Model-Based Design for High Integrity Software and Hardware
Agenda

- Relevant standards
- DO workflow – Common Elements
- DO-178B - Software Considerations and Workflows
- DO-254 – Hardware Considerations and Workflows
- Additional Topics
Standards Background

System Development Processes (ARP 4754)

Benefits of Model-Based Design

- Use models to validate and verify requirements and designs early in the process
- Re-use tests throughout design cycle
- Automatically generate design and verification artifacts
- Streamline process by qualifying verification tools
Agenda

- Relevant standards

- **DO workflow – Common Elements**

- **DO-178B - Software Considerations and Workflows**

- **DO-254 – Hardware Considerations and Workflows**

- Additional Topics
DO Workflow Example

- Requirements
- Model
- Conformance
- Verify
- Validate
- Trace
DO Workflow Example

Trace
Simulink Verification and Validation: Requirements Management Interface

Validate

Requirements

Simulink & Stateflow

Model

* DO Qualifiable Tool
Requirements linking and traceability

- Bi-directional linking with requirements
  - For Simulink and Stateflow
  - Requirements consistency checks
  - Extensibility API
  - Report generation

- Links to Documents and Requirements Management Packages.

IBM DOORS
ReqTracer
Microsoft Word
Microsoft Excel
PDF
HTML
DO Workflow

- Trace
- Validate

Requirements

Simulink & Stateflow

Model

Conformance Model Advisor*

* DO Qualifiable Tool
Simulink Model Advisor

- Model Advisor is used to
  - Enforce model standards and best practices
  - Detect modeling and code generation issues
  - Pre-defined sets of checks for DO-178B and MAAB Style Guides
  - Automated report generation
DO Workflow Example

- Trace
- Validate
- Conformance

Requirements → Model

Verify
SystemTest*
Simulink Design Verifier:
Property Proving
Model Coverage

* DO Qualifiable Tool
SystemTest

- Manage tests and analyze results for system verification and validation.

Simulink System Model
Simulink Design Verifier
Property Proving & Test Generation

- Property Proving
  - Functional testing
    - Property proving
    - Generates an example of a violation
  - Produces detailed analysis reports

- Test Generation
  - Automatically generates test vectors for model coverage
  - Detects unreachable states
  - Saves test vectors and generates report
  - Uses formal methods, not simulation

Chapter 2. Test/Proof Objectives
Table of Contents
Status
Verify True Output

Status
Table 2.1. Objectives Falsified with Counterexamples
<table>
<thead>
<tr>
<th>#</th>
<th>Type</th>
<th>Model Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Assert</td>
<td>Assertion</td>
<td>Assertion &quot;Assertion&quot;</td>
</tr>
</tbody>
</table>

Verify True Output
Objectives of: Assertion
<table>
<thead>
<tr>
<th>#</th>
<th>Status</th>
<th>Test Cases</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Falsified</td>
<td>TC 1</td>
<td>Assert</td>
</tr>
</tbody>
</table>

Chapter 3. Test Cases / Counter Examples
Table of Contents
Counterexample 1

Counterexample 1
Summary
Length: 0 Seconds (2 sample periods)
DO Workflow Example

Trace
Simulink Verification and Validation: Requirements Management Interface

Validate
Requirements
Simulink & Stateflow

Conformance
Model Advisor*

Verify
SystemTest*
Simulink Design Verifier: Property Proving Model Coverage

* DO Qualifiable Tool
Agenda

- Relevant standards
- DO workflow – Common Elements
  - DO-178B - Software Considerations and Workflows
  - DO-254 – Hardware Considerations and Workflows
- Additional Topics
DO-178B Workflow Example

- Validate
- Trace
- Conformance
- Source Code
- Object Code
- Model
- Conformance
- Verify
- Verify
- Verify
- Verify
DO-178B Workflow Example

Validate

Requirements

Verify

Conformance

Model

Source Code

Trace

Model/Code Trace Report

Real-Time Workshop®
Embedded Coder™
Real-Time Workshop® Embedded Coder

- Automatically generates C code from Simulink® and Stateflow® models
- Code is ANSI/ISO-C compliant
Model-to-Code and Code-to-Model Traceability

Simulink Verification and Validation

Real-Time Workshop Embedded Coder

Traceability Report

Eliminated / Virtual Blocks

<table>
<thead>
<tr>
<th>Block Name</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;Root&gt;/Build ERT</td>
<td>Empty SubSystem</td>
</tr>
<tr>
<td>&lt;Root&gt;/Mux</td>
<td>Mux</td>
</tr>
<tr>
<td>&lt;Root&gt;/Scope</td>
<td>Eliminated unused block</td>
</tr>
<tr>
<td>&lt;Root&gt;/View RTW</td>
<td>Empty SubSystem</td>
</tr>
</tbody>
</table>

Traceable Blocks

<table>
<thead>
<tr>
<th>Block Name</th>
<th>Code Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;Root&gt;/INC2</td>
<td>rtwdemo_hyperlinks.c:36</td>
</tr>
<tr>
<td>&lt;Root&gt;/INC3</td>
<td>rtwdemo_hyperlinks.c:37</td>
</tr>
<tr>
<td>&lt;Root&gt;/LIMIT</td>
<td>rtwdemo_hyperlinks.c:44</td>
</tr>
<tr>
<td>&lt;Root&gt;/RESET</td>
<td>rtwdemo_hyperlinks.c:54</td>
</tr>
<tr>
<td>&lt;Root&gt;/RelOpt</td>
<td>rtwdemo_hyperlinks.c:43</td>
</tr>
<tr>
<td>&lt;Root&gt;/Sum</td>
<td>rtwdemo_hyperlinks.c:35</td>
</tr>
<tr>
<td>&lt;Root&gt;/Switch</td>
<td>rtwdemo_hyperlinks.c:55</td>
</tr>
</tbody>
</table>
DO-178B Workflow Example

Validate

Requirements

Model

Source Code

Conformance

PolySpace*

Trace

Trace

Verify
PolySpace

- Verification of C/C++ and Ada code
- Detects run-time errors
- Streamlines high integrity DO-178B workflows
  - Rule checking features (MISRA-C and JSF++)
  - Source code color scheme
  - DO-178B artifact generation capabilities
  - Qualification kit available
DO-178B Workflow Example

- **Validate**
- **Trace**
- **Conformance**
- **Verify**

1. Requirements
2. Model
3. Source Code
4. Object Code

**Verify**
- SystemTest*
  - Embedded IDE Link

**Verify**
- SystemTest*
  - Simulink Design Verifier: Test Generation
  - Embedded IDE Link
Simulink:

- Model in simulation and code on the processor running in parallel

PIL also provides execution profiling, code coverage reports, and interactive debugging.
DO-178B Workflow Summary

Trace
- Simulink Verification and Validation: Requirements Management Interface
- Model/Code Trace Report

Validate
- Requirements
- Model
- Source Code
- Object Code

Conformance
- PolySpace*
- Model Advisor*
- Simulink Design Verifier: Property Proving Model Coverage
- Simulink Design Verifier: Test Generation

Verify
- SystemTest*
- Embedded IDE Link

* DO Qualifiable Tool
Agenda

- Relevant standards

- DO workflow – Common Elements

- DO-178B - Software Considerations and Workflows

- DO-254 – Hardware Considerations and Workflows

- Additional Topics
DO-254 Workflow Example

- High Level Requirements
  - Conceptual Model
    - Detailed Design (RTL)
      - Detailed Design (Net-List)
        - Hardware
  - Trace
  - Validate
  - Conformance
  - Verify
  - Verify
  - Verify
  - Verify
  - Verify

- Trace
DO-254 Workflow Example

1. High Level Requirements
2. Conceptual Model
3. Simulink
4. HDL Coder
5. Detailed Design (RTL)

Trace: Model/Code Trace Report

Validate
Conformance
Verify

DO-178B  DO-254
HDL Code Generation with Simulink HDL Coder

- Simulink HDL Coder
  - Generate behavioral HDL
  - Readable and traceable to requirements
  - Target-Independent
  - Bit Accurate/Cycle True
  - Customizable via options and Control Files

- Full model support
  - Simulink (datapath)
  - Stateflow® (control logic)
  - Embedded MATLAB
Model-to-HDL and HDL-to-Model Traceability

Simulink Verification and Validation

- Use the Traceability Report section of the Simulink HDL Coder HDL generation report to review mapping
DO-254 Workflow Example

1. High Level Requirements
   - Trace
   - Conformance
   - Validate

2. Conceptual Model
   - Trace
   - Validate

3. Detailed Design (RTL)
   - Trace
   - Conformance
   - Verify

4. Detailed Design (Net-List)
   - Verify

* DO Qualifiable Tool
EDA Simulator Link™ block Brings Together Leading Tools for Modeling and HDL Simulation

- EDA Simulator Link™ block is a fast, bidirectional cosimulation interface for system-level functional verification using ModelSim, Incisive or Discovery

- Benefit: Reuse test environment in the executable specification to verify the implementation.
DO-254 Workflow Example

EDA Partner Tools

High Level Requirements -> Conceptual Model

Conceptual Model -> Detailed Design (RTL)

Detailed Design (RTL) -> Detailed Design (Net-List)

Detailed Design (Net-List) -> Hardware

Trace

Validate

Conformance

Verify

SystemTest™*
Vendor HIL Solution

EDA Simulator Link
HDL Simulator

SystemTest™*
EDA Simulator Link
HDL Simulator

* DO Qualifiable Tool
DO-254 Workflow Example (Partners)

**EDA Partner Tools**

- **Trace**
  - Simulink® Verification and Validation™: Requirements Management Interface
- **Trace**
  - Model/Code Trace Report

**High Level Requirements**

- **Simulink & Stateflow**

**Conceptual Model**

- **Simulink HDL Coder**

**Auto Generated HDL**

- **Conformance Model Advisor**

**Detailed Design (Net-List)**

- **Hardware**

**CoSim / HIL**

- **Verify** SystemTest™*
  - EDA Simulator Link
  - HDL Simulator
- **Verify** SystemTest™*
  - Vendor HIL Solution
- **Verify** SystemTest™*
  - Property Proving Model Coverage

* DO-254 Qualifiable Tool
Agenda

- Relevant standards
- Benefits of Model-Based Design
- DO-178B - Software Considerations and Workflows
- DO-254 – Hardware Considerations and Workflows
- Additional Topics
DO Qualification Kit

- Tool Qualification Plan and Tool Operational Requirements
- Test case models and code, test procedures, and expected results
- Traceability tables mapping test cases to requirements
- Qualification materials for Simulink verification, validation, and test tools
- Qualification materials for PolySpace code verification tools