The application of MBD in automotive functional safety projects

Cheng Hui
KOSTAL ASIA
2015.06.18
1. MBD status in KOSTAL ASIA
   科世达亚洲MBD现状

2. KOSTAL ASIA software develop process (MBD)
   科世达亚洲软件开发流程

3. The application of MBD in functional safety projects
   MBD在功能安全项目中的应用
1. MBD status in KOSTAL ASIA
科世达亚洲MBD现状
KOSTAL Sites and Workforce

- Detroit Rochester Hills (313)
- Querétaro Acajutla (2,175)
- Sao Paulo Cravinhos Manaus (1,299)
- Shanghai Changchun (2,597)
- Kawasaki Hiroshima (30)
- Ranipet (227)

Source: acc. to IRT, average 2014

Sales
R&D
Production
Business Area: Automotive Electrical systems - Product Mix

**Mechatronics**

- Steering Column Modules
- DAC / RLS
- Roof Module
- E-Shifter

**Electronics**

- Body
- Door
- Access Electronics
- On-Board Charger (OBC)

**Operating Elements & Switches**

- Seat
- Seat sw. / Module
- Door sw. / Module
- Door
- Faceplate
- Steering Wheel sw.
Projects with MBD

**BFM**
- Light and wiper controller
- Sop: 2013.12
- SOP

**ESCL**
- ASIL-D required
- Sop: 2016
- Ongoing

**PEPS**
- First Sop: 2014.05
- Next Sop: 2015.08
- SOP

**SCM**
- Steering switch with LIN
- Sop: 2016
- Ongoing
2. KOSTAL ASIA software develop process (MBD)
科世达亚洲软件开发流程
V-Model

- System Architecture Design
- Software Requirement Analysis
- System Test
- System Integration Test

Software Design
Software Integration
Software Construction

MBD in component level especially for application layer

Domain: Elect. Hardware
Domain: Mechanical
Domain: Software
Flowchart of Model Based Design

**MBD Process**

- **Step 1**: Requirement Analysis
- **Step 2**: Architecture Design
- **Step 3**: Model Design
- **Step 4**: Unit Test
- **Step 5**: Integration Test
- **Step 6**: Code Generation
- **Step 7**: Back to Back Test

**Validation & Verification**
- Model Check Report
- Unit Test Model
- Test cases
- Unit Test Report
- Coverage Report

**Product**
- Customer Requirement
- System Requirement

**Design**
- Model Structure
  - Interface definition
  - arxml/xml

**Coding**
- Source Code
- Code Generation Report

**Embedded Coder**
- Matlab Coder
- Simulink Coder
- Embedded Coder

**Product**
- Software Requirement
- Requirement Analysis
- EA/ DaVinCi/Kostal RTE
- Simulink/Stateflow

**Product**
- Data Dictionary
- Model design docs
- Model file
- Execution order list

**Product**
- Integ. Test Model
- Integ. cases
- Integ. Test Report

**Product**
- Back to Back test Report

**Product**
- DOORS

**Product**
- Embedded Tester

**Product**
- Embedded Tester
The application of MBD in functional safety projects

MD在功能安全项目中的应用
Structure of ISO26262

<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>2-5 Overall safety management</td>
<td>2-6 Safety management during the concept phase and the product development</td>
</tr>
<tr>
<td>2-7 Safety management after the item’s release for production</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>3. Concept phase</th>
<th>4. Product development at the system level</th>
</tr>
</thead>
<tbody>
<tr>
<td>3-5 Item definition</td>
<td>4-5 Initiation of product development at the system level</td>
</tr>
<tr>
<td>3-6 Initiation of the safety lifecycle</td>
<td>4-6 Specification of the technical safety requirements</td>
</tr>
<tr>
<td>3-7 Hazard analysis and risk assessment</td>
<td>4-7 System design</td>
</tr>
<tr>
<td>3-8 Functional safety concept</td>
<td>4-8 Item integration and testing</td>
</tr>
<tr>
<td>4-9 Safety validation</td>
<td>4-10 Functional safety assessment</td>
</tr>
<tr>
<td>4-11 Release for production</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>5. Product development at the hardware level</th>
<th>6. Product development at the software level</th>
</tr>
</thead>
<tbody>
<tr>
<td>5-5 Initiation of product development at the hardware level</td>
<td>6-5 Initiation of product development at the software level</td>
</tr>
<tr>
<td>5-6 Specification of hardware safety requirements</td>
<td>6-7 Software architectural design</td>
</tr>
<tr>
<td>5-7 Hardware design</td>
<td>6-8 Software unit design and implementation</td>
</tr>
<tr>
<td>5-8 Evaluation of the hardware architectural metric</td>
<td>6-9 Software unit testing</td>
</tr>
<tr>
<td>5-9 Evaluation of the safety goal violations due to random hardware failures</td>
<td>6-10 Software integration and testing</td>
</tr>
<tr>
<td>5-10 Hardware integration and testing</td>
<td>6-11 Verification of software safety requirements</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>8. Supporting processes</th>
<th>9. ASIL-oriented and safety-oriented analyses</th>
</tr>
</thead>
<tbody>
<tr>
<td>8-5 Interfaces within distributed developments</td>
<td>9-5 Requirements decomposition with respect to ASIL tailoring</td>
</tr>
<tr>
<td>8-6 Specification and management of safety requirements</td>
<td>9-6 Criteria for coexistence of elements</td>
</tr>
<tr>
<td>8-7 Configuration management</td>
<td>9-7 Analysis of dependent failures</td>
</tr>
<tr>
<td>8-8 Change management</td>
<td>9-8 Safety analyses</td>
</tr>
<tr>
<td>8-9 Verification</td>
<td>8-10 Documentation</td>
</tr>
<tr>
<td>8-11 Confidence in the use of software tools</td>
<td></td>
</tr>
<tr>
<td>8-12 Qualification of software components</td>
<td></td>
</tr>
<tr>
<td>8-13 Qualification of hardware components</td>
<td></td>
</tr>
<tr>
<td>8-14 Proven in use argument</td>
<td></td>
</tr>
</tbody>
</table>

| 10. Guideline on ISO 26262 |
KOSTAL has many years of experience in safety management, starting with IEC61508 projects in 2003 (SIL3)
ISO26262 is managed by KOSTAL since 2010
There are 18 engineers have AFSP(Automotive Functional Safety Professional) certificate

ISO 26262 references (assessed projects)

**ASIL-C/D:**
- VW PQ25 Steering Angle Sensor (ASIL-D)
- SGM Electric Steering Column Lock(ASIL-D)
- Daimler BR222 Steering Angle Sensor (ASIL-D)

**ASIL-B:**
- ......
Safety relevant functionalities

Example

转向锁止作动器 (带转向锁止ECU)
Safety relevant functionalities

Example

转向前锁止马达

提供电源

为马达提供电源

转向锁止 ECU

锁止/开锁命令

电源控制 ECU

ID识别码存储器

验证 ECU

霍尔 IC

转向前锁止 ECU
What’s New for software development?

### Software unit testing

<table>
<thead>
<tr>
<th>Methods</th>
<th>ASIL</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>A</td>
</tr>
<tr>
<td>1a</td>
<td>++</td>
</tr>
<tr>
<td>1b</td>
<td>++</td>
</tr>
<tr>
<td>1c</td>
<td>+</td>
</tr>
<tr>
<td>1d</td>
<td>+</td>
</tr>
<tr>
<td>1e</td>
<td>+</td>
</tr>
</tbody>
</table>

### Methods for deriving test cases for Software unit testing

<table>
<thead>
<tr>
<th>Methods</th>
<th>ASIL</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>A</td>
</tr>
<tr>
<td>1a</td>
<td>++</td>
</tr>
<tr>
<td>1b</td>
<td>+</td>
</tr>
<tr>
<td>1c</td>
<td>+</td>
</tr>
<tr>
<td>1d</td>
<td>+</td>
</tr>
</tbody>
</table>

### Structural coverage metrics at the software unite level

<table>
<thead>
<tr>
<th>Methods</th>
<th>ASIL</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>A</td>
</tr>
<tr>
<td>1a</td>
<td>++</td>
</tr>
<tr>
<td>1b</td>
<td>+</td>
</tr>
<tr>
<td>1c</td>
<td>+</td>
</tr>
</tbody>
</table>
Back to Back TEST

SIL TEST

PIL TEST
This is test case 1.
The benefit of MBD at unit test phase

- Long test time
- Multi Test Tools
- More Test Cases

- More effective when executive test
- Test case reuse saves time
- Test result visualization
Conclusion

Advantages of Model-Based Design

- **Executable models**
  - unambiguous
  - only “one truth”

- **Simulation**
  - reduces “real” prototypes
  - systematic “what-if” analysis

- **Design with Simulation**

- **Continuous Test and Verification**
  - Test with Design
  - detects errors earlier

- **Automatic Code Generation**
  - minimizes coding errors

- **Time Efficiency**
Thank You