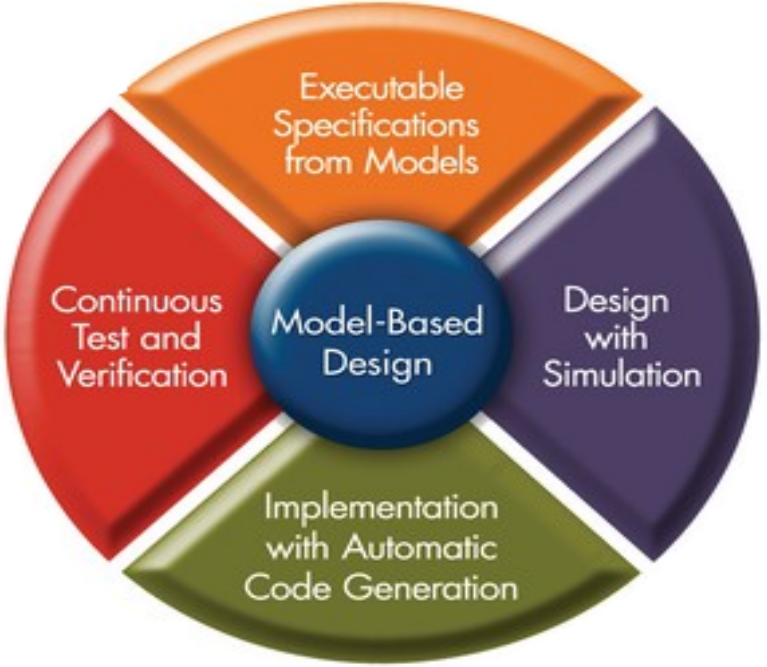
Certifiable Production Code Development



David Owens

Rolls-Royce Control Systems

Trusted to deliver excellence

Rolls-Royce

Overview

The Trent 7000 EMU is the first time that Rolls-Royce has gone through the DO-178C (ED-12C) certification process for certifying software in an airborne system using the Model Based Supplement DO-331 (ED-218)

This presentation discusses the Model Based Design method used during the application software development and the MathWorks tools which enabled the workflow.

Contents

- **Background**
  - What is the EMU
  - What steps are required for DO-178C certification
- **Development Process**
  - Model development from requirements
  - Model verification activities
  - Code production and verification
  - Tool Qualification
- **Future Enhancements**
  - Further tool adoption

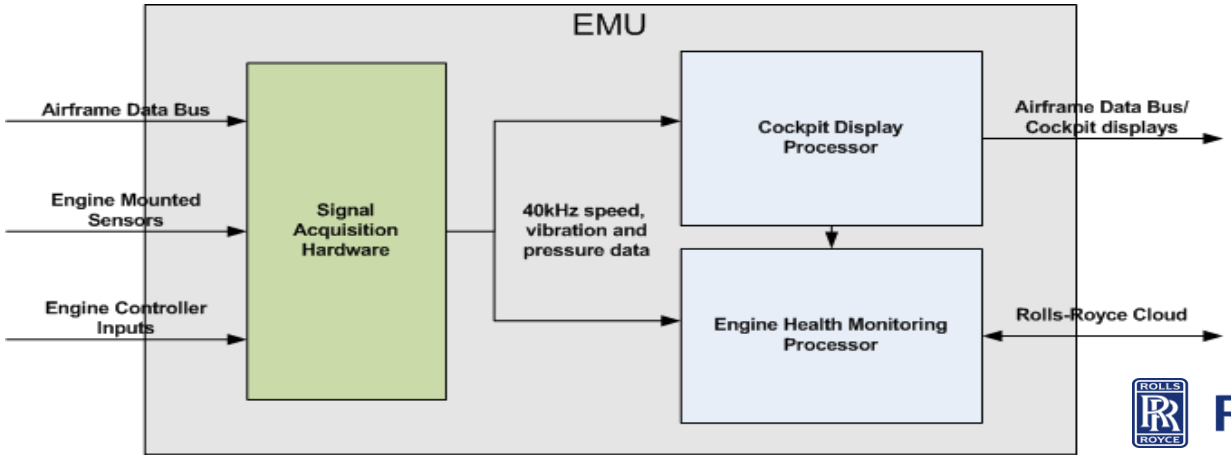
Background

What is the EMU

The **EMU** is the **E**ngine **M**onitoring **U**nit
Also known as the **EVHMU** **E**ngine **V**ibration and **H**ealth **M**onitoring **U**nit.

The unit has two primary functions:-

1. A high integrity partition which interfaces with the cockpit displays.
2. A engine health monitoring partition which sends data in the form of reports back to Rolls-Royce to feed into the TotalCare® aftermarket service.



What is the EMU

From 2006 to 2016:

Hardware: Supplied by sub-tier supplier

Cockpit Displays Software: Specified by Rolls-Royce, supplied by sub-tier supplier

EHM Software : Specified by Rolls-Royce, supplied by sub-tier supplier

Engines fitted with EMUs: Trent900, 1000, XWB

From 2016 onwards:

Hardware: Rolls-Royce Control Systems

Cockpit Displays Software: Rolls-Royce Control Systems

EHM Software: Rolls-Royce Control Systems

Engines (to be) fitted with EMUs: All future Civil Large, and Medium Corporate engines ……

A software development method which could deliver to the tight engine timescales was required:

The solution:- **Model Based Design**

What is the EMU

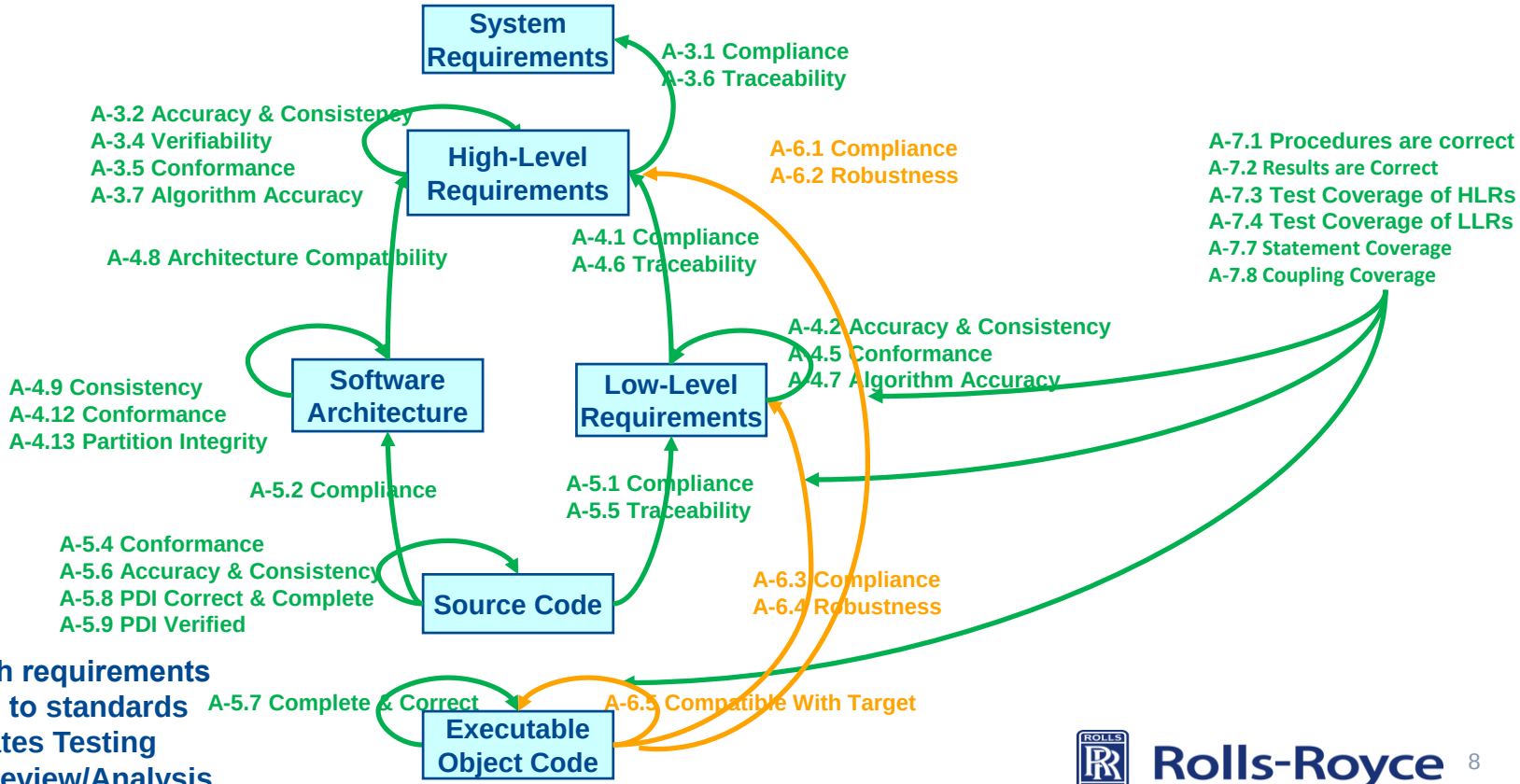
Aircraft level certification for large aeroplanes requires that:-

"An indicator to indicate rotor system unbalance" is displayed to the pilot for turbo-jet engine-powered aeroplanes. [CS-25 directive 25.1305 d3]

As the pilot makes decisions based on information provided by the EMU the HW and SW need to be developed to the appropriate **D**esign **A**ssurance **L**evel (DAL).

The EMU is designated as a DAL C unit as misleading indication could lead to increased pilot workload and as such it must be certified by EASA against the standard "Software Considerations In Airborne Systems and Equipment Certification" also known as DO-178 (ED-12)

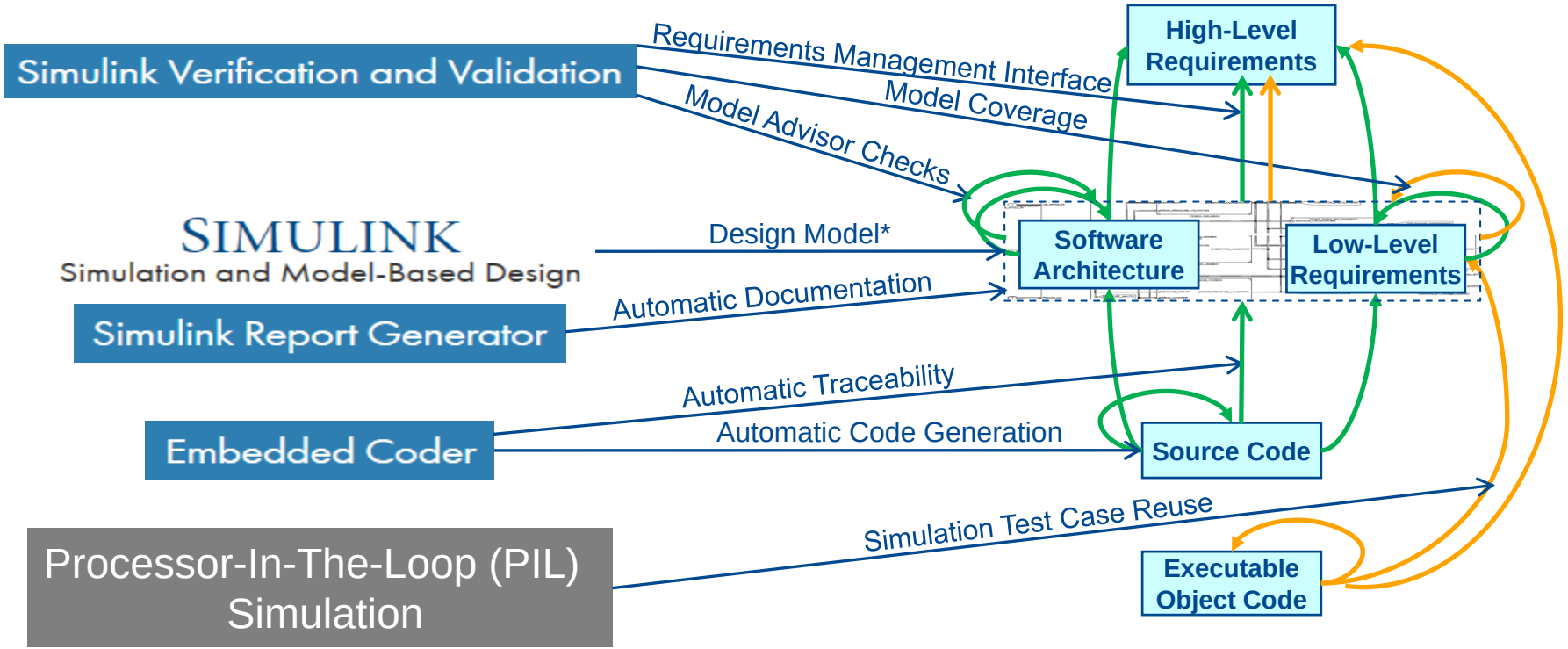
What steps are required for DO-178C certification



"Compliance" with requirements
"Conformance" to standards
Orange indicates Testing
Green indicates Review/Analysis

Rolls-Royce

What steps are required for DO-178C certification



"Orange indicates Testing
Green indicates Review/Analysis

* A Design Model includes low-level requirements and/or architecture [DO-178C MB.1.6.2]

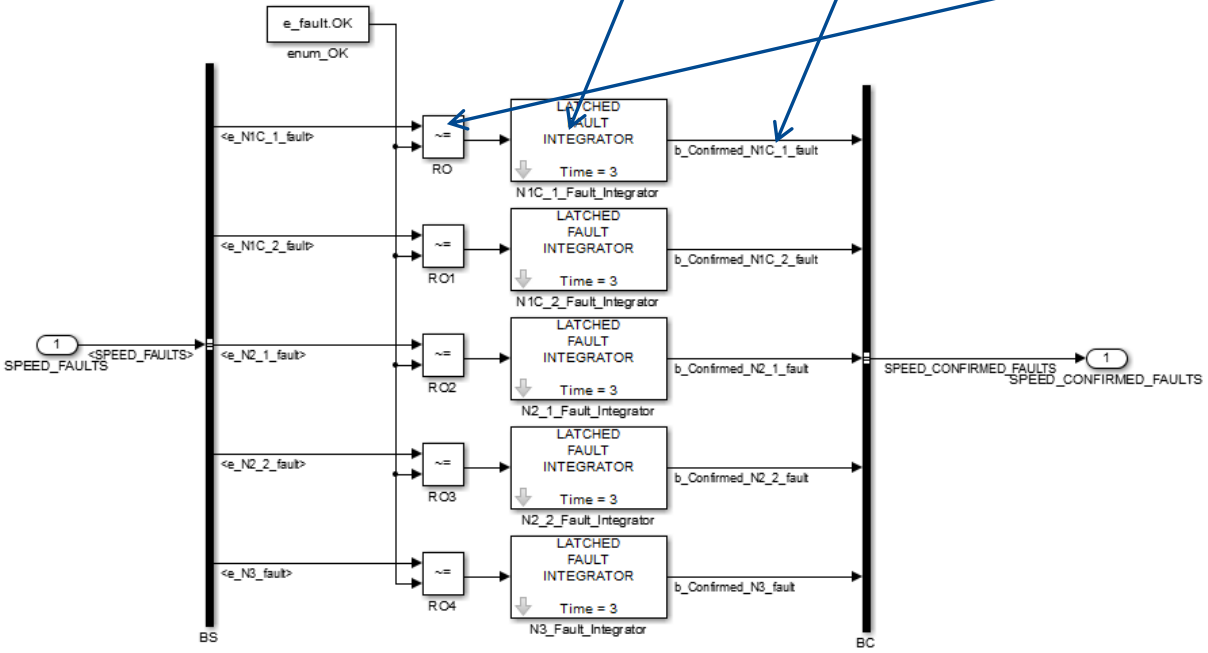
Rolls-Royce

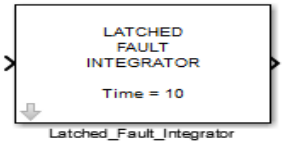
Development Process

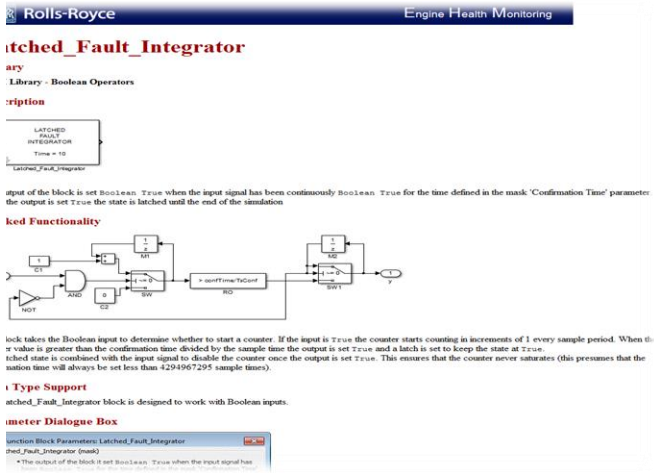
Model development from requirements

Starting with an example High Level Requirement:

Rqmt : The Boolean fault status associated with the N1C-1 speed signal shall be confirmed and latched as being True if the N1C-1 fault status is continuously set to any other state than OK for more than 3 seconds.



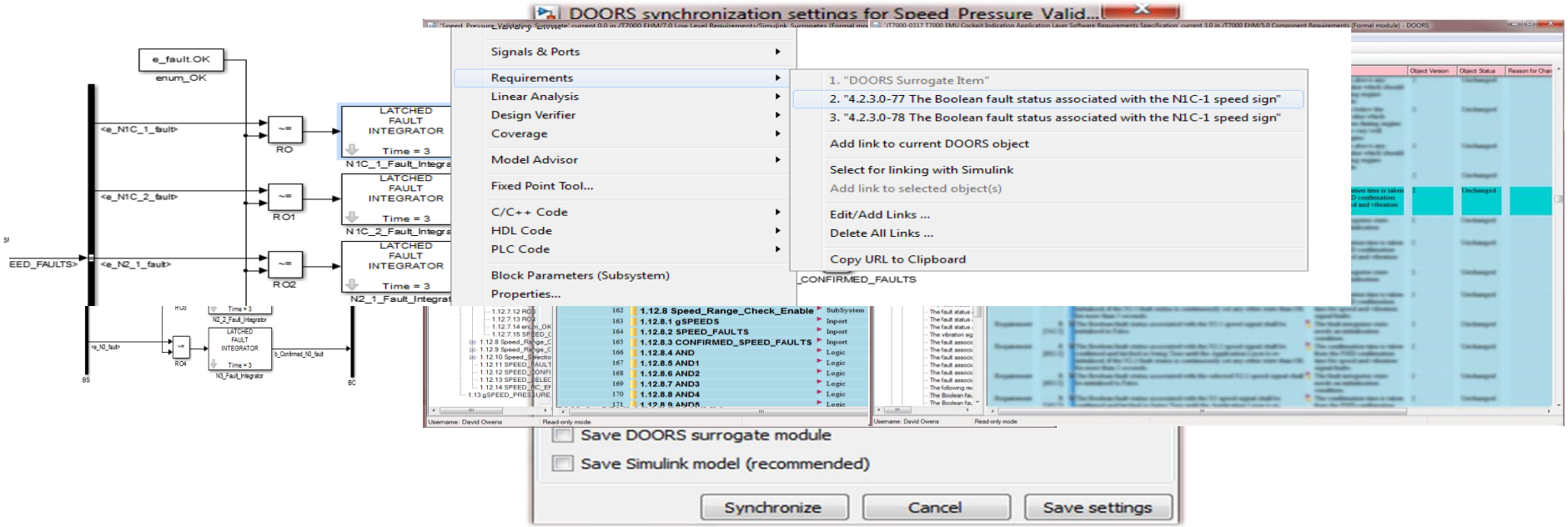




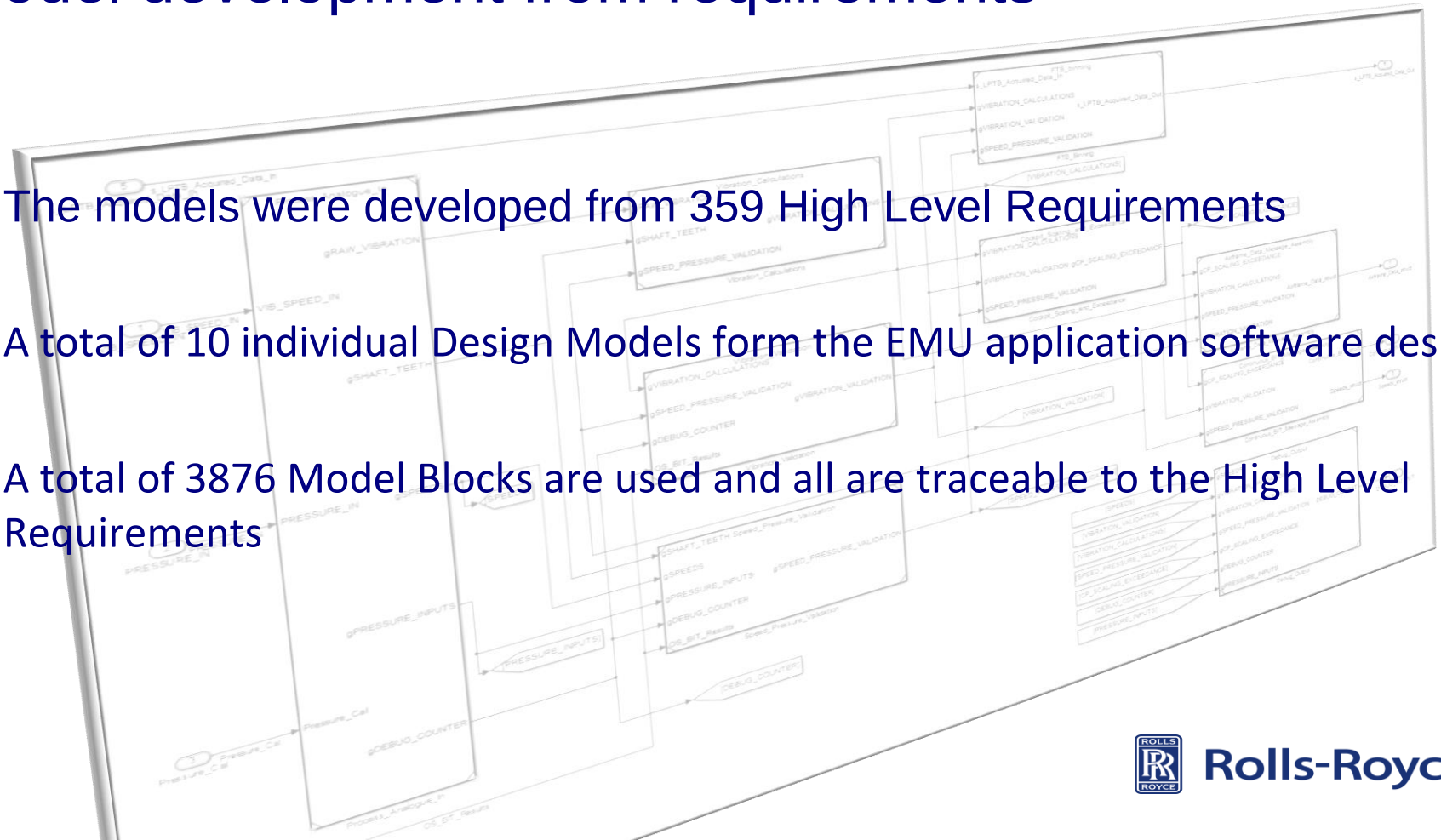
Model development from requirements

Adding traceability between the model and the requirement.

The **Requirements Management Interface** from the **Verification and Validation Toolbox** is used to add bidirectional traceability.



Model development from requirements

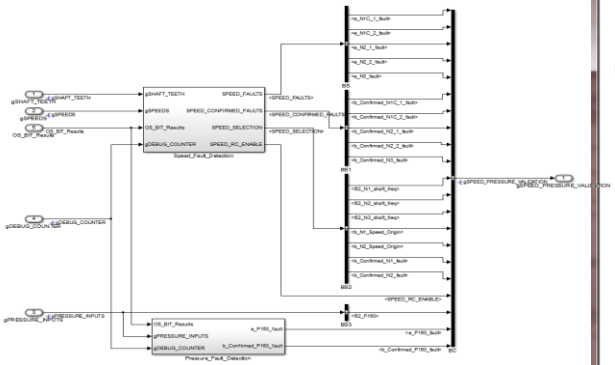


- The models were developed from 359 High Level Requirements
- A total of 10 individual Design Models form the EMU application software design
- A total of 3876 Model Blocks are used and all are traceable to the High Level Requirements

Model verification activities

Check that the model conforms to standards.

The **Model Advisor Checks** from the **Verification and Validation Toolbox** are used.



Model Advisor - Speed_Pressure_Validation

File Edit View Favorites Tools Help

Model Advisor Report - Speed_Pressure_Validation.slx

Simulink version: 8.1 **Model version: 1d**

System: Speed_Pressure_Validation **Current run: 07-Nov-2016 15:28:13**

Run Summary

Pass	Fail	Warning	Not Run	Total
51	0	2	10	63

Modeling Standards for DO-178C/DO-331

Display model version information

Display model configuration and checksum information.

Model configuration and checksum information

Attribute	Value
Model Version	1d
Author	EMU_DEV_PC1
Date	Wed Nov 02 11:50:32 2016
Model Checksum	1684242041 1349583278 410703400 385426520

Check safety-related optimization settings

Check optimization settings in the model configuration that might impact safety.

Verify 'Block reduction' setting
Check whether the **Block reduction** check box is cleared.

Passed
Optimization > Block reduction is cleared.

Verify 'Implement logic signals as boolean data (vs. double)' setting
Check whether the **Implement logic signals as boolean data (vs. double)** check box is selected.

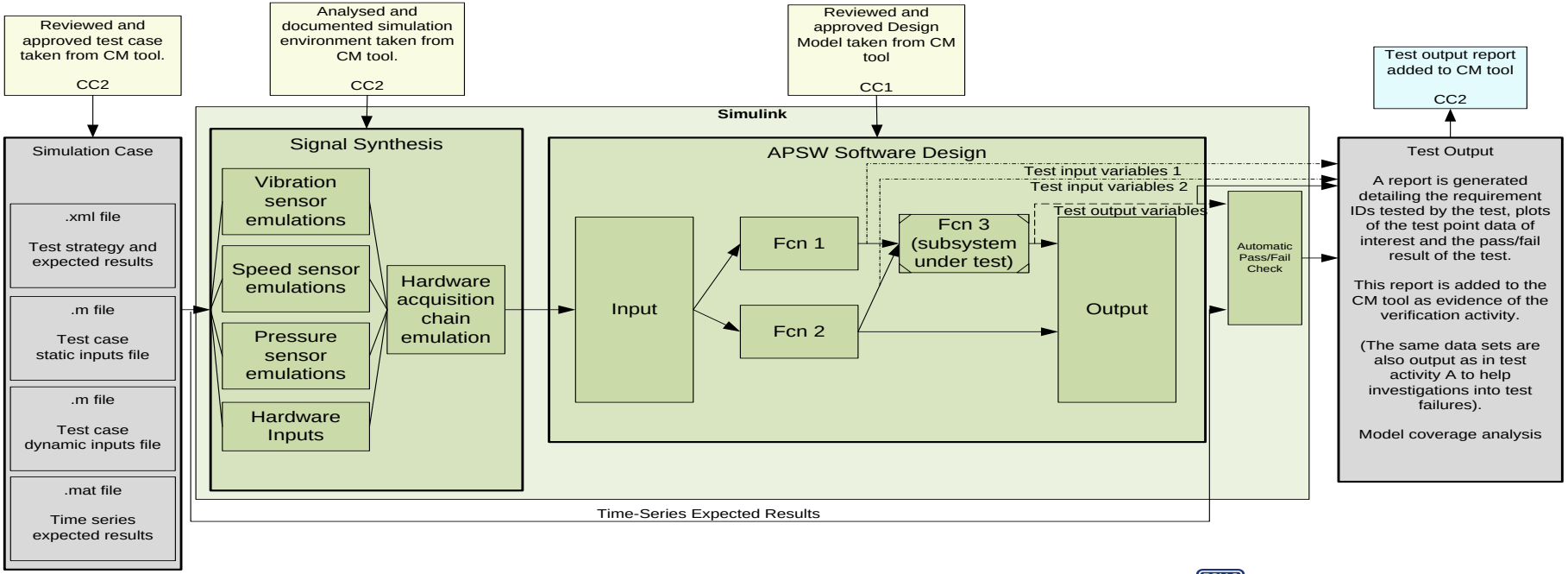
Passed
Optimization > Implement logic signals as Boolean data (vs. double) is selected.

Verify 'Application lifespan' setting

Model verification activities

Simulate the model

Simulation cases written against the high level requirements were used to fully exercise the model using a test tool developed in house.



Model verification activities

Simulation activity outputs - Results Report

[illegible]

REQUIREMENT TESTING [MIL] **SPV_TEST26**

This is a Matlab auto-generated report. It provides a results summary of requirement testing performed on a set of Design Model blocks targeting EMU APSW functionality. Refer to the test process documentation.

TEST SUMMARY

Tester Identification: RRLocal [EMU_DEV_PC1] EMU-DEV-PC-ONE

Date and Time: 10 March 2017 17:03:24

Unit Under Test: Design Model

Module Name			
	Compiled Checksum	Unit Version	Configured Version
Speed_Pressure_Validation			
	1684242041-1349583278-410703400-385426520	1D	MATCHED

TEST RESULT PASS

A point-to-point comparison of output signals is carried out against a manually reviewed set of expected signals. A PASS result indicates that all output signals yield their expected values at all times during the simulation.

REQUIREMENTS UNDER TEST

	Requirement Number	Requirement Description
1	203	The Boolean fault status associated with the N1C-1 speed signal shall be confirmed and latched as being True until the Application Layer is re-initialised, if the N1C-1 fault status is continuously set any other state than OK for more than 3 seconds.
2	541	The Boolean fault status associated with the N1C-1 speed signal shall be initialised to False.

Linked Requirements: 203 541

Expected Results: The hardware fail flag is set to fail for the first 2.9952s which is 1 software cycle under the three second timer therefore the fault should not be confirmed.

The Confirmation output is initialised to 0.

At 3.072s a combination of continuous faults are introduced; these are out of range low then high then hardware acquisition fail. The combined failures persist for longer than 3 seconds so the fault is confirmed at 6.0928s.

All faults are then cleared and the confirmed fault stays latched.

Floating-Point Tolerance: Pass/Fail tolerance value set to 0.

Signal Name	Description	Data Type	Dims
e_N1C_1_fault	Enumeration indicating operational state of the N1C_1 sensor	Enum: e_fault	1
b_Confirmed_N1C_1_fault	Confirmed fault with N1C_1 sensor, True: N1C-1 Fault (latched)	boolean	1

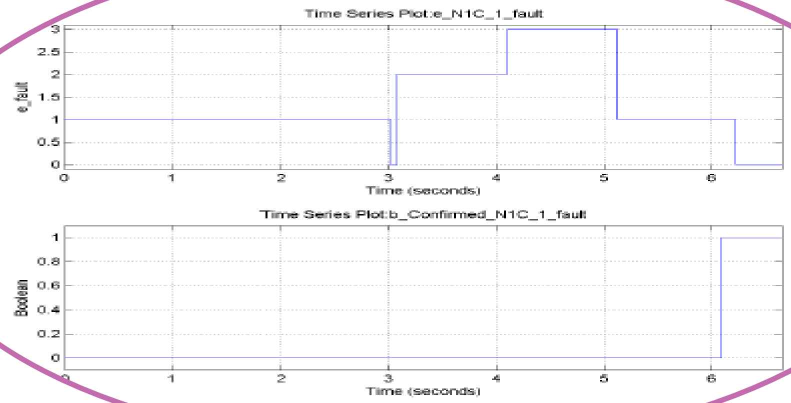


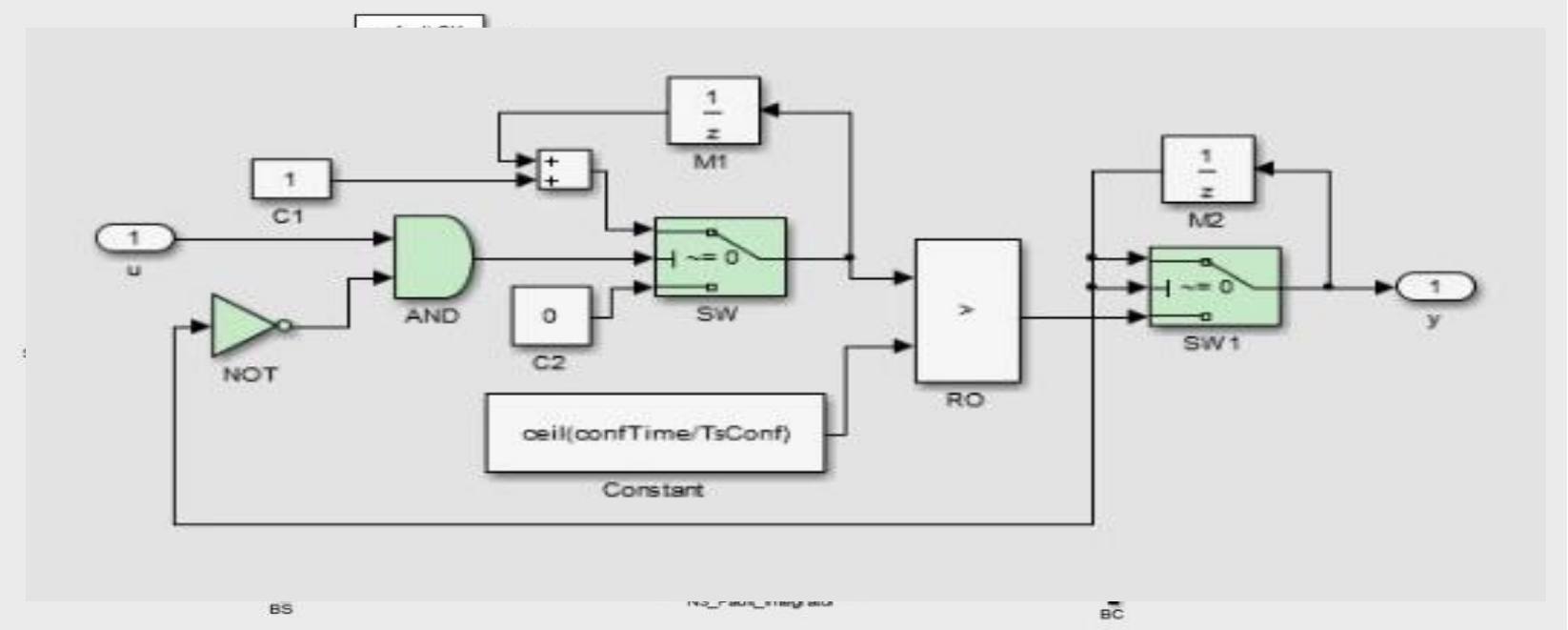
Figure 2: Matlab Figure: SPV_Test26_OutputPlot1.fig

Model-in-the-Loop: Simulink Test Case Auto-Generated Report

Model verification activities

Simulation activity outputs – Model Coverage

Decision and Condition



Model verification activities

Simulation activity outputs – Model Coverage Report

27. SubSystem block "N1C_1_Fault

Parent: Speed_Pressure

Metric	Coverag
Cyclomatic Complexity	0
Condition (C1)	NA
Decision (D1)	NA

Logic block "AND"

Parent: Speed_Pressure

Metric	Coverag
Cyclomatic Complexity	0
Condition (C1)	100% (4

Conditions analyzed:

Description:
input port 1
input port 2

Logic block "NOT"

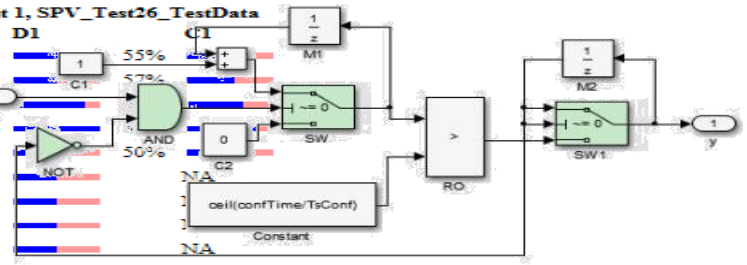
Parent: Speed_Pressure

Metric	Coverag
Cyclomatic Complexity	0
Condition (C1)	100% (2

Conditions analyzed:

Description:
input port 1

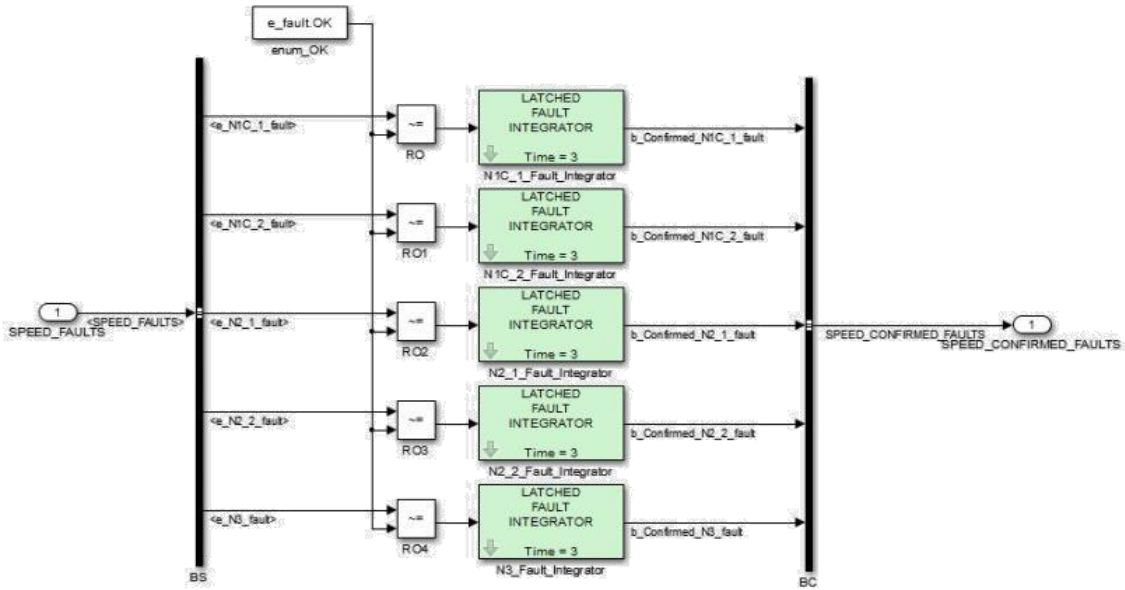
Model Hierarchy/Complexity:			Test 1, SPV_Test26_TestData D1		C1	
1. Speed_Pressure_Validation	62	56%		55%		
2. . . Pressure_Fault_Detection	19	50%		57%		
3. . . . . Fault_Confirmation	3	83				
4. . . . . N1C_1_Fault_Integrator	2	100%				
5. . . . . P160_Calibration_Fail	14	50%				
6. . . . . P160_pres_0V5Cal	2	50%				NA
7. . . . . P160_pres_0VCal	2	50%				
8. . . . . P160_temp_0VCal	2	50%				
9. . . . . P160_temp_2V5Cal	2	50%				NA
10. . . . . Pressure_Excitation_Wrap	2	50%				NA
11. . . . . Pressure_Excitation_Wrap_0V_Cal	2	50%				NA
12. . . . . Pressure_Excitation_Wrap_5V_Cal	2	50%				NA
13. . . . . P160_Range_Check	2	0%				NA
14. . . . . P160_Fault	2	0%				NA
15. . . Speed_Fault_Detection	42	58%		55%		
16. . . . . HW_Speed_Faults	5	60%		54%		
17. . . . . OPR_HW_Fail	8	50%		55%		
18. . . . . Resetable_Fault_Integrator	4	50%		50%		
19. . . . . Count_Clear	2	50%		50%		
20. . . . . Count_Set	2	50%		50%		
21. . . . . Reset_Priority_Latch	0		NA	50%		
22. . . . . Resetable_Fault_Integrator1	4	50%		50%		
23. . . . . Count_Clear	2	50%		50%		
24. . . . . Count_Set	2	50%		50%		
25. . . . . Reset_Priority_Latch	0		NA	50%		
26. . . . . Speed_Fault_Confirmation	10	60%		60%		
27. . . . . N1C_1_Fault_Integrator	2	100%		100%		
28. . . . . N1C_2_Fault_Integrator	2	50%		50%		
29. . . . . N2_1_Fault_Integrator	2	50%		50%		
30. . . . . N2_2_Fault_Integrator	2	50%		50%		
31. . . . . N3_Fault_Integrator	2	50%		50%		
32. . . . . Speed_Range_Check_Enable	0		NA	60%		
33. . . . . Speed_Range_Checks	15	57%				NA



Model verification activities

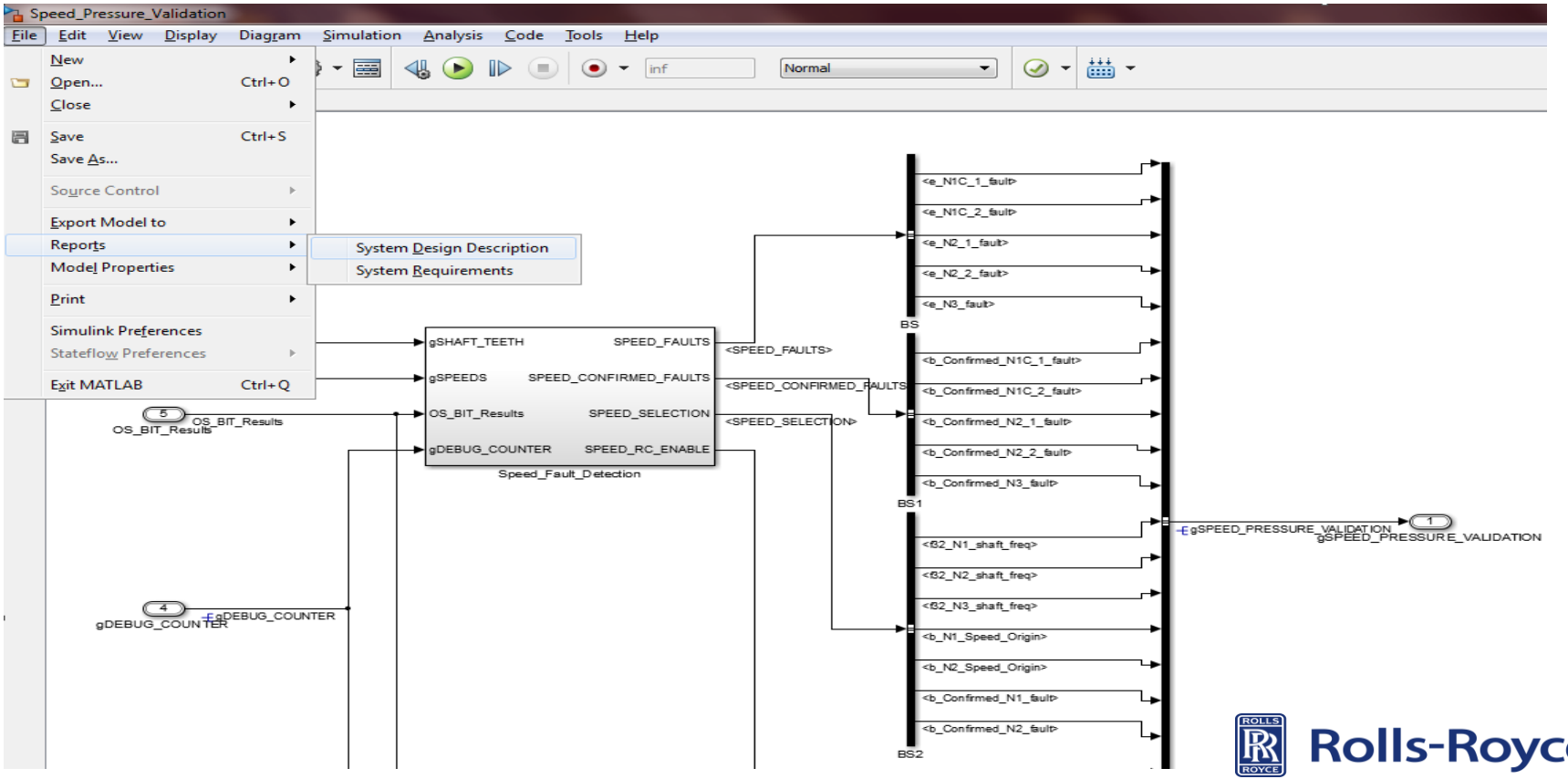
Simulation activity outputs – Model Coverage Report - Cumulative

19.	Count_Clear	2	100%	100%		
20.	Count_Set	2	100%	100%		
21.	Reset_Priority_Latch	0		NA	100%	
22.	Resetable_Fault_Integrator1	4	100%	100%		
23.	Count_Clear	2	100%	100%		
24.	Count_Set	2	100%	100%		
25.	Reset_Priority_Latch	0		NA	100%	
26.	Speed_Fault_Confirmation	10	100%	100%		
27.	N1C_1_Fault_Integrator	2	100%	100%		
28.	N1C_2_Fault_Integrator	2	100%	100%		
29.	N2_1_Fault_Integrator	2	100%	100%		
30.	N2_2_Fault_Integrator	2	100%	100%		
31.	N3_Fault_Integrator	2	100%	100%		
32.	Speed_Range_Check_Enable	0		NA	100%	
33.	Speed_Range_Checks	15	100%		NA	
34.	N1C_1_Fault	2	100%		NA	
35.	N1C_2_Fault	2	100%		NA	
36.	N2_1_Fault	2	100%		NA	
37.	N2_2_Fault	2	100%		NA	
38.	N3_Fault	2	100%		NA	
39.	Speed_Selection	4	100%	100%		



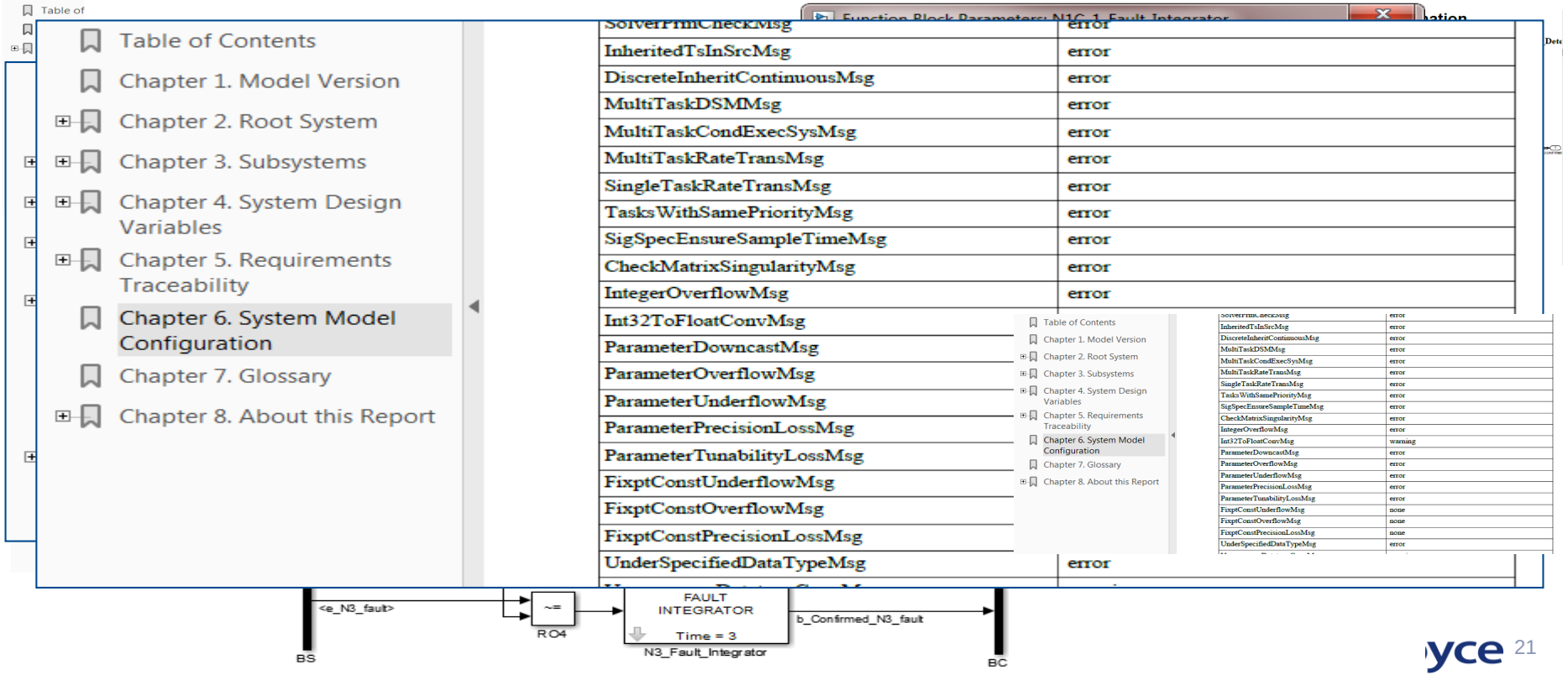
Model verification activities

The model is automatically documented using the report generation tool



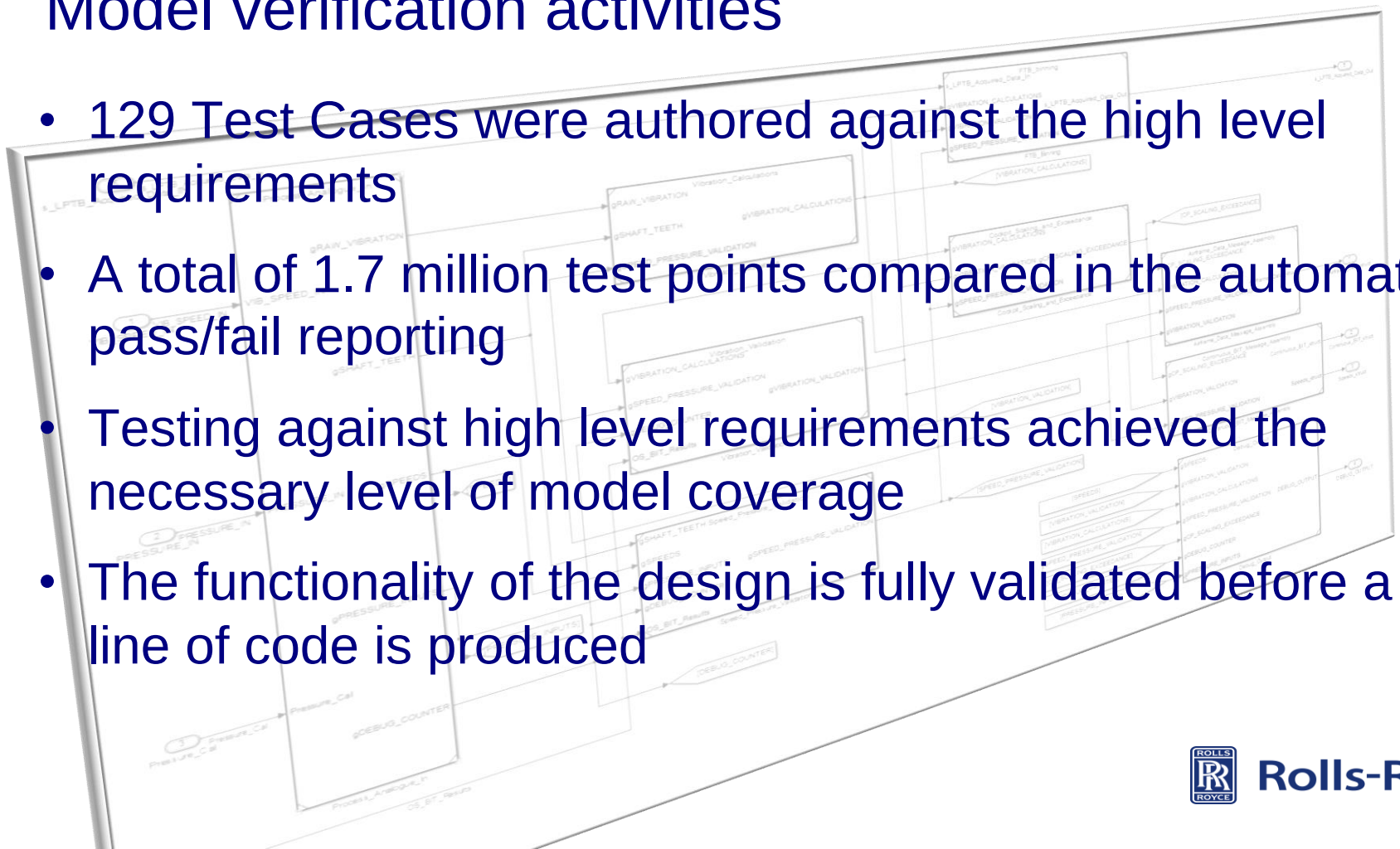
Model verification activities

The Design Description Report captures the complete In-Memory Representation of the Model

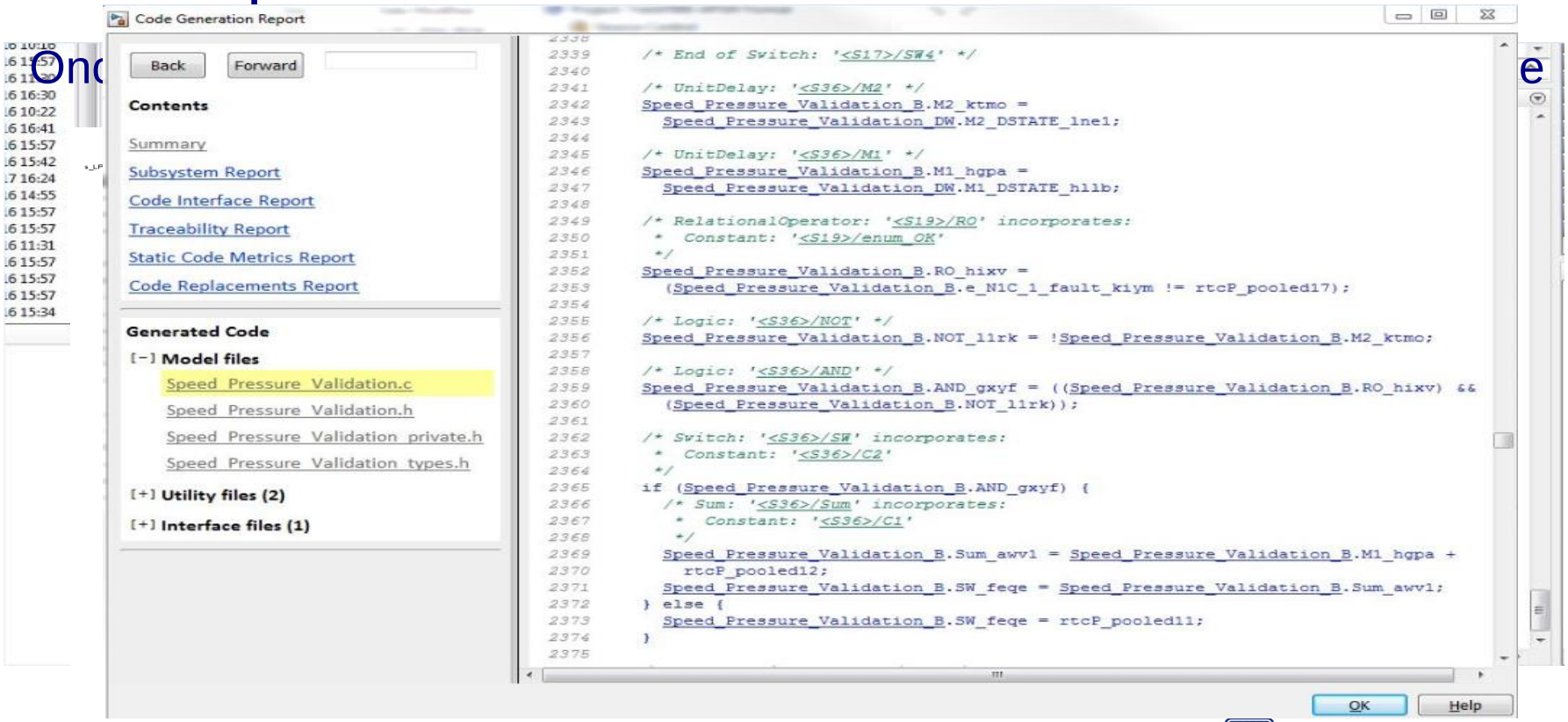


Model verification activities

- 129 Test Cases were authored against the high level requirements
- A total of 1.7 million test points compared in the automatic pass/fail reporting
- Testing against high level requirements achieved the necessary level of model coverage
- The functionality of the design is fully validated before a single line of code is produced



Code production and verification



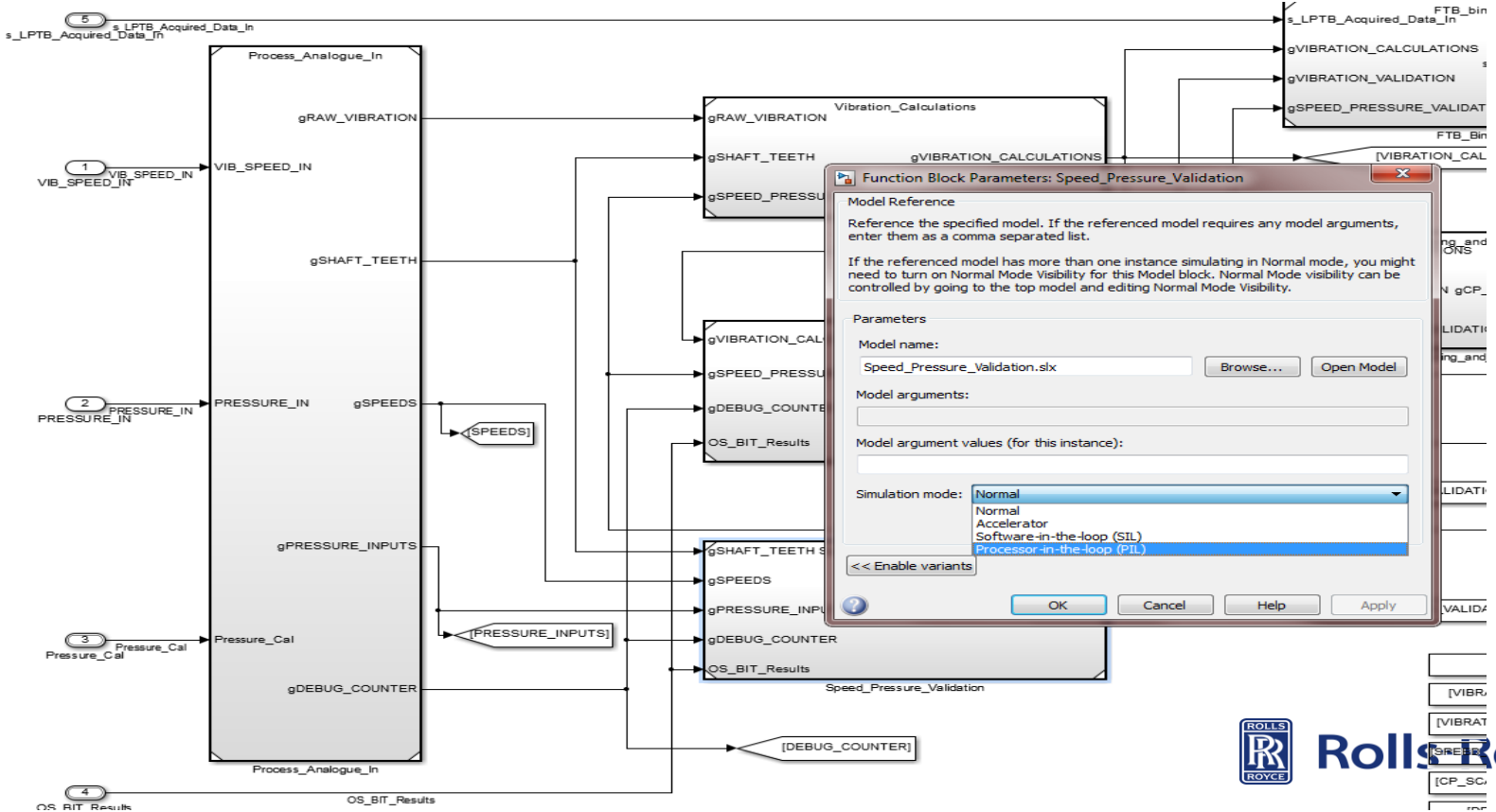
Code production and verification

- The compiled code must be tested to show compliance with the high level requirements.
  - Tests written against the high level requirements already exist from the model simulation.
- The compiled code must be tested to show compliance against the Design Model.
  - This can be achieved by equivalence testing the object code behaviour against the Design Model behaviour.
- The compiled code must be shown to be compatible with the target architecture.
  - This can be achieved by running all paths of the compiled code on representative target architecture

Is there a way that all this can be achieved with the test cases which already exist?

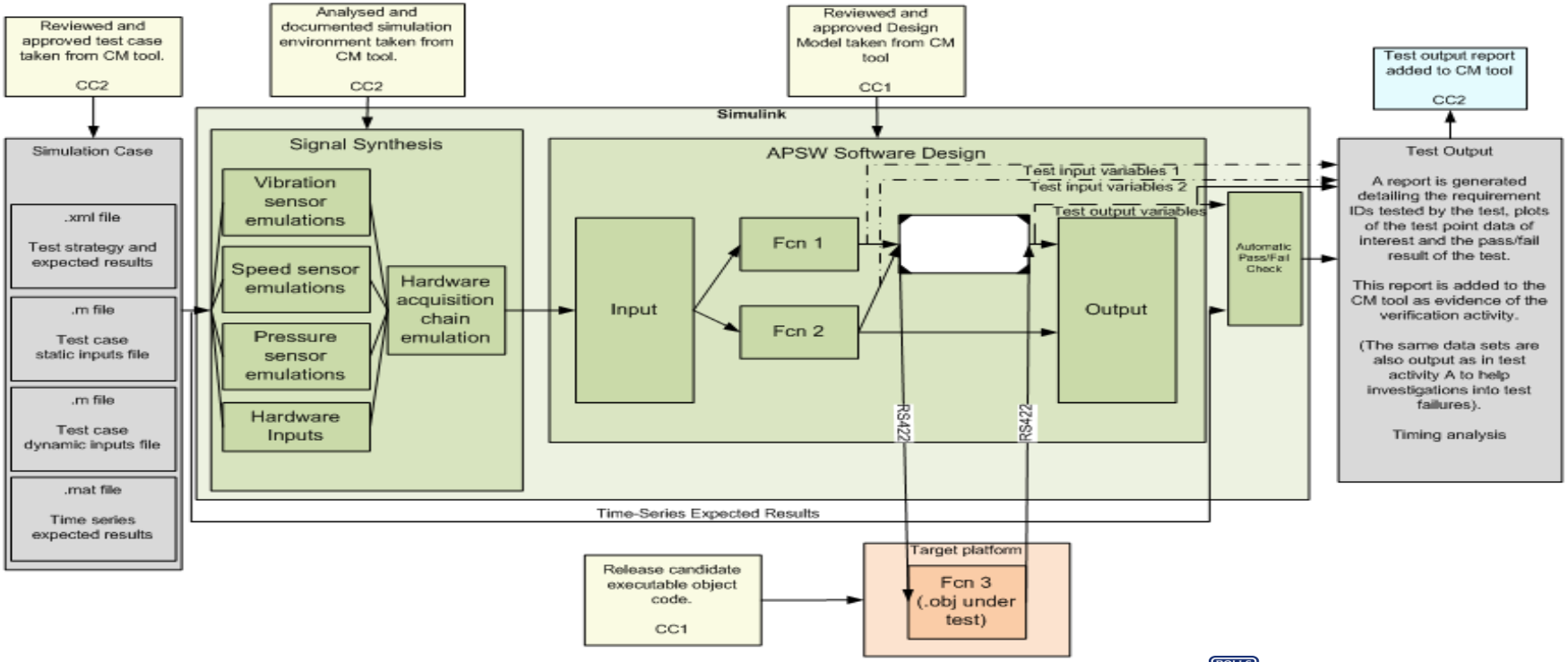
Code production and verification

Processor-in-the-Loop testing



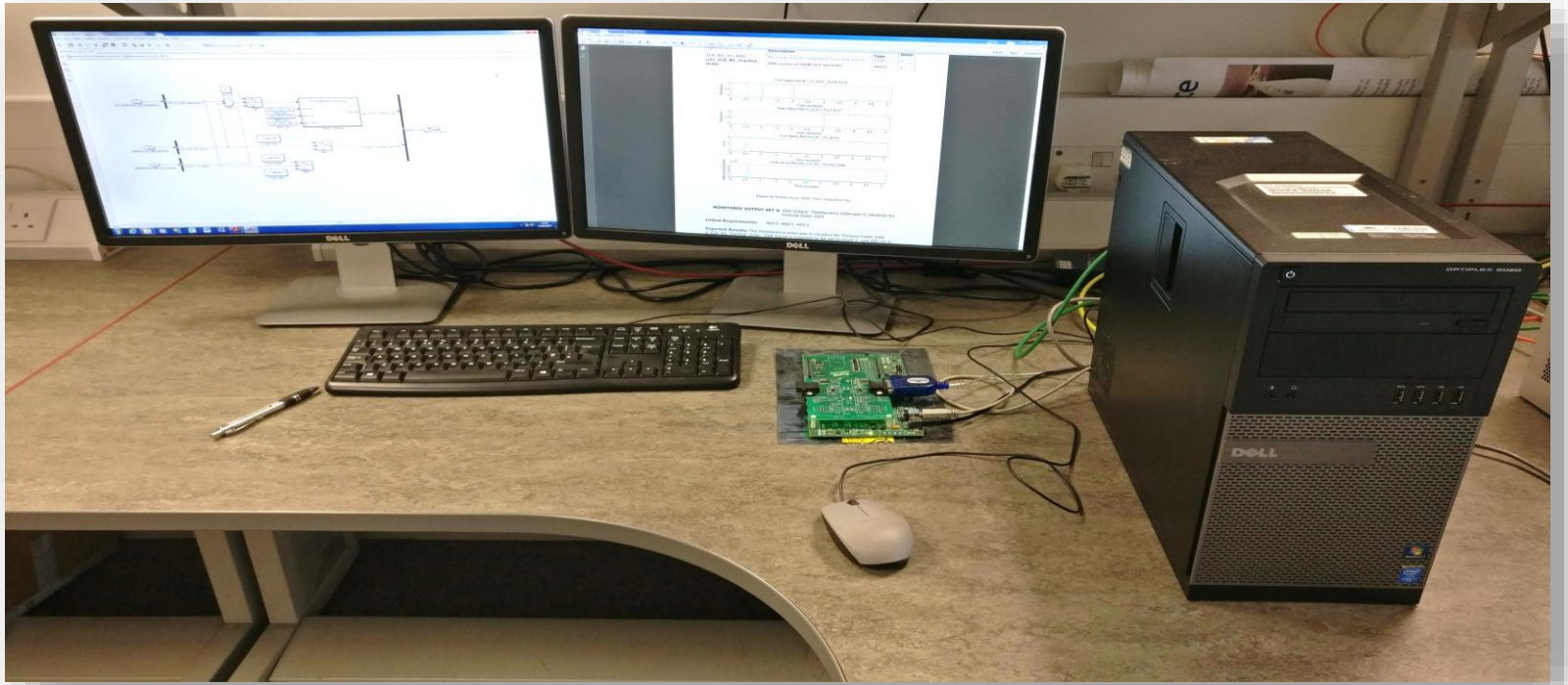
Code production and verification

Processor-in-the-Loop testing



Code production and verification

Processor-in-the-Loop testing



Code production and verification

Processor-in-the-Loop testing – Results Report

The Version number of the object code under test is reported

The PIL infrastructure permits execution timing on a code fragment level

REQUIREMENT TESTING [PIL] **SPV_TEST26**

This is a Matlab auto-generated report. It provides a results summary of requirement testing performed on a set of Software Object Code blocks targeting EMU APSW functionality. Refer to the test process documentation.

TEST SUMMARY

Tester Identification: RRLocal [EMU_DEV_PC1] EMU-DEV-PC-ONE

Date and Time: 10 March 2017 17:41:16

Unit Under Test: Software Object Code

Module Name	Unit Version
Speed_Pressure_Validation	1.0

TEST RESULT PASS

A point-to-point comparison of output signals is carried out against a manually reviewed set of expected signals. A PASS result indicates that all output signals yield their expected values at all times during the simulation.

CODE EXECUTION PROFILE

	Profiled Code Section		
	Maximum Execution Time [μs]	Average Execution Time [μs]	Calls
1	Speed_Pressure_Valid_initialize		
	2.3733	2.3733	1
2	Speed_Pressure_Validation [0.0256 0]		
	190.24	162.6198	261

REQUIREMENTS UNDER TEST

	Requirement Number	Requirement Description
1	203	The Boolean fault status associated with the N1C-1 speed signal shall be confirmed and latched as being True until the Application Layer is re-initialised, if the N1C-1 fault status is continuously set any other state than OK for more than 3 seconds.
2	541	The Boolean fault status associated with the N1C-1 speed signal shall be initialised to False.

Linked Requirements: 203 541

Expected Results: The hardware fail flag is set to fail for the first 2.9952s which is 1 software cycle under the three second timer therefore the fault should not be confirmed.

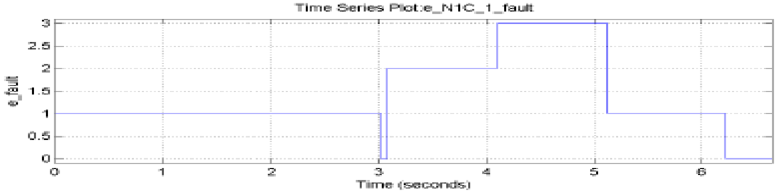
The Confirmation output is initialised to 0.

At 3.072s a combination of continuous faults are introduced; these are out of range low then high then hardware acquisition fail. The combined failures persist for longer than 3 seconds so the fault is confirmed at 6.0928s.

All faults are then cleared and the confirmed fault stays latched.

Floating-Point Tolerance: Pass/Fail tolerance value set to 0.

Signal Name	Description	Data Type	Dims
e_N1C_1_fault	Enumeration indicating operational state of the N1C_1 sensor	Enum: e_fault	1
b_Confirmed_N1C_1_fault	Confirmed fault with N1C_1 sensor; True: N1C-1 Fault (latched)	boolean	1



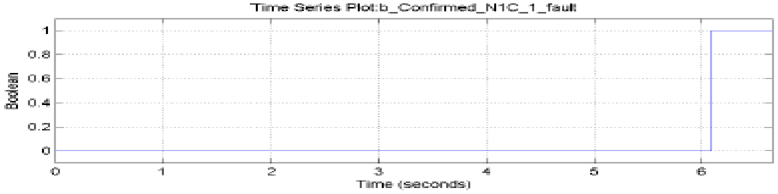


Figure 2: Matlab Figure: SPV_Test26_OutputPlot1.fig

Model-in-the-Loop: Simulink Test Case Auto-Generated Report

Rolls-Royce

Tool Qualification

When is tool qualification required?

"Qualification of a tool is needed when processes of this document are **eliminated**, **reduced**, or **automated** by the use of a software tool without its output being verified [DO-178C 12.2.1]"

Which parts of this process required Tool Qualification?

- The Model Advisor checks automated part of the 'compliance with standards' process.
- The Report Generator is qualified to show equivalence with the in-memory representation of the model.
- The Model Coverage tool is qualified as this automates the metrics associated with the 'model coverage' process.
- The pass/fail reporting of test results automated parts of the 'compliance' processes
- The 'PIL' infrastructure automates code coverage.

Tool Qualification

DO Qualification Kit (for DO-178)

- DO Qualification Kit provides documentation, test cases, and procedures that let you qualify Simulink software verification tools
- The kit contains tool qualification plans, tool operational requirements, and other materials required for qualifying software verification tools

Future Enhancements

Further tool adoption

Automated code review

- The source code required a manual review to show compliance against the Design Model.
- This activity could be automated by using the ‘Simulink Code Inspector’

Simulink Test

- The in-house test frame work has a maintenance overhead, especially around version migration.
- The introduction of ‘Simulink Test’ in means we now have an alternative solution which is natively supported by the MathWorks.


Thank You
© AIRBUS 2017 - photo by A. DOUMENJOU / master films
AIRBUS