Verification and Validation of Models for Embedded Software Development

Manohar Reddy
Durvesh Kulkarni
Prashant Hegde
Why Early Verification of Models

It is easier and less expensive to fix errors early in the process when they happen

We will talk about ....

How Model Based Design (MBD) facilitates early testing to incrementally increase confidence in your design
Conceptual Test
Challenges with Conceptual/Traditional Testing

- Are all requirements implemented?
- Does Implementation comply to modeling standards?
- Absence of design errors (dead logic, divide by zero, overflow)?
- Does the implementation pass all functional requirements?
- Do the tests completely exercise the implementation?
- Does the existing test framework allows to reproduce field issues?
Gaining Confidence in the Design

- Requirement Traceability Gap Analysis
- Modeling Standards Compliance
- Design Error Detection
- Functional Testing And Debugging
- Structural Coverage Analysis
- Reproduce Field Issues

Confidence vs. Effort / Time
Traceability: Requirements ↔ Model
Requirements Management Interface

- Creating links between textual documents and model objects
Traceability Report

Requirements Traceability Report for CruiseControl

Table of Contents

1. Model Information for "CruiseControl"
2. Traceability Summary for "CruiseControl"
3. Chart - Compute target speed

List of Tables

1.1. CruiseControl
2.1. Artifacts linked in model
3.1. Objects in "Compute target speed" that have requirements

Chapter 1. Model Information for "CruiseControl"

Table 1.1. CruiseControl

<table>
<thead>
<tr>
<th>ModelVersion</th>
<th>1.307</th>
<th>ConfigurationManager</th>
<th>None</th>
</tr>
</thead>
<tbody>
<tr>
<td>LastModifiedDate</td>
<td>Wed Feb 11 22:16:45 2015</td>
<td>LastModifiedBy</td>
<td>mreddy</td>
</tr>
</tbody>
</table>
Enforcing Modeling Standards:
Model Advisor Checks

- Static analysis of models against a set of checks
  - Checks for simulation configuration
  - Checks for code generation configuration
  - Requirements consistency
  - Modeling Standards

- Modeling Standards Checks (SLVNV)
  - MAAB Style Guidelines
  - DO-178
  - IEC 61508 and ISO 26262 Checks
  - IEC 62304 Checks

- API for process customization

- Create custom checks and configurations
Gaining Confidence in the Design

- Confident Modeling
- Standards Compliance
- Requirement Traceability
- Gap Analysis
- Design Error Detection
- Functional Testing and Debugging
- Structural Coverage Analysis
- Reproduce Field Issues

Confidence over Effort / Time
Identifying Design Errors Early

The Cost of Failure...

News reports:

Recall

Due to ECU software bug
What do all these systems have in common?

- Complex software developed to rigorous standards
- Extensively reviewed, analyzed and tested
- Yet still succumbed to costly failure
Identifying Design Errors Early

Automatic identification of hard-to-find design inconsistencies in the model

- Dead logic
- Division by zero
- Overflow
- Out of bound Array Access
Automating Design Error Detection

Design error checking capability integrated in Model Advisor

- Checks that can be turned on/off in Model Advisor
- Design error results reported
Gaining Confidence in the Design Effort / Time

- Confidence
- Design Error Detection
- Modeling Standards Compliance
- Requirement Traceability Gap Analysis
- Functional Testing and Debugging
- Structural Coverage Analysis
- Reproduce Field Issues
Simulation Testing Workflow

- **Simulate and Validate Controller Response**
- **Review Functional Behavior**
- **Detect Unreachable Design Logic**
- **Structural Coverage Report**

**Requirements**

- **Functional Requirements**
  - 1.1.1. Enable (on) during start-up and not engaged (in action)
  - 1.1.2. Not engaged (inactive) with enabling (on)
  - 1.1.3. Disable (off) when engaged (active)

**Tests**

**Design**

**Report Generated by Test Manager**

- Title: LandingGearControl-Regression Tests
- Author: Jessica Johnson
- Date: 20-Feb-2015 18:24:22

**Test Environment**

- Platform: PC
- MATLAB: R2015a
Simulation Testing using Simulink Test

Capabilities to test simulation outputs throughout the project lifecycle

Do legacy subsystem models work in new application?
The model worked last week… does it still?
Another engineer modified the algorithm… still working ok?
Does the generated code performance match the model?
Does subsystem/system meet the design requirements?
Functional Testing
# Simulink Test

*authoring, managing, and executing simulation-based tests*

<table>
<thead>
<tr>
<th>1. Test Harnesses</th>
<th>2. Test Sequence Block</th>
<th>3. Test Manager</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Synchronized, simulatable test environment</td>
<td>• Inputs and assessments based on logical, temporal conditions</td>
<td>• Author, execute and manage test cases • Review, export, report</td>
</tr>
</tbody>
</table>

**Test Harness**

- Synchronized, simulatable test environment

**Main Model**

- Component under test

**Test Sequence**

- inputs and assessments based on logical, temporal conditions

**Test Manager**

- Author, execute and manage test cases
- Review, export, report
1. Simulation Test
   Pass/Fail assessments with blocks (Assertion)

2. Baseline Test
   Regression test for float-to-fixed point conversion of low-pass filter

3. Equivalence Test
   Comparison between Normal and Software-In-the-Loop tests
I need to make a change to my large-scale model.

Can you show me just the relevant pieces of the model?
Model Slicer: Test and Debug Complex Models
Model Coverage Report

- Coverage metrics identifies untested portions of the model
Code Coverage for SIL Simulation Mode

Collect code coverage for model running in Software-In-the-Loop simulation mode

- Record Decision, Condition, MC/DC and Statement coverage on the generated code
- Same command line API as for Model Coverage. Similar HTML Report.
- Map code coverage results to model elements
Gaining Confidence in the Design

- Requirement Traceability Gap Analysis
- Modeling Standards Compliance
- Design Error Detection
- Functional Testing And Debugging
- Structural Coverage Analysis
- Reproduce Field Issues
Addressing Coverage Challenges

Types of Coverage objectives

- **Condition**
- **Decision**

**MCDC (Modified Condition Decision Coverage)**

**MC/DC Coverage:** Each condition independently changes the decision outcome.

```matlab
if (X & Y) Z = 1;
else Z = -1;
end
```

<table>
<thead>
<tr>
<th>X</th>
<th>Y</th>
<th>X&amp;Y</th>
<th>MC/DC?</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>Yes</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>Yes</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>Yes</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>No</td>
</tr>
</tbody>
</table>
Manual test vectors creation for MCDC objectives
Does this process takes time?
Is this exhaustive?
Automatic Test Generation

- Automatic Test Vector Generation Modified condition and Decision Coverage (MCDC)
- Address the missing coverage by reusing the existing coverage or test cases
- Generate Test cases with Relational Boundary
- Include S-functions in the test generation process
- Reports in HTML and PDF format
- Data Inspector view
- Exporting test cases to Simulink Test
Improving Test Suite for Structural Coverage
Already have requirements based test vectors, 100% coverage, and all passed

Still got a field issue !!!

ANY ONE FACED SUCH SITUATION ?

ANALYSIS OF SUCH FIELD ISSUE TAKES TIME : YES/NO ?
Field Issue Reported

From: [Redacted]
Sent: Monday, March 2, 2015 9:29 PM
To: Prashant Hegde
Subject: Cruise Control Target Speed Issue

I ran into an unexpected issue with the new Cruise Control feature during vehicle testing.

While going downhill, random target speed increases with “reduce speed” button
a) I set the target speed while going downhill on the track going around 40 mph
b) While pulsing the “reduce speed” button, target speed reduced to 20 mph limit
c) Next time I hit “reduce speed”, it increased the target speed from 20 mph to 33 mph
d) Another time it increased the target speed from 20 mph to 29 mph

Hopefully we can fix this issue quickly. These issues that don’t repeat themselves can be hard to find. Let me know if I can help.

Factors Influencing Field Issues

- Environmental conditions
- User Inputs
- Calibration values
- Sequence of events
Field Issue Behavior
Target speed increases with “reduce speed” button

Use Model-Based Design to Reproduce Field Issue
- Construct a model of field issue for Simulink Design Verifier
- Constrain inputs to represent field issue
- Create model of field issue behavior
- Ask tool to prove whether errant condition can occur

a) I set the target speed to the vehicle speed (40 mph) while going downhill on the track
b) I was pulsing the “reduce speed” button until it decreased the target speed to the 20 mph limit
c) The next time I hit the “reduce speed” button it increased the target speed from 20 mph to 33 mph
Formal Verification: Requirements and Issues

Property Proving
- Validate implementation against the requirements
- Generate Test Cases

By using formal verification methods… Simulink Design Verifier proves correct behaviors over a larger amount of the design space
Reproducing Field Issue in Simulation
Gaining Confidence in the Design

Confidence

Effort / Time

- Requirement Traceability Gap Analysis
- Modeling Standards Compliance
- Design Error Detection
- Functional Testing And Debugging
- Structural Coverage Analysis
- Reproduce Field Issues
Toyota and DENSO adopt Simulink Design Verifier, Simulink Test

“Using R2015a … Toyota and DENSO engineers realize efficiency and productivity benefits of model based incremental development in current and future production vehicle development”

“Incremental development support with the Model Slicer feature of Simulink Design Verifier, and simulation testing with Simulink Test”

Importance of Early Verification
Based on MBD Return-on-Investment (ROI) Analysis

It is easier and less expensive to fix errors early in the process when they happen

Biggest Return:
From Early Verification

Additional Return:
From Automation
MathWorks Training Offerings

MODEL VERIFICATION

Verification and Validation of Simulink Models

- Identifying the role of verification and validation in Model-Based Design
- Creating test cases for Simulink models
- Analyzing simulation results to verify model behavior
- Automating testing activities and managing results
- Formally verifying model behavior
- Automatically generating artifacts to communicate results

Prerequisites: MATLAB Fundamentals and Simulink for System and Algorithm Modeling. This course is intended for intermediate or advanced Simulink users.

http://www.mathworks.com/services/training/
See You at Technology/Demo Showcase

Model and Code Verification Through Iterative Test and Analysis

- Checking compliance with modeling guidelines
- Detecting hard-to-find design errors
- Simulation-based testing using Simulink Test™
- Model slicer for isolating the problematic behavior
- Automatic test generation for structural coverage
- Proving the absence of run-time errors in embedded software

Technology Focus: Verification and validation