Simulink를 이용한 AUTOSAR SW 개발
From Architecture to Design to Testing

류성연
Demo: AUTOSAR ASW Architecture Design
Agenda

- AUTOSAR Blockset Introduction
  - Adaptive Platform
  - Classic Platform

- AUTOSAR ASW Development
  - AUTOSAR ASW architecture design
  - Testing in AUTOSAR Composition Editor

- System Composer with AUTOSAR Blockset
Introduction to AUTOSAR Blockset

- Model and simulate AUTOSAR software in Simulink
  - Model AUTOSAR Classic and Adaptive software
  - Simulate AUTOSAR compositions and ECUs
  - Import and export AUTOSAR descriptions (ARXML files)
  - Create AUTOSAR software architecture

- Supports C/C++ production code generation and AUTOSAR ARXML export (with Embedded Coder)
  - Blocks for AUTOSAR library routines
  - Qualified for ISO 26262 standard (with IEC Cert Kit).

AUTOSAR Blockset Library

- Blocks to model and simulate Basic Software and AUTOSAR library routines
  - Blocks for both Classic and Adaptive Platforms
  - Out-of-the-box AUTOSAR Basic Software emulation blocks
  - Generate code that calls into Basic Software services using Basic Software caller blocks
  - Reliably generate IFX and IFL AUTOSAR library routines during code generation
AUTOSAR Component Quick Start

- **Motivation**
  - Make users more comfortable with the **bottom-up workflow**
  - AUTOSAR wizard is infrequently used by customers
  - Use same layout as Embedded Coder Quick Start

- **Design**
  - Step-by-step workflow
  - Help panel on the right of the UI
  - Additional configuration options
    - Specify Component type
    - Import properties
  - Supports Adaptive AUTOSAR
AUTOSAR Software Architecture Modeling

**AUTOSAR Composition Editor**

→ Create architecture models

- Create an AUTOSAR architecture model in a canvas for developing AUTOSAR composition and component models for the Classic Platform.
- In the architecture model:
  - Add and connect AUTOSAR compositions and components.
  - Link components to requirements (requires Simulink Requirements™).
  - Define component behavior by creating or linking Simulink models.
**AUTOSAR Today**

**Non- AUTOSAR**
- Software
- Hardware

**Classic AUTOSAR (C Code)**
- Application Software
- RTE
- Basic Software
- Hardware

**Adaptive AUTOSAR (C++ Code)**
- Adaptive Application Software
- ARA
- Services
- Basis
- High Performance Hardware

**OVER THE AIR UPDATE**

**N** Non - AUTOSAR

**C** Classic - AUTOSAR

**A** Adaptive - AUTOSAR
Support for AUTOSAR Adaptive Platform

AUTOSAR Run-time for Adaptive (ARA)

API
- OS
- Execution
- Communication

Service
- S/W CM
- Diagnostics

Adaptive AUTOSAR Foundation

High Performance Hardware/Virtual Machine
Generate Production AUTOSAR Adaptive C++ Code

Out-of-box AUTOSAR support
1. Configure Model
   ✓ Target: autosar_adaptive.tlc
   ✓ AUTOSAR Dictionary
2. Generate C++ code
Demo: Model Adaptive Software Components
Agenda

▪ AUTOSAR Blockset Introduction
  – Adaptive Platform
  – Classic Platform

▪ AUTOSAR ASW Development
  – AUTOSAR ASW architecture design
  – Testing in AUTOSAR Composition Editor

▪ System Composer with AUTOSAR Blockset
AUTOSAR Blockset and Embedded Coder for Classic Platform

Software Architecture Definition

Application Layer

Run Time Environment (RTE)

Basic Software

ECU Abstraction Layer

Microcontroller Abstraction Layer

Complex Device Drivers

ECU Hardware

Modeling and Simulation

Modeling & C Code Generation

BSW Configuration & RTE Generation

NVRAM Manager

Diagnostics Event Manager
Supported AUTOSAR Design Workflows

- **AUTOSAR authoring tool**
  - Integrate/Update
  - Export
  - Import
  - Top-down workflow

- **ARXML**
  - Export
  - Bottom-up workflow

- **Simulink**
  - Export

- **Embedded Coder**
  - C Code

Round trip workflow
Remind S/W Component Workflows

Integrate/Update

AUTOSAR authoring tool

Export

Design of software components

Export software component description (arxml) and integrate C/C++ code

Import/ update external software component description (arxml)
AUTOSAR Software Architecture
Success so far

- Design of software components
- Integration test using composition
- Export software component description (arxml) and integrate C/C++ code
- Import external software composition/component description (arxml)

Integrate/Update

Export
AUTOSAR Software Architecture

Success so far?

• Does not look like an AUTOSAR architecture model
  - Too much noise
  - Missing relevant information
• No purpose-built AUTOSAR composition editor in Simulink
• Can’t export Composition to ARXML
• Workflow requires using different tools
Now, You can start from AUTOSAR architecture design!
AUTOSAR Software Architecture
Today’s Focus - Make it easier

INTEGRATE INTO PRODUCTION

ARXML

Step 3
Integrate software description (arxml) and C/C++ code

C/C++ code

Step 1
Design of Software Composition(s)

Step 2
Design of Software Components & AUTOSAR configuration

Step 2
Design of Software Components & AUTOSAR configuration

Top down

Bottom up

AUTOSAR Composition Editor
Example: Automotive Throttle Body
Create AUTOSAR ASW Composition
Create Models from AUTOSAR ASW Component
Detailed Design using Simulink
Design and Configure for AUTOSAR SW
AUTOSAR Software Architecture

**Bottom-up Workflow**

**Step 1**
Design of Software Composition(s)

**Step 2**
Design of Software Components & AUTOSAR configuration

**Step 3**
Integrate software description (arxml) and C/C++ code

**INTEGRATE INTO PRODUCTION**

**ARXML**

**C/C++ code**

**AUTOSAR Composition Editor**

**Bottom up**

**MathWorks AUTOMOTIVE CONFERENCE 2019**
Launch Code Perspective for AUTOSAR Configuration
AUTOSAR Component Quick Start
Simulink to AUTOSAR Configuration
Assemble AUTOSAR SW-C Models into a Composition
AUTOSAR Software Architecture

Import ARXML

Step 1
Design of Software Composition(s)

Step 2
Design of Software Components & AUTOSAR configuration

Step 3
Integrate software description (arxml) and C/C++ code

INTEGRATE INTO PRODUCTION
AUTOSAR Architecture from ARXML
Agenda

- AUTOSAR Blockset Introduction
  - Adaptive Platform
  - Classic Platform

- AUTOSAR ASW Development
  - AUTOSAR ASW architecture design
  - Testing in AUTOSAR Composition Editor

- System Composer with AUTOSAR Blockset
Simulate with BSW Service Blocks and Schedule Editor
Generate AUTOSAR Code and Export ARXML
Agenda

- AUTOSAR Blockset Introduction
  - Adaptive Platform
  - Classic Platform

- AUTOSAR ASW Development
  - AUTOSAR ASW architecture design
  - Testing in AUTOSAR Composition Editor

- System Composer with AUTOSAR Blockset
What is Systems Engineering?

- Interdisciplinary approach and means to enable the realization of successful systems

- Systems engineers:
  - Ensure requirements of customers, users and other stakeholders are met
  - Design optimized system architectures
  - Validate system architecture meets requirements
  - Evaluate system level behaviors

Source: SEBoK v. 1.9.1, released 16 October 2018
What is System Architecture?

- A conceptual model that defines the structure, behavior, and other views of a system, organized in a way that supports reasoning about the system.
Consideration in System Architecture

- Simulation in a system level using Simulink

AUTOSAR Composition

Throttle body control system
System Architecture with System Composer

- System architect considers physical and logical architecture

- Mechanical & Electrical hardware

- Basic software

- Application software

- Connection
  - Hard-wires
  - CAN
  - LIN
  - Etc.
Demo: Simulation in System Composer
System Engineering with System Composer

Intuitively design system and software architectures

Add stereotypes and trade study to optimize architecture

Link requirements, architectures, design, code and test

Simulation with multi-domain environment
요구사항에서 아키텍처 설계와 시뮬레이션까지 시스템 엔지니어링을 위한 방안

ystems 엔지니어링과 모델 기반 시스템 엔지니어링은 서로 다른 곳에 대해 서로 다른 의미를 가질 수 있지만, 대부분의 정의는 시스템 설계와 요구사항 전달 프로세스를 수행하는 데 사용되는 일반의 시스템 레벨의 요구사항에서 시작하는데 그로 인해 공통적인 개념을 가집니다. 그 중 다중 시스템 아키텍처 설계에 대해 트레이드 오프 분석을 통하여 설계가 진행되고, 요구 사항이 충족되면서 확인되며 설계에 의한 시뮬레이션 기반 사항의 요구사항을 제공하려는 의도로 사용되는 것을 말합니다.
Key Takeaways

- Design and develop AUTOSAR software
- Create Simulink component models
- Author AUTOSAR composition architecture
- Support for both Classic and Adaptive Platforms
- AUTOSAR perspective to view/edit mapping
- Blocks to model and simulate Basic Software and AUTOSAR library routines
- Optimized AUTOSAR C/C++ code generation and ARXML export
**System Composer**

Design and analyze system and software architectures

- [Download a free trial](https://kr.mathworks.com/products/system-composer.html)

System Composer™ enables the definition, analysis, and specification of architectures and compositions for model-based systems engineering and software design. With System Composer, you allocate requirements while refining an architecture model that can then be designed and simulated in Simulink®.

System Composer lets you create or import architecture models that describe a system in terms of components and interfaces. You can also populate an architecture model from the architectural elements of Simulink designs or C/C++ code. You can create custom live views of the model to study specific design or analysis concerns. With these architecture models you can analyze requirements, capture properties via stereotyping, perform trade studies, and produce specifications and ICDs.
AUTOSAR 블록셋은 Simulink® 모델을 사용하여 AUTOSAR Dictionary와 클래스의 몇 개의 블록의 구조를 설정하는 AUTOSAR 소프트웨어 개발을 위한 블록을 제공합니다. AUTOSAR 블록셋의 모델은 Simulink 모델의 동작을 이해하는 데 도움을 줍니다. 또한, AUTOSAR 블록셋의 모델은 Simulink 모델의 동작을 이해하는 데 도움을 줍니다.

AUTOSAR 블록셋은 에러 및 Diagnostics를 포함한 AUTOSAR 의존성의 특정과 BSW(기본 소프트웨어) 세부사항 블록을 제공합니다. 이는 소프트웨어의 모델의 의존성과 BSW 서비스와 자료로 Simulink를 사용하여 AUTOSAR ECU 소프트웨어를 설계할 수 있습니다.

AUTOSAR 블록셋은 C, C++ 및 DBC의 스펙의 모델링을 사용하여 AUTOSAR XML 파일 내의 내용을 기반하여 Embedded Coder®의 코드를 생성하며 ISO 26262 표준(EC 26262 표준)에 따라 사용할 수 있습니다.
Thank You!