Simulink Model Architecture and Modeling Patterns for ISO 26262 Compliance

Jason Moore - MathWorks Consulting

ISO 26262 “Road Vehicles - Functional Safety”

- ISO 26262 is a functional safety standard for road vehicles

- MathWorks has seen an increased interest in ISO 26262 compliant workflows
  - Increase in System Complexity
  - Demand from ADAS and AD related applications

- ISO 26262 facilitates modern software engineering concepts such as
  - Model-Based Design

Challenges with ISO 26262

- Do I have an ISO 26262 compliant **workflow**?
- How to **efficiently** reach unit testing coverage criteria?
- How to achieve **Freedom from Interference**?
- Can we use **AUTOSAR** and meet ISO 26262 at the same time?
- Is **Simulink suitable for use** for ISO 26262?

ISO 26262-6:2018 notes Simulink and Stateflow as Suitable for Software Architecture, Design and as basis for Code Generation

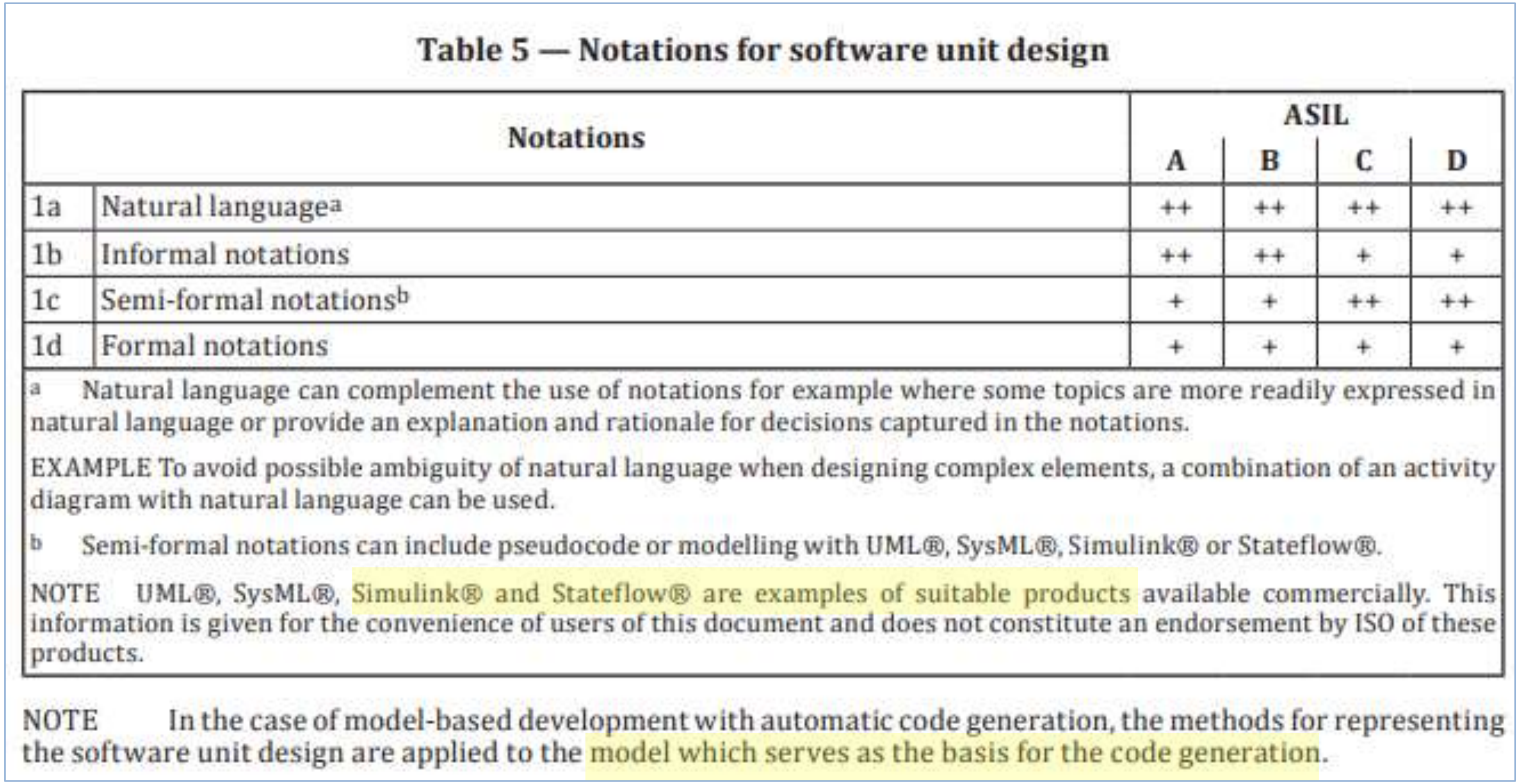


Table 5 — Notations for software unit design

Notations		ASIL			
		A	B	C	D
1a	Natural language[a]	++	++	++	++
1b	Informal notations	++	++	+	+
1c	Semi-formal notations[b]	+	+	++	++
1d	Formal notations	+	+	+	+

[a] Natural language can complement the use of notations for example where some topics are more readily expressed in natural language or provide an explanation and rationale for decisions captured in the notations.

EXAMPLE To avoid possible ambiguity of natural language when designing complex elements, a combination of an activity diagram with natural language can be used.

[b] Semi-formal notations can include pseudocode or modelling with UML®, SysML®, Simulink® or Stateflow®.

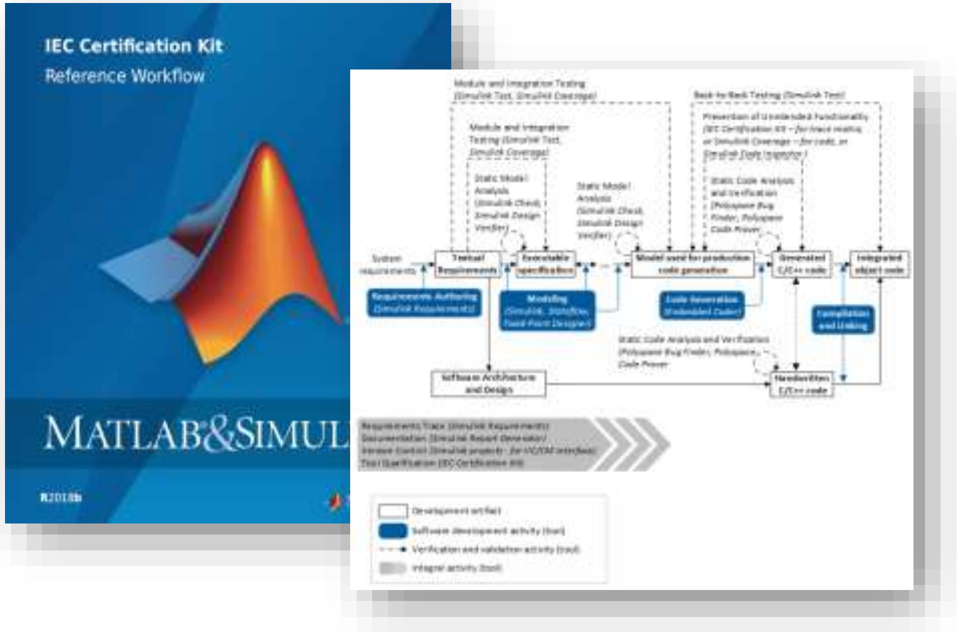
NOTE UML®, SysML®, Simulink® and Stateflow® are examples of suitable products available commercially. This information is given for the convenience of users of this document and does not constitute an endorsement by ISO of these products.

NOTE In the case of model-based development with automatic code generation, the methods for representing the software unit design are applied to the model which serves as the basis for the code generation.

Table 2 Software Architecture Design Notations has similar suitability wording for use of Simulink and Stateflow

MathWorks Support

- IEC Certification Kit
  - Model-Based Design Reference Workflow
    - Proven in use
  - Tool Qualification Package
    - Software Tool Criteria Evaluation Report
    - Software Tool Qualification
    - Tool Validation Suite


IEC Certification Kit
Reference Workflow
MATLAB&SIMULINK
R2018b
Module and Integration Testing (Simulink Test, Simulink Coverage)
Back-to-Back Testing (Simulink Test)
Module and Integration Testing (Simulink Test, Simulink Coverage)
Prevention of Unintended Functionality (IEC Certification Kit – for trace matrix or Simulink Coverage – for code, or Simulink Code Inspector)
Static Model Analysis (Simulink Check, Simulink Design Verifier)
Static Model Analysis (Simulink Check, Simulink Design Verifier)
Static Code Analysis and Verification (Polyspace Bug Finder, Polyspace Code Prover)
System requirements
Textual Requirements
Executable specification
Model used for production code generation
Generated C/C++ code
Integrated object code
Requirements Authoring (Simulink Requirements)
Modeling (Simulink, Stateflow, Fixed-Point Designer)
Code Generation (Embedded Coder)
Compilation and Linking
Static Code Analysis and Verification (Polyspace Bug Finder, Polyspace Code Prover)
Software Architecture and Design
Handwritten C/C++ code
Requirements Trace (Simulink Requirements)
Documentation (Simulink Report Generator)
Version Control (Simulink projects - for VC/CM interface)
Tool Qualification (IEC Certification Kit)
Development artifact
Software development activity (tool)
Verification and validation activity (tool)
Integral activity (tool)

KOSTAL Asia R&D Center Receives ISO 26262 ASIL D Certification for Automotive Software Developed with Model-Based Design

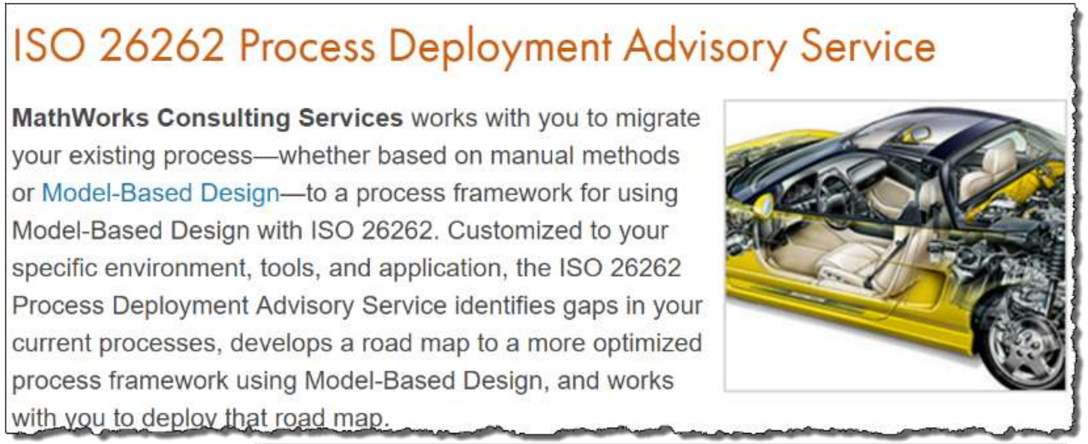
> "Without Model-Based Design, we would have needed at least 30% more time to develop and certify the ESCL application software. We saved time and effort by generating efficient code that satisfied all our speed and memory requirements."
>
> *— Cheng Hui, platform and process manager,* KOSTAL

MathWorks Support

ISO 26262 Consulting Services

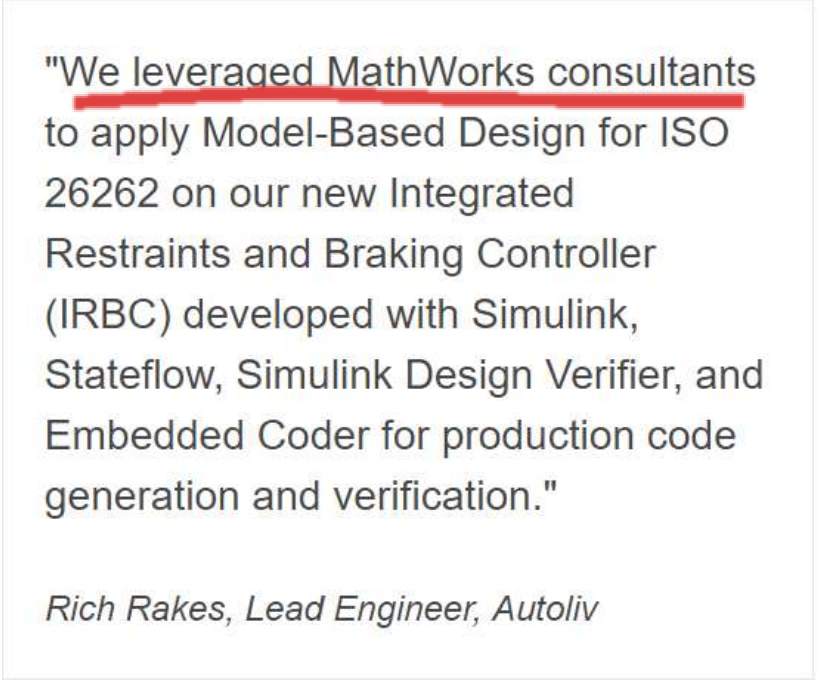
- Process establishment
  - Development Processes
  - Verification process
  - Gap analysis

- Tool qualification support
  - Analyze customer specific tools
  - Provide guidance on tool qualification activities



ISO 26262 Process Deployment Advisory Service

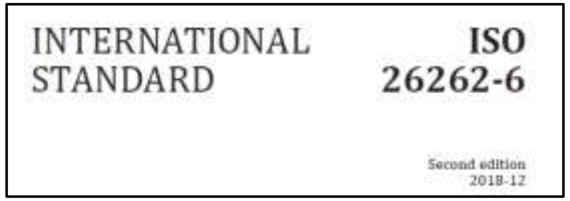
MathWorks Consulting Services works with you to migrate your existing process—whether based on manual methods or Model-Based Design—to a process framework for using Model-Based Design with ISO 26262. Customized to your specific environment, tools, and application, the ISO 26262 Process Deployment Advisory Service identifies gaps in your current processes, develops a road map to a more optimized process framework using Model-Based Design, and works with you to deploy that road map.



"We leveraged MathWorks consultants to apply Model-Based Design for ISO 26262 on our new Integrated Restraints and Braking Controller (IRBC) developed with Simulink, Stateflow, Simulink Design Verifier, and Embedded Coder for production code generation and verification."

Rich Rakes, Lead Engineer, Autoliv

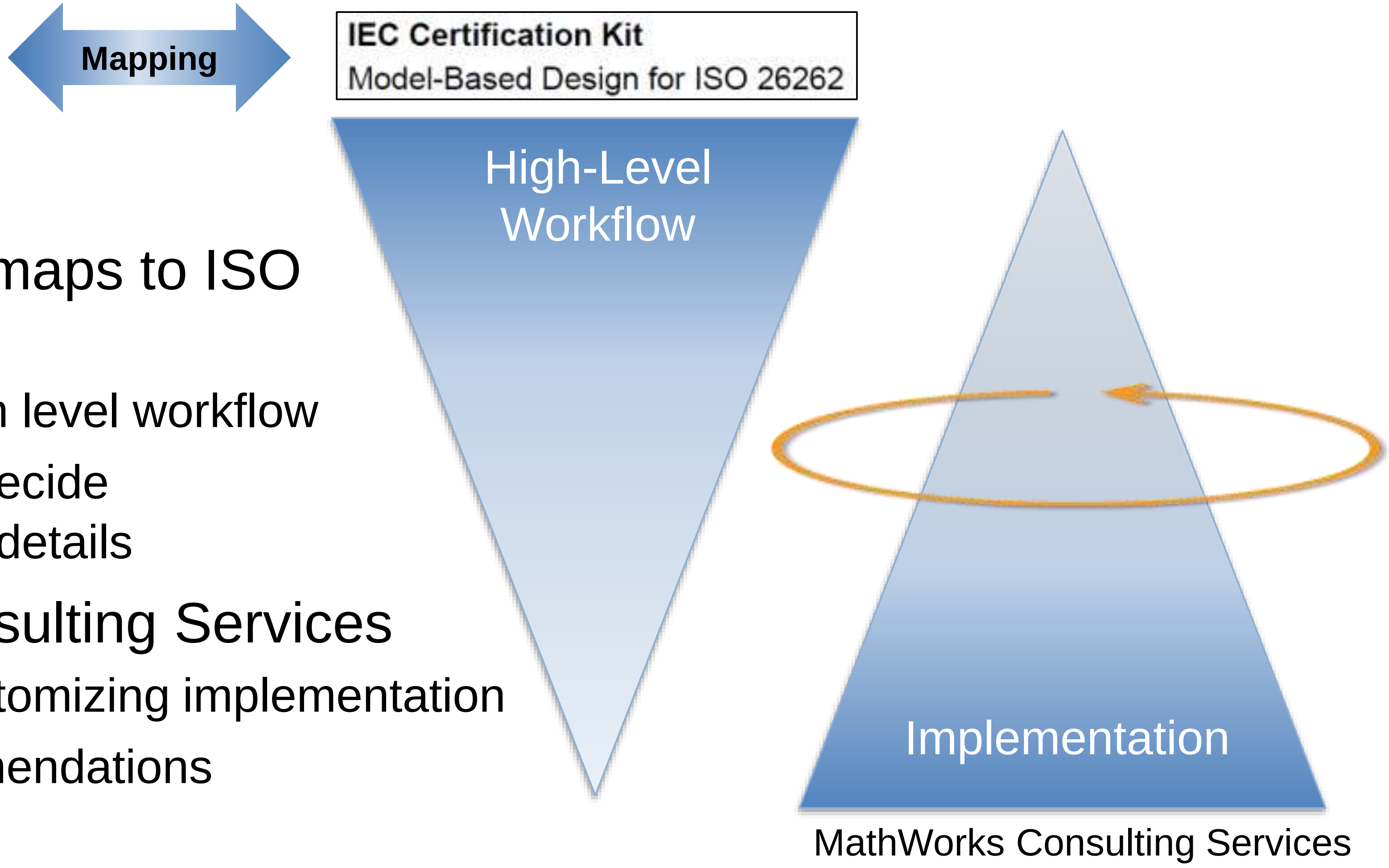
Certification Kit and Consulting Services



INTERNATIONAL STANDARD

ISO 26262-6

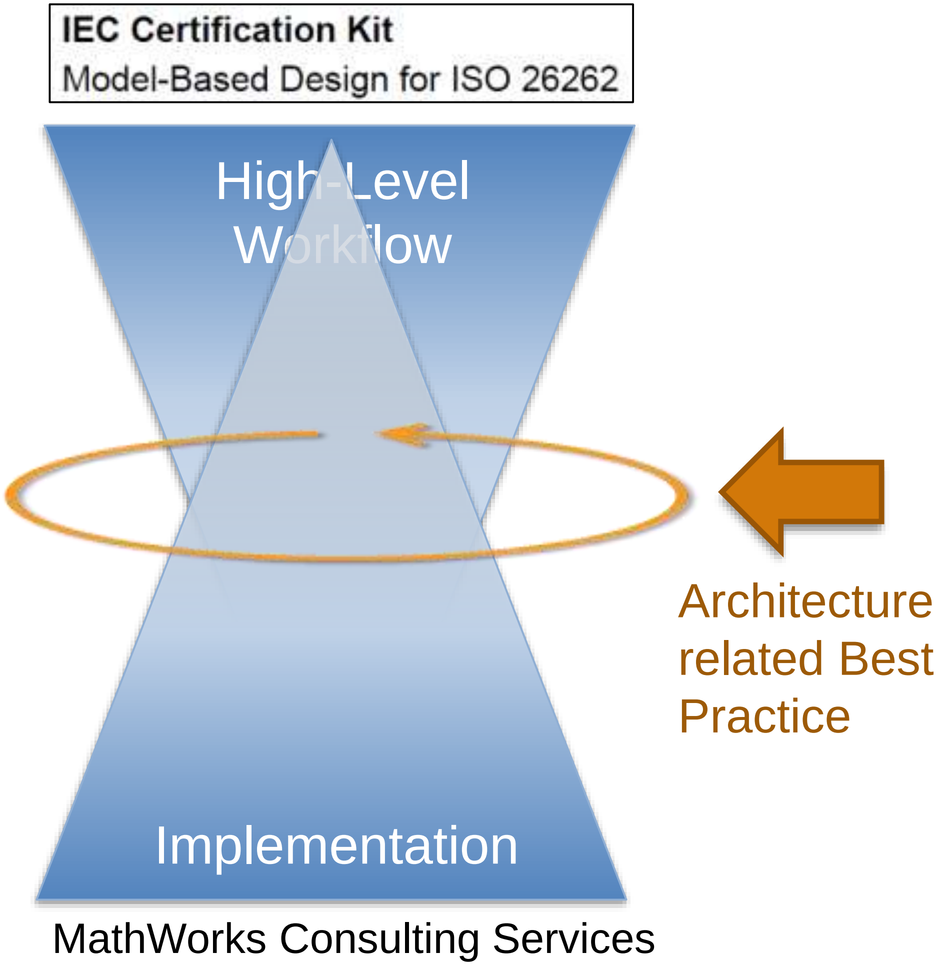
Second edition 2018-12


Mapping
IEC Certification Kit
Model-Based Design for ISO 26262
High-Level Workflow
Implementation
MathWorks Consulting Services

- Certification Kit maps to ISO 26262.
  - Focuses on high level workflow
  - Allows user to decide implementation details
- MathWorks Consulting Services
  - Focuses on customizing implementation
  - Provide recommendations

Certification Kit and Consulting Services

- Through numerous engagements, we have found a set of common **Simulink model architecture** related Best Practices.
- Best Practices assume a “top down” code generation approach.
- Example modeling patterns shown are an “enablers” to meeting ISO 26262.


IEC Certification Kit
Model-Based Design for ISO 26262
High-Level Workflow
Implementation
Architecture related Best Practice
MathWorks Consulting Services

Best Practices

- Architecture
- Signal Routing
- Data Definition
- Code Generation Configuration

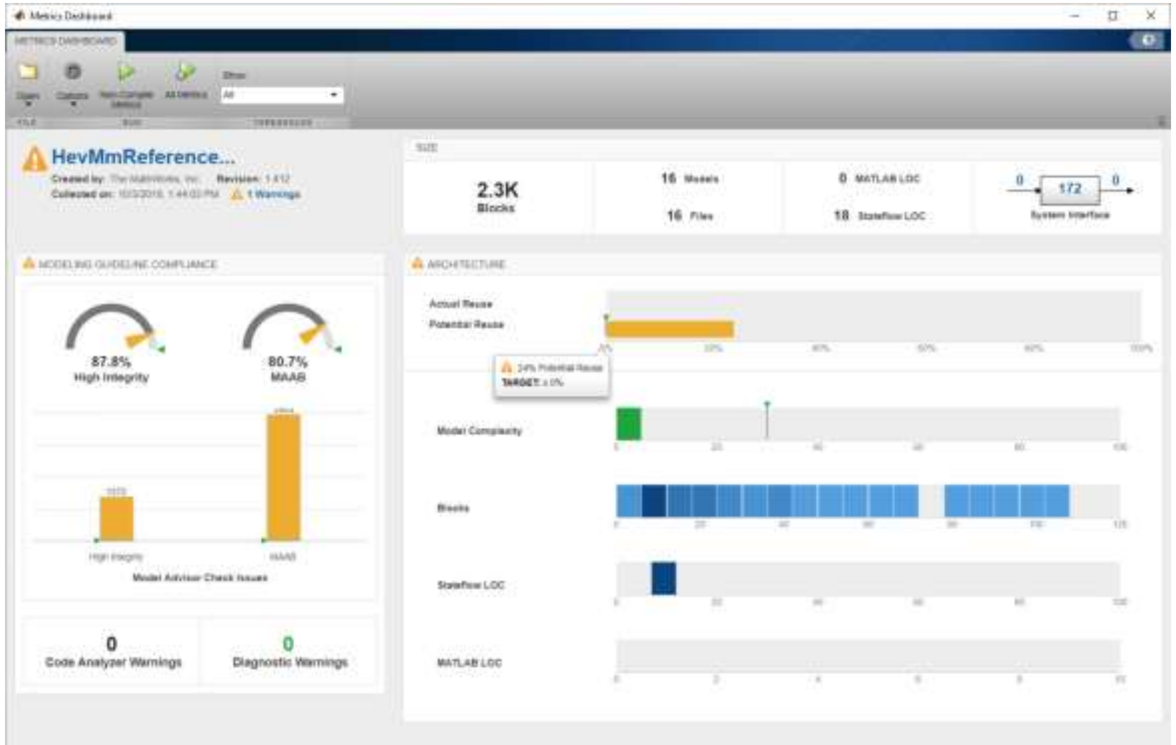
- Modeling Best Practice for ISO 26262
  - Paper
  - Please request www.mathworks.com/services/consulting/contact.html

Use Model Metrics to Monitor Unit Complexity

Architecture

- Issues:
  - Model verification gets increasingly difficult
  - Unable to efficiently achieve unit coverage
- Best Practice:
  - Define complexity metrics
    - Number of I/O
    - Reusable libraries
    - Cyclomatic complexity (<=30)*
    - Number of elements (<500)*
    - …etc.
  - Monitor coverage metrics using Continuous Integration tools.
- Reference:
  - *Paper: Model Quality Objectives
    - Authors: Jérôme Bouquet(Renault), Stéphane Faure(Valeo), Florent Fève(Valeo), Ursula Garcia(Bosch), François Guérin(MathWorks), Thierry Hubert(PSA), Florian Levy(Renault), Stéphane Louvet(Bosch), Patrick Munier(MathWorks), Pierre-Nicolas Paton(Delphi), Alain Spiewek(Delphi), and Yves Touzeau(Renault)

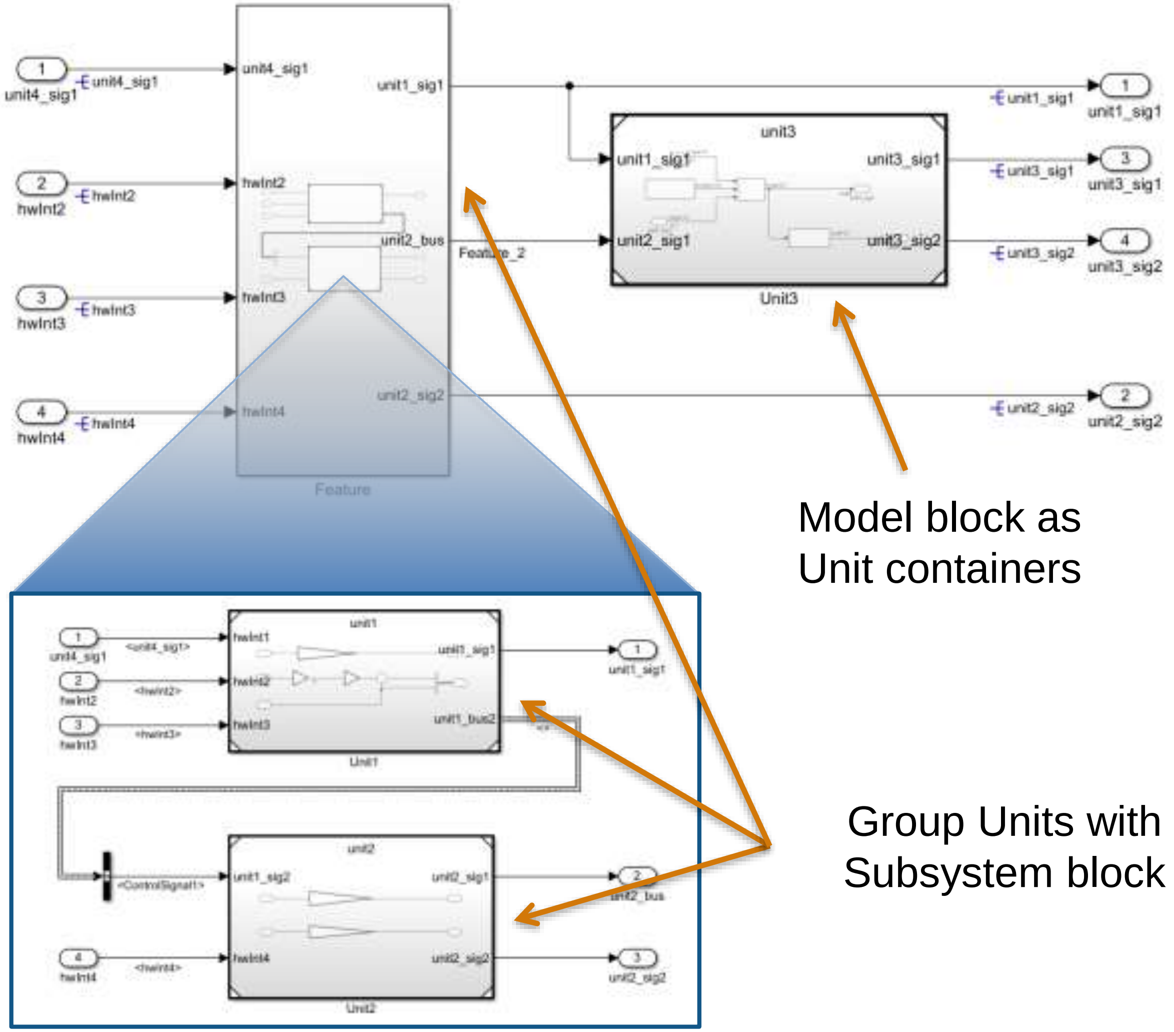
Model Metric Dashboard



Use Model Reference for Unit Level Model

Simulink Architecture

- Issues:
  - Poor modularity of algorithm (reuse)
  - Unable to preform unit level testing
  - Configuration Management difficulties
  - Unable to achieve Freedom from Interference
- Best Practice
  - Use Model Reference for unit level model
  - Group units to form functional hierarchy (features/components) with virtual Subsystems



Model block as Unit containers

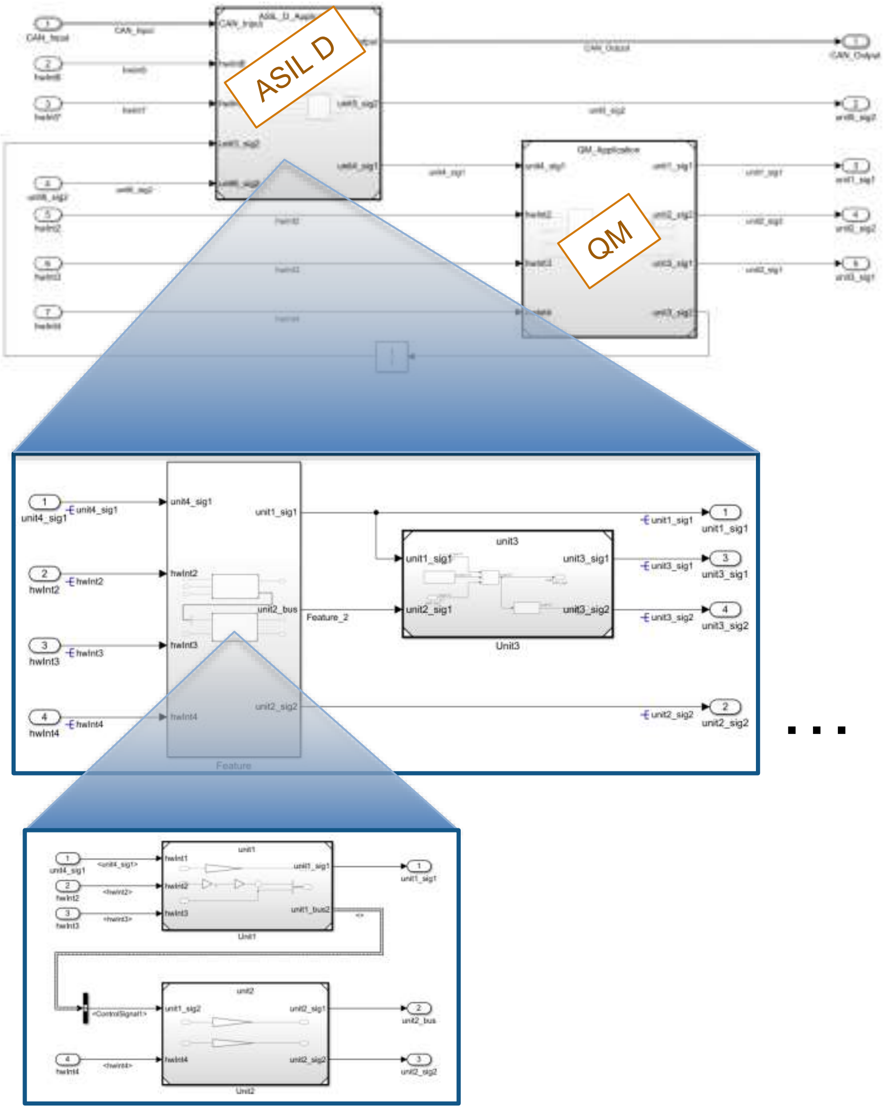
Group Units with Subsystem block

Split ASIL and QM Levels at Top Level of Control Model

Simulink Architecture

- Issues:
  - Difficulty in achieving Freedom from Interference
  - Complexity in code integration
- Best Practice:
  - Code generation should be done at as high as level as possible.

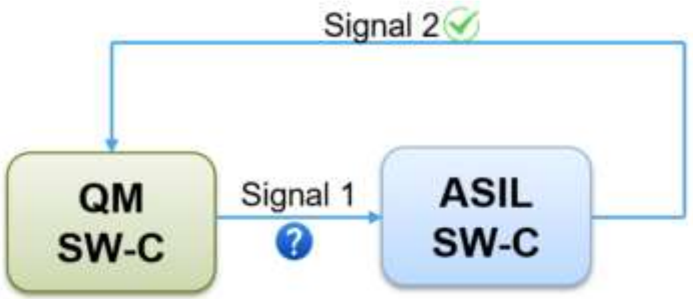
Model Hierarchy	Modeling Pattern
Top level (ASIL / QM)	Model Reference
Integration	Subsystem (multiple layers)
Unit	Model Reference



Data Protection Between ASIL and QM Levels

Code Generation Configuration

- Issues:
  - How to provide signal protection between ASIL and QM functions?

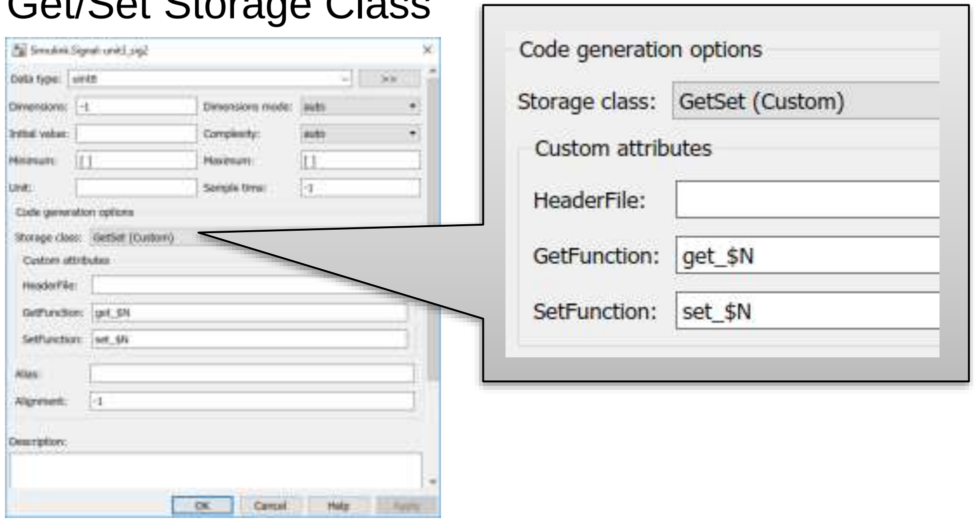


Data Protection Between ASIL and QM Levels

Code Generation Configuration

- Issues:
  - How to provide signal protection between ASIL and QM functions?
- Best Practice
  - Use Get/Set storage class for signals between ASIL and QM levels

Get/Set Storage Class



Simulink.Signal: unit3_sig2

Data type: uint8

Dimensions: -1   Dimensions mode: auto

Initial value: Complexity: auto

Minimum: [ ]   Maximum: [ ]

Unit: Sample time: -1

Code generation options

Storage class: GetSet (Custom)

Custom attributes

HeaderFile:

GetFunction: get_$N

SetFunction: set_$N

Alias:

Alignment: -1

Description:

Code generation options

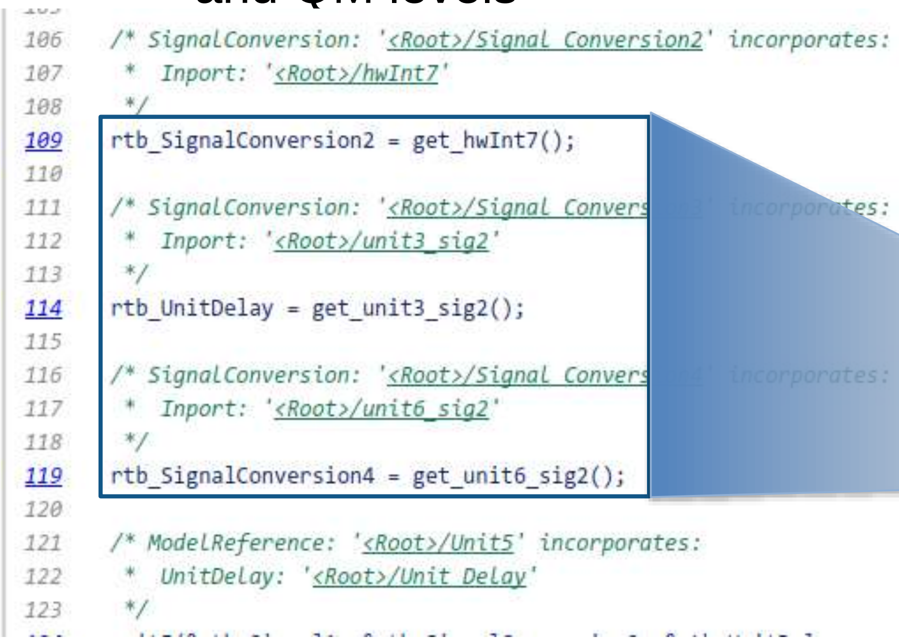
Storage class: GetSet (Custom)

Custom attributes

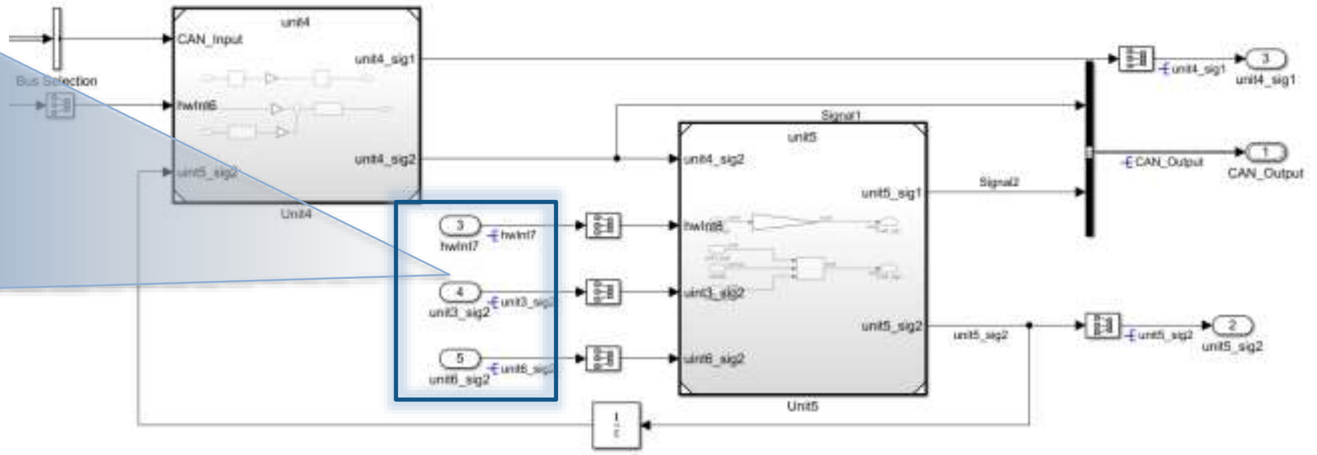
HeaderFile:

GetFunction: get_$N

SetFunction: set_$N



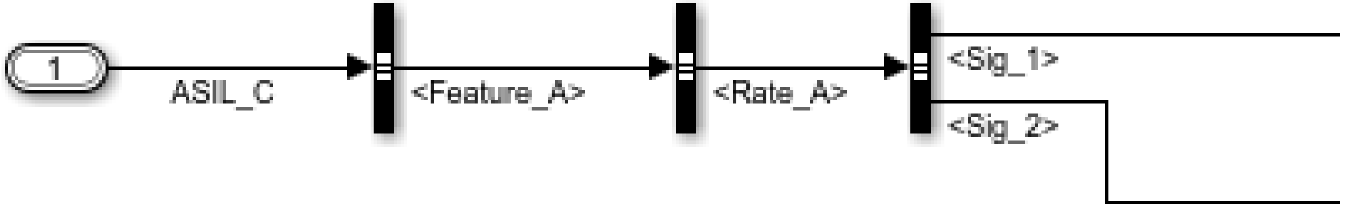
```
/* SignalConversion: '<Root>/Signal Conversion2' incorporates:
 *  Inport: '<Root>/hwInt7'
 */
rtb_SignalConversion2 = get_hwInt7();

/* SignalConversion: '<Root>/Signal Convers[illegible]' incorporates:
 *  Inport: '<Root>/unit3_sig2'
 */
rtb_UnitDelay = get_unit3_sig2();

/* SignalConversion: '<Root>/Signal Convers[illegible]' incorporates:
 *  Inport: '<Root>/unit6_sig2'
 */
rtb_SignalConversion4 = get_unit6_sig2();

/* ModelReference: '<Root>/Unit5' incorporates:
 *  UnitDelay: '<Root>/Unit Delay'
 */
```



Design Bus Hierarchy

Signal Routing

- Issues:
  - Inefficient bus segmentation
  - Inconsistent bus grouping by developers
  - Modeling difficulty from splitting and recreating bus signals
- Best Practice:
  - Bus hierarchy should be designed as a function of ASIL levels, QM, and rates at a minimum.



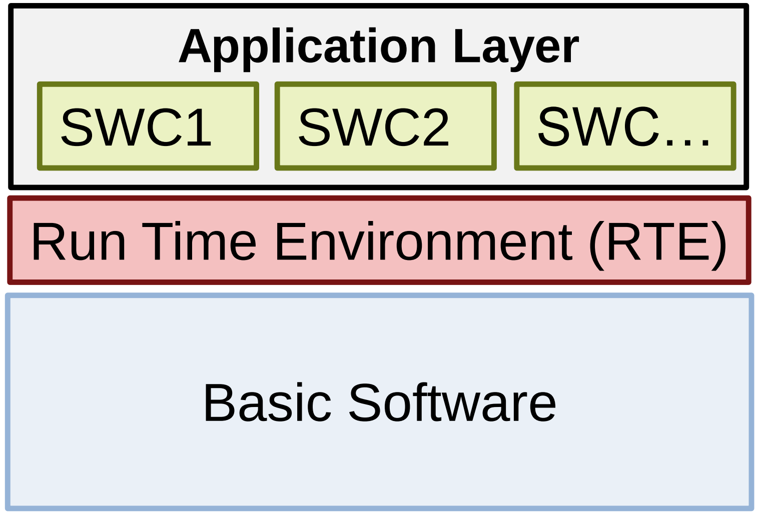
AUTOSAR Implications

- Best Practices shown are compatible to AUTOSAR

- Many of the configuration and customization can be implemented within an AUTOSAR architecture

- AUTOSAR addresses some of the issues listed above. For example:
  - Get/Set function would be implemented using Send/Receiver port with RTE protection

- Reference Workflow shown in IEC Certification Kit supports AUTOSAR architecture



Layered AUTOSAR Architecture

Summary

- Best Practice based on Consulting Engagements
- Modeling Best Practice for ISO 26262
  - Use Model Reference for Unit Level Model
  - Split ASIL and QM Levels at Top Level of Model
  - Eliminate Algorithm Content at Integration Level
  - Use Model Metrics to Monitor Unit Complexity
  - Pass Only Used Signal into Unit
  - Design Bus Hierarchy
  - Modeling Construct for Data
  - Data Protection Between ASIL and QM Levels
  - Partition Different ASIL levels and QM to Separate Memory Section
  - Use Different Name Token for Shared Utility
  - AUTOSAR Implications
- Paper
  - Please request via www.mathworks.com/services/consulting/contact.html
- ISO 26262 booth