Production Code Generation and Verification for Industry Standards

Sang-Ho Yoon
Senior Application Engineer
High-Integrity Applications

Software-based systems that are designed and maintained so that they have a high probability of carrying out their intended function

Definition: cf. Buncefield Investigation Glossary
www.buncefieldinvestigation.gov.uk/glossary.htm

Often Require Certification
DO-178C “Airborne Software - Functional Safety”

- **Functional safety standard** for airborne software
  - Concerned with production of software that performs its intended function with a level of confidence in safety that complies with airworthiness requirements

- Facilitates modern software concepts through supplements:
  - Model-Based Design
  - Object Oriented Techniques
  - Formal Methods
ISO 26262 “Road Vehicles - Functional Safety”

- **Functional safety standard** for passenger cars
  - Concerned with avoidance of unreasonable risks due to hazards caused by malfunctioning E/E systems

- Facilitates modern software engineering concepts such as
  - Modeling and simulation
  - Early verification / validation
  - Code generation
IEC 61508 “Industrial Equipment - Functional Safety”

- **Functional safety standard** for Industrial Equipment
  - Concerned with avoidance of unreasonable risks due to hazards caused by malfunctioning E/E systems

- Serves as an umbrella standard for industry specific adaptions:
  - EN 50128 - Rail
  - IEC 62304 - Medical
  - IEC 61511 - Process Control
  - Other standards and industries
Model-Based Design for Certified Systems

Support → Workflow → Tools

#1 Understand workflow to meet the standard
#2 Learn to use software tools effectively
#3 Demonstrate compliance and manage safety cases
High Integrity Workflows

Exemplary verification and validation processes for safety-related software created using Model-Based Design and production code generation.
High Integrity Workflows

With MathWorks Tools

- Simulink Verification and Validation
- Simulink Design Verifier
- Simulink Code Inspector

Module and integration testing at the model level
Equivalence testing
Prevention of unintended functionality

Textual requirements → Executable specification → Model used for production code generation → Generated C code → Object code

Modeling → Code generation → Compilation and linking

High Integrity Workflows
Demo: Code Generation using Embedded Coder
Simulink Code Inspector
Automate DO-178 Code Reviews

Independently verify that Embedded Coder generated code traces to and complies with low-level requirements

- Demonstrate that model and source code match structurally
- Provide model→code traceability data
- Eliminate / reduce manual code reviews for DO-178B software
- Same certification credits as qualified code generator
Simulink Code Inspector Overview

Model and code development

Independent code inspection

- Static verification tool, that checks the generated code against model
- Automates DO-178 Source Coder verification activities
Demo: Code Verification using Simulink Code Inspector

Function Interface Verification Results: Verified

Model To Code Verification Results: Verified

Code To Model Verification Results: Verified
Meeting DO-178 Objectives

Anticipated Certification Credits for Simulink Code Inspector with other code verification products

<table>
<thead>
<tr>
<th>Annex A Table</th>
<th>Objective</th>
<th>DO-178B Ref.</th>
<th>Software Levels</th>
<th>Anticipated Certification Credit [Tool(s)]</th>
</tr>
</thead>
<tbody>
<tr>
<td>A-5</td>
<td>(1) Source Code complies with low-level requirements</td>
<td>Section 6.3.4a</td>
<td>A, B, C</td>
<td>Full [Simulink Code Inspector]</td>
</tr>
<tr>
<td>A-5</td>
<td>(2) Source Code complies with software architecture</td>
<td>Section 6.3.4b</td>
<td>A, B, C</td>
<td>Full [Simulink Code Inspector]</td>
</tr>
<tr>
<td>A-5</td>
<td>(3) Source Code is verifiable</td>
<td>Section 6.3.4c</td>
<td>A, B</td>
<td>Full [Simulink Code Inspector]</td>
</tr>
<tr>
<td>A-5</td>
<td>(4) Source Code conforms to standards</td>
<td>Section 6.3.4d</td>
<td>A, B, C</td>
<td>Full [Polyspace MISRA-AC ACG rules checker]</td>
</tr>
<tr>
<td>A-5</td>
<td>(5) Source Code is traceable to low-level requirements</td>
<td>Section 6.3.4e</td>
<td>A, B, C</td>
<td>Full [Simulink Code Inspector]</td>
</tr>
<tr>
<td>A-5</td>
<td>(6) Source Code is accurate and consistent</td>
<td>Section 6.3.4f</td>
<td>A, B, C</td>
<td>Full (for source code based criteria) [Simulink Code Inspector, Polyspace verifier]</td>
</tr>
</tbody>
</table>
Simulink Code Inspector, Simulink Design Verifier, Simulink Report Generator, Simulink Verification and Validation, and Polyspace are qualifiable to DO-178 for all safety levels.

www.mathworks.com/products/do-178

Note: Simulink and Polyspace products were not developed using certified processes.
Embedded Coder, Simulink Design Verifier, Simulink Verification and Validation, and Polyspace are pre-qualified by TÜV SÜD to ISO 26262 for all ASILs.

Note: Simulink and Polyspace products were not developed using certified processes.
Demo: Using Cert and Qual Kits
MathWorks Support

DO-178 Process Deployment Advisory Service

Consulting services to quickly adopt Model-Based Design for DO-178

- **Objectives**
  - Identify gaps in current processes
  - Provide a roadmap to an optimized DO-178 process
  - Assist with deployment of that roadmap
  - Educate on the DO-178 standard

[Website Link](http://www.mathworks.com/services/consulting/areas/do-process-deployment.html)
Objectives

- Identify gaps in current processes
- Provide a roadmap to an optimized ISO 26262 process
- Assist with deployment of that roadmap
- Educate on the ISO 26262 standard

www.mathworks.com/services/consulting/areas/iso26262-process-deployment.html
Model-Based Design for Industry Standards

- MathWorks is committed to supporting industry standards
  - Reference workflows for using Model-Based Design in context of standards
  - Code generation and verification tools
  - Tool qualification kits
  - Process deployment advisory services
  - ...

- We look forward to collaborating with you to accelerate the transition towards certification compatible processes and tool chains for Model-Based Design
Questions?