贯彻自动化

在典型的 AUTOSAR 工作流程中, 整个团队会反复执行一些共有步骤。手动配置和部署 AUTOSAR 可能相当困难, 原因在于:

- 标准的复杂性, 例如命名约定
- AUTOSAR 带来的迭代工作周期
- 复杂的代码 API 和 XML 文件定义

最佳实践是使用 API 实现工作流程自动化。您可以使用公开的 MATLAB API (如图 14 所示), 在 Simulink 中配置 SWC。如果选择创建函数以自动执行上述任务, 请花时间确保每项自动化任务都有充分的文档记录并可验证, 以便每次都能顺利升级到新版本, 并且可以随着时间推移逐渐弃用一些自定义解决方案。

如果完全只使用 [公开的 API](#), 将有助于未来使用新版本 MATLAB 进行迁移。

```
% Setup AUTOSAR Configuration Programmatically
model = 'autosar_sw_counter';

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

图 14. AUTOSAR 目标的 MATLAB API。



## 9.针对 ISO 26262 提前做好规划

ISO 26262 是道路车辆功能安全的全球标准。与实施 AUTOSAR 类似，实施 ISO 26262 合规会对原有的开发流程和实现造成一定的冲击。最佳实践是在引入更改、实施 AUTOSAR 的同时，对 ISO 26262 的实施开展评估和规划。这有助于确保 AUTOSAR 流程和工具能够根据新出现的需求进行扩展，以满足 ISO 等安全标准。

IEC Certification Kit 可对以下工具进行鉴定，从而支持 ISO 26262：

- Embedded Coder
- Simulink Check™
- Simulink Coverage™
- Simulink Design Verifier™
- Simulink Requirements™
- Simulink Test™
- Polyspace Bug Finder™
- Polyspace Code Prover™

IEC Certification Kit 中的交付物经 TÜV SÜD 认证 (图 15)。

[了解有关 ISO 26262 标准的更多信息。](#)

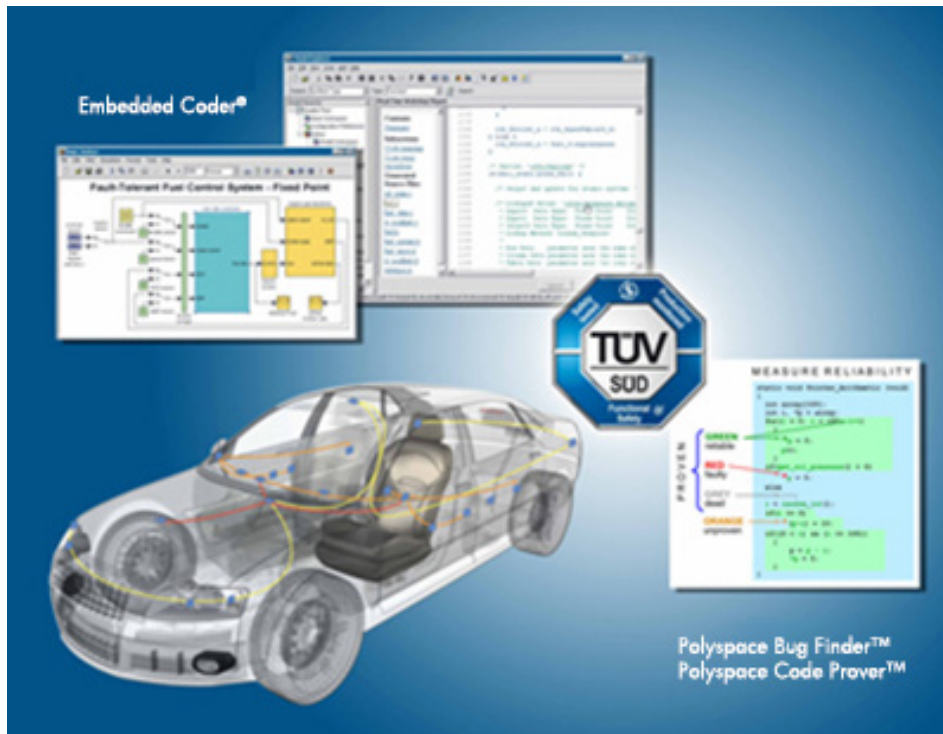


图 15.适用于 ISO 26262 的 IEC Certification Kit。

## 10.积极规划迁移

相对于汽车行业嵌入式系统的发展历史，AUTOSAR 仍是一项新兴标准，还在不断发展。MathWorks 致力于在产品版本升级过程中不断扩展 AUTOSAR 方面的工具支持。每隔半年，Simulink 和 Embedded Coder 都会推出新版，针对 AUTOSAR 及其他方面进行重要功能更新。

最佳实践是实施一种策略来管理 AUTOSAR 和 Simulink 产品的版本更新，确定升级的频率，以及选择新版本的基准。

总的来说，MathWorks 推荐实施持续升级理念。持续开展升级活动有助于让升级越来越轻松。为了促进实施这一理念，您可以参加 MathWorks 的预发行版本测试和行业模型测试，以及各类 MathWorks 研讨会、网络研讨会和会议。

阅读白皮书《[大型企业中的 MATLAB 和 Simulink 版本升级](#)》，进一步了解如何管理升级流程。

## 了解更多

利用以下资源，开始实施 AUTOSAR 开发：

- [文档](#)
- [培训](#)
- [咨询服务](#)

### 关于作者

David Jaffry 是一名 MathWorks Consulting Services 高级顾问工程师。他帮助众多汽车、航空航天、生物技术和工业自动化公司实现基于模型的设计、运用验证和确认方法以及开发嵌入式系统。加入 MathWorks 之前，他曾从事嵌入式系统的软件开发和代码验证。David 拥有布雷斯特国立工程师学院的计算机系统工程硕士学位。

Holly Keener 是一名 MathWorks Consulting Services 经理，办公地点位于密歇根。她与汽车、航空航天、国防、通信和医疗设备等各行各业的领军企业开展合作，涉及众多应用领域。她尤其擅长帮助企业实施基于模型的设计，以开发和验证嵌入式系统。Holly 与全球整车厂和零部件供应商合作，率先推进基于模型的设计在 AUTOSAR 中的应用。加入 MathWorks 之前，Holly 在汽车行业开发嵌入式系统。Holly 在密歇根大学获得电气工程学士学位和生物医学工程硕士学位。