Volvo Cars/Danlaw

Automation and Optimization of System Verification in Volvo’s Model Based Development Process
What led to the adoption of the Mx-Suite For System Verification

- Volvo’s System Verification Framework.
- Volvo’s Test Tool Selection Criteria.
Verification Test Suite of a System

- Tests that prove a system meets its Requirements.
- Tests that qualify a system for delivery to your customer.

Typically:
- Run often (regression testing) and so are a good candidate for automation.
- Want high levels of Requirements Coverage (thus high code/model coverage).
Decide in what phase you test a Requirement and ensure you have Full Coverage

**Integration Level**
- Feature (Unit) --- Subsystem --- System

**Simulation Level**
- MIL --- SIL --- PIL --- HIL

**Requirement/Test Coverage Matrix**

<table>
<thead>
<tr>
<th>Requirement/Test Coverage Matrix</th>
<th>Req 1</th>
<th>Req 2</th>
<th>Req 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Req 1</td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Req 2</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Req 3</td>
<td></td>
<td>✓</td>
<td></td>
</tr>
</tbody>
</table>

- Easy to access signals
- Easier to hit test points
- Tests are less realistic (e.g. no CAN bus)
- Tests can be trivial
- Often no requirements

- Access to more internal signals
- Issues are easier to debug
- Coverage measurement easier

- Harder to diagnose issues
- Much more value from Test Case reuse
- Much higher confidence in your delivery to your customer

- Test equipment more expensive
- Test equipment harder to deploy
- System more realistic (real timing, compiler, …)
- Customer really interested how ECU operates
Code/Model Coverage

Functional Requirements

Code/Model Coverage Analysis
- Requirements Incomplete
- Tests Incomplete
- Unused code

Translation into test cases

Automated Execution of Tests and Code Coverage Measurement
Automating Regression Testing

- Developers check in Code/Models and the corresponding test cases.
- Triggers based on time and check-in actions initiate an automated test cycle.
- The Build Server checks out the full source code set, builds and deploys the executable.
- An automated regression test is initiated.
- Developers notified automatically of a failure.

People don’t do the Testing
Computer Systems Do
Test Tool Selection Criteria

- **Data Driven** (preferred over language based).
- **Abstraction** (preferred over coupling between test cases and test environment).
- **Enterprise Solution** (preferred over several Point Tools).
- **Comprehensive base feature set.**
Data Driven Testing - Preferred

**Code or Data Driven**

- **Code – Python, VB, CAPL, Other Script**
  - **Benefits**
    - Some advantage for building tests with rules or many combinations.
  - **Drawbacks**
    - Maintenance is labor intensive.
    - Need trained programmers in several languages (CAPL, Python, …).
    - Even when you buy a programming test environment you need to create your own support library.

- **CSV, Excel, XML, State Tables, Other**
  - **Benefits**
    - Maintenance is significantly simpler. A system behavior can be captured and used as a future expected behavior.
    - Test information is coupled closely to requirements because code and test data are not intermixed.
    - It is easier to present test data in an intuitively understandable format.
    - Need little skill/training to author/understand tests.
  - **Drawbacks**
    - Typically poor user interface and poor support for testing tasks like reporting and Pass/Fail Analysis - Till now.

Danlaw Confidential
Benefits

- Test cases can be run on different equipment (not locked into one HIL).
- Tests can be in one form, instead of Python, CAPL, CSV, VB, RTRT Script, …
- People creating tests don’t need to learn several tools
- Test cases can be reused across test phases (SIL, MIL, HIL).
- Test tool can be used across all phases.
Benefits:

- Simpler licensing - one product to license
- Improved test and test result communication across lifecycle
- Customizations (like reports) apply across the development V
t
- Generic features apply across the lifecycle
- Requirements Coverage Matrix - SUT Integration/Simulation Level selection not made on availability of tool
- Auxiliary tools and automation efforts can be targeted towards the one product
Desired base feature set

- Support for automation/Continuous Integration.
- Executable specification - Requirements/Use Cases/Test Cases in same format.
- Support for 26262 (Traceability and code/model Coverage).
- Scalable licensing (You never have licenses for unused components, you always have access to all features).
- Easily extended (e.g. to accommodate Elektrobit’s AUTOSAR Wincore).
- Easy to learn and easy to understand the work of others.
- Powerful Pass/Fail judgment features.
- Scalable to high data rates, large I/O count, extended test duration, high precision.
- Co-simulation
- Automotive support: CAN, LIN, Flexray, CCP, XCP, Most, AUTOSAR, Diagnostics,
Joint Volvo/Danlaw white paper

Mx-Suite™ User Application
Volvo Vehicle Electronics System Validation

Some Mx-VDev™ Users

- Volvo
- Jaguar
- Land Rover
- Nissan
- Toyota
- Chrysler
- Daimler
- Continental
- Magna Steyr
- BorgWarner
- Meritor
- Lear Corporation
- Delta
- Delphi
- Nippon Seiki
- HELLA
- A123 Systems
- Magneti Marelli
- Magna
- TRW
- Yazaki
- KPI Cummins
- Tata Consultancy Services
- CMC Limited
Thank You for Your Time

Mike McCormack
Director Product Development
Tel: (248) 736 2518
Email: mikem@danlawinc.com
Web: www.danlawinc.com