Topic Group Testing and Validation

Johannes Baumgartl, Kai Hudalla
the team

Daniel Lueddecke
Johannes Baumgartl
Kai Hudalla
Sara Gallian
Thilo Schmitt
Badii Ennouri
Sören Frey
Motivation

Topic Group: Testing and Validation

Topic Group: Unified In-Vehicle Software Orchestration

COPYRIGHT (C) 2023, ECLIPSE FOUNDATION. THIS WORK IS LICENSED UNDER A CREATIVE COMMONS ATTRIBUTION 4.0 INTERNATIONAL LICENSE (CC BY 4.0)
Helping SDV Projects to become automotive grade

Validate automotive grade Software* by providing

Guidelines
• How to validate FOSS software for automotive use cases*

Code
• Enhance existing code bases with vendor neutral T&V Cases

Projects
• Automated tool chain for vendor neutral automatic T&V-cases

*starting with QM use cases
Framing of the Topic Group so Far

3-Step approach from a neutral T&V-Case Definition to an executed and understandable Test and Result:

1. **Create the test setting** based on the test case.
   - Request virtual or real HW
   - Set up Software
   - Set up Rest of Vehicle and Sensors
   - Set up Network
   - Check QoS

2. **Test** in an open or closed loop on real or emulated hardware.
   - Execute the neutral T&V-Case definition
   - Systematic/Stochastic config variations
   - Internal Monitoring and Diagnostics

3. **Process the results and insights** for the different roles/users.
   - Observe and provide understandable information
   - What should be changed by developers to pass the validation
   - Offer reports on the validation for legal aspects

**Cucumber + gherkin**

**e.g. DLT, openTelemetry**
Goals:
- Evaluate suggested technologies
- Evaluate if our approach gives value to the community

Scenario Outline: Reading the current value works
Given a Data Entry \(<path>\) of type \(<type>\) having value \(<value>\)
When a client gets the current value of \(<path>\)
Then the current value for \(<path>\) is \(<value>\) having type \(<type>\)

Examples:

<table>
<thead>
<tr>
<th>path</th>
<th>type</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vehicle.Cabin.Sunroof.Position</td>
<td>int8</td>
<td>-128</td>
</tr>
<tr>
<td>Vehicle.Width</td>
<td>uint16</td>
<td>65535</td>
</tr>
<tr>
<td>Vehicle.Powertrain.Range</td>
<td>uint32</td>
<td>4294967295</td>
</tr>
<tr>
<td>Vehicle.TraveledDistanceHighRes</td>
<td>uint64</td>
<td>2342546246563924</td>
</tr>
<tr>
<td>Vehicle.CurrentLocation.Longitude</td>
<td>double</td>
<td>145.023544</td>
</tr>
<tr>
<td>Vehicle.Speed</td>
<td>float</td>
<td>45.5</td>
</tr>
<tr>
<td>Vehicle.IsMoving</td>
<td>bool</td>
<td>true</td>
</tr>
</tbody>
</table>

Scenario Outline: Setting current value of wrong type fails
When a client sets the current value of \(<path>\) of type \(<type>\) to \(<value>\)
Then setting the value for \(<path>\) fails with error code 400

Examples:

<table>
<thead>
<tr>
<th>path</th>
<th>type</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vehicle.Cabin.Sunroof.Position</td>
<td>bool</td>
<td>true</td>
</tr>
<tr>
<td>Vehicle.Powertrain.ElectricMotor.Speed</td>
<td>uint16</td>
<td>13648</td>
</tr>
<tr>
<td>Vehicle.Width</td>
<td>int8</td>
<td>-35</td>
</tr>
<tr>
<td>Vehicle.Powertrain.Range</td>
<td>int16</td>
<td>-7295</td>
</tr>
<tr>
<td>Vehicle.TraveledDistanceHighRes</td>
<td>float</td>
<td>-6.3924</td>
</tr>
<tr>
<td>Vehicle.CurrentLocation.Longitude</td>
<td>uint16</td>
<td>14502</td>
</tr>
<tr>
<td>Vehicle.Speed</td>
<td>bool</td>
<td>false</td>
</tr>
<tr>
<td>Vehicle.IsMoving</td>
<td>uint8</td>
<td>0</td>
</tr>
</tbody>
</table>
Topics you would like to work on – join us and shape the road ahead

Reproducible builds

Qualifying FOSS for safety relevant use cases
THANK YOU!