Automated Driving Systems with Model-Based Design for ISO 26262:2018 and SOTIF

Konstantin Dmitriev
The MathWorks, Inc. – Certification and Standards Group
Agenda

- Use of simulation to satisfy ISO 21448 SOTIF objectives
- Model-Based Design methods to satisfy ISO 26262-6:2018
- Confidence in the use of the tools
ISO 21448 SOTIF – Safety of the Intended Functionality

- Deal with system limitations not related to failures
  - Insufficient robustness of sensor
  - Incomplete system requirements

- Supplement ISO 26262
SOTIF vs ISO 26262

**SOTIF Risks Identification**

**Functional Improvement and SOTIF V&V Strategy**

**Functional Description**

**HARA and Functional Safety Concept**

**Technical Safety Concept**

**System Verification Tests**

**Vehicle Validation Tests**

**Software and Hardware Development**

**SOTIF: system limitations not related to system failures**

**ISO 26262: systematic and random system failures**

SOTIF: Systematic Observability and Traceability for Functional Improvement (SOTIF) focuses on system limitations not related to system failures, whereas ISO 26262 addresses systematic and random system failures.
SOTIF Verification and Validations

- Environment for VnV
  - MIL - Model-in-the-loop
  - SIL - Software-in-the-loop
  - PIL – Processor-in-the-loop
  - HIL - Hardware-in-the-loop
  - Vehicle-level testing
    - On selected scenarios

- SOTIF Hazard Identification and Evaluation

Requirements-Based Testing

- Known Unsafe Scenarios
- Known Safe Scenarios
- Unknown Unsafe Scenarios
- Unknown Safe Scenarios

Stochastic Testing / Simulation
Driving Scenarios with MathWorks Automated Driving Toolbox

ROADS + ACTORS + ENVIRONMENTAL

RANDOMIZATION
Model-Based Design Methods to Satisfy ISO 26262:2018
ISO 26262 Functional Safety Standard

- Significant process rigor and engineering effort
- Modern methods including Model-Based Design (MBD)
- Second edition is coming in 2018
# ISO 26262-6 Methods and Model-Based Design

## Table 9 — Structural coverage metrics at the software unit level

<table>
<thead>
<tr>
<th>Methods</th>
<th>ASIL A</th>
<th>ASIL B</th>
<th>ASIL C</th>
<th>ASIL D</th>
</tr>
</thead>
<tbody>
<tr>
<td>1a</td>
<td>++</td>
<td>++</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>1b</td>
<td>+</td>
<td>++</td>
<td>++</td>
<td>++</td>
</tr>
<tr>
<td>1c</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>++</td>
</tr>
</tbody>
</table>

### Diagram

- **++** Highly Recommended
- **+** Recommended
- **o** No Recommendation

For each ASIL level (A to D), the diagram shows the coverage metrics for each test case (T-1 to T-15) with symbols indicating the level of recommendation.
Advanced Simulink-Based Workflow for ISO 26262
MBD for Modelling and Coding Guidelines

ISO 26262-6 Table 1:
- MISRA
- CERT-C
- Language Subsets
- Low Complexity
- Strong Typing
- Naming Conventions
- Style Guides
MBD for Software Architecture and Unit Design Notation

ISO 26262-6
Tables 2 & 5:
- Natural Language
- Informal Notation
- Semi-Formal Notation
MBD for SW Architecture and Unit Design Principles

Integration Testing at the EOC level (Simulink Test)

Module and Integration Testing at the model level (Simulink Test, Simulink Coverage, Simulink Design Verifier)

Static Model Analysis (Simulink Check, Simulink Design Verifier)

Module and Integration Testing at the model level (Simulink Test, Simulink Coverage, Simulink Design Verifier)

Equivalent Testing (Simulink Test)

Static Model Analysis (Simulink Check, Simulink Design Verifier)

Back to Back Testing (Simulink Test)

Static Code Analysis and Verification (Polyspace Bug Finder, Polyspace Code Prover)

Prevention of Unintended Functionality (IEC Certification Kit – for trace matrix, or Simulink Coverage – for code, or Simulink Code Inspector)

ISO 26262-6 Tables 3 & 6:
- Hierarchical structure
- Restricted size and complexity
- Restricted use of interrupts
- No dynamic objects
- No multiple name use
- No recursion
MBD for Verification of Software Architecture, Units and Integration

Integration Testing at the EOC level (Simulink Test)

Module and Integration Testing at the model level (Simulink Test, Simulink Coverage, Simulink Design Verifier)

Equivalence Testing (Simulink Test)

Static Model Analysis (Simulink Check, Simulink Design Verifier)

Static Model Analysis (Simulink Check, Simulink Design Verifier)

Back-to-Back Testing (Simulink Test)

Prevention of Unintended Functionality (IEC Certification Kit – for trace matrix, or Simulink Coverage – for code, or Simulink Code Inspector)

Static Code Analysis and Verification (Polyspace Bug Finder, Polyspace Code Prover)

Textual Requirements

Reusable requirements

Executing software

Model used for production code generation

Generated C/C++ code

Integrated object code

System requirements

Requirements Authoring (Simulink Requirements)

Modeling (Simulink, Stateflow, Fixed-Point Designer)

Code Generation (Embedded Coder)

Compilation and Linking

Handwritten C/C++ code

Software Architecture and Design

Tables 4, 7 & 10:

- Simulation (MIL), inspection, walkthrough
- Testing
  - Requirements-based
  - Back-to-back with SIL and PIL
  - Fault injection
- Static Code Analysis
- Semi-formal and formal verification
- Control and data flow analysis
MBD for Methods for Deriving Tests and Structural Coverage Metrics

ISO 26262-6
Tables 8, 9, 11 & 12:
- Analysis of requirements
- Analysis of boundary values
- Analysis of equivalence classes
- Structural coverage
  - Statement, Branch, MCDC
  - Functions, Calls
MBD for Testing of Embedded SW

ISO 26262-6:2018 Tables 13-15

- Requirements-based tests
- Fault injection tests
- Test deriving with
  - Analysis of requirements
  - Analysis of boundary values
  - Analysis of equivalence classes
- Hardware-in-the-loop
Example of Automation with Qualified MathWorks Toolchain

- Interactive Model Coverage
- Requirements Linking
- Test Automation
- Static Model Analysis
- Structural Coverage Summary
ISO 26262:2018 Highlights

- Evaluated ISO 26262:2018 updates with TÜV SÜD
  - … The certified versions of Simulink Test are suitable to be used in safety critical development regarding the draft of the second edition of ISO 26262…

- Clarifications of some Model-Based Design aspects
  - SW design review at model level
  - Coverage analysis at model level

- More focus on static verification
  - Testing -> Verification
ISO 26262-6:2018 and Model-Based Simulink Workflow

ISO Methods

Model-Based Workflow
Confidence in the Tools Use for ISO 26262

Pre-qualification based on reference use cases / workflows

Independent Assessment

Certification Kit

Tool User

Project-specific adaptation
## ISO 26262-8 Tool Qualification Methods

<table>
<thead>
<tr>
<th>Method</th>
<th>TCL 2</th>
<th>TCL 3</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ASIL A</td>
<td>ASIL B</td>
</tr>
<tr>
<td>Increased confidence from use</td>
<td>++</td>
<td>++</td>
</tr>
<tr>
<td>Evaluation of the tool development process</td>
<td>++</td>
<td>++</td>
</tr>
<tr>
<td>Validation of the software tool</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>Development in compliance with a safety standard</td>
<td>+</td>
<td>+</td>
</tr>
</tbody>
</table>
ISO 26262-8 Tool Certification Artifacts

- Reference workflow with conformance demonstration template
- Evidences of independent assessment
  - Assessment certification report
  - Certificate
- Pre-filled templates for qualification artifacts
  - Conformance Demonstration Template
  - Tool Qualification Package
- Validation test suite setup files
Summary

- Simulation is the key technology to comply with SOTIF objectives

- Model-Based Design enables you to comply with ISO 26262:2018 automating development and verification

- ISO 26262-8 tools qualification process provides confidence in the use of the tools