Panel Session 1 “Design Verification, Test”

Automotive Functional Safety Standard ISO26262 and Design Verification Technology

Contents

1. Company profile
2. Development technology trends of the automotive electronic system
3. Automotive Functional Safety Standard ISO26262
4. Virtual ECU application technology

June 8, 2012
Yoshihiro Miyazaki
Executive Chief Engineer, Electronic Platform Technology GM R&D Div.
Hitachi Automotive Systems, Ltd.

Copyright © Hitachi Automotive Systems, Ltd. All rights reserved
Hitachi advanced into domestic production of automotive electric parts in 1930. Having 80 years history in the automotive industry, Hitachi Automotive Systems, Ltd., was established on July 1st, 2009 by the split-off from Hitachi, Ltd.

<table>
<thead>
<tr>
<th><strong>Name</strong></th>
<th>Hitachi Automotive Systems, Ltd.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Business</strong></td>
<td>Development, manufacture, sales and services of automotive components, transportation related components, industrial machines and systems, etc</td>
</tr>
</tbody>
</table>
| **Representative** | Kunihiko Ohnuma  
President and Chief Executive Officer |
| **Established** | July 1, 2009 |
| **Headquarter** | Shin-Otemachi Bldg.  
2-1, Otemachi 2-chome, Chiyoda-ku, Tokyo, Japan |
| **Capital** | 15,000 million yen (Wholly-owned subsidiary of Hitachi, Ltd) |
| **Revenues** | 811.5 billion yen (Year ended March 31, 2012, Consolidated basis) |
With enlargement and advancement of the in-vehicle software, forming the common basis from the base of each software is becoming much effective and it is named as “electronic platform”.

**[In a wide sense] Electronic platform**

**[In a narrow sense] Electronic platform (Implementation platform)**

- Microcomputer, in-vehicle LAN, the basic OS, BIOS, communication software, etc.

**Development platform**

Methods and tools such as control model description, programming, verification, etc.

### Software architecture

**Electronic platform (implementation platform)**

- ECU for engines
  - Application Software
  - Base software (the basic OS, communication software)
  - Base hardware (microcomputers)
- ECU for AT
  - Application Software
  - Base software
  - Base hardware
- ECU for brakes
  - Application Software
  - Base software
  - Base hardware

**In-vehicle network (LAN)**

### Software development process

1. **Concept**
   - Control design
   - Design tests methods and tools supporting the development process
2. **Implementation**
   - Software design
   - Software test
3. **Conformity with vehicle/its verification**
Many problems come to the front with progress of applying electronic control
- Increase of the in-vehicle controller number
- Complexity, advancement of the control
- Enlargement of the in-vehicle controller software
- Keeping & improvement of reliability

Measures approach 1: Reduce things to be developed
Electronic platform (implementation platform)
\(\rightarrow\) standardization, high-level function
- Standardization of software hierarchical structure specifications
- Standardization of basic software specifications
- Standardization of applications software data interface specifications
\(\rightarrow\) industry standardization: AUTOSAR, JasPar

Measures approach 2: Ease and facilitate development work (abstraction, automation)
Development platform \(\rightarrow\) advancement, standardization
"model based development method"
- Control model description language, tool
- Modeling and simulation for the controller and the control target
- Automatic cord generation (programming-less)
\(\rightarrow\) industry standardization: JMAAB, the Society of Instrument and Control Engineers
2. Development technology trends of the automotive electronic system

Advancement / complexity of the in-vehicle control

Evolution from aggregate of the single function control to integrated control

- The outside world information
  - Camera
  - Radar
  - Other sensors

Extract information
- Target information

In-vehicle information system
- Map information
- Position information
- Infrastructure information

Outside recognition system
- Camera
- Radar
- Other sensors

ITS integration control
- Control target decision

Vehicle dynamics control
- Coordination of actuation systems/Regenerative brake systems, etc

Brake control

Suspension control

Drive control system
- Battery control
- Engine control
- Motor control

HEV control
- Energy management

Steering control

User attentions to safety of the electronic control system → becoming higher

Further advancement, complexity of the electronic control function

Correspondence to functional safety standard ISO26262 (2011/Nov. established)

Remarkable improvement in safety/efficiency/quality for verification is required
Automotive functional safety standard ISO26262 inherits characteristics from the higher level standard i.e. functional safety standard IEC61508. It also adds the adaptation for the automotive field shown below.

1. Introduction of Automotive Safety Integrity Level
   SIL in IEC61508: recognized as the property of the target failure rate
   ASIL in ISO26262: defined as the integrated safety requirement level with both random failure and systematic failure (including software bugs, etc.)
   ASIL A (lower level) ~ ASIL D (higher level)

2. Definition of H&R (Hazard analysis & Risk assessment) for the ASIL derivation
   Evaluated by three factors shown below
   E(Exposure) : frequency of cases exposed at the event or assumed driving status
   C(Controllability) : possibility or difficulty of avoidance
   S(Severity) : severity of damage or injury

<table>
<thead>
<tr>
<th></th>
<th>S1 [Light and moderate]</th>
<th>S2 [Severe]</th>
<th>S3 [fatal]</th>
</tr>
</thead>
<tbody>
<tr>
<td>E1 [very low]</td>
<td>E1 [very low]</td>
<td>QM</td>
<td>QM</td>
</tr>
<tr>
<td>E2 [low]</td>
<td>E2 [low]</td>
<td>QM</td>
<td>QM</td>
</tr>
<tr>
<td>E3 [medium]</td>
<td>E3 [medium]</td>
<td>QM</td>
<td>QM</td>
</tr>
<tr>
<td>E4 [high]</td>
<td>E4 [high]</td>
<td>QM</td>
<td>A</td>
</tr>
<tr>
<td>E1 [very low]</td>
<td>E1 [very low]</td>
<td>QM</td>
<td>QM</td>
</tr>
<tr>
<td>E2 [low]</td>
<td>E2 [low]</td>
<td>QM</td>
<td>QM</td>
</tr>
<tr>
<td>E3 [medium]</td>
<td>E3 [medium]</td>
<td>QM</td>
<td>A</td>
</tr>
<tr>
<td>E4 [high]</td>
<td>E4 [high]</td>
<td>A</td>
<td>R</td>
</tr>
<tr>
<td>E1 [very low]</td>
<td>E1 [very low]</td>
<td>QM</td>
<td>A</td>
</tr>
<tr>
<td>E2 [low]</td>
<td>E2 [low]</td>
<td>QM</td>
<td>A</td>
</tr>
<tr>
<td>E3 [medium]</td>
<td>E3 [medium]</td>
<td>A</td>
<td>B</td>
</tr>
<tr>
<td>E4 [high]</td>
<td>E4 [high]</td>
<td>B</td>
<td>C</td>
</tr>
</tbody>
</table>

* QM : Quality Management (no requirement to comply with ISO 26262)
Overview of ISO 26262

1. Vocabulary

2. Management of functional safety

3. Concept phase

4. Product development at the system level

5. Product development at the hardware level

6. Product development at the software level

7. Production and operation

8. Supporting processes

9. ASIL-oriented and safety-oriented analyses

10. Guideline for ISO 26262 understanding
Activities in Japan related to ISO26262

3. Automotive Functional Safety Standard ISO26262

- ISO
- JSAE (Society of Automotive Engineers of Japan, Inc.)
- JAMA (Japan Automobile Manufacturers Association, Inc.)
- JASPAR* (Japan Automotive Software Platform and ARchitecture)
- JARI (Japan Automobile Research Institute)

Deliberations of the standard
- ISO
- DIS
- FDIS
- IS

Translation and general information guidebook

Deliberations of the standard

ISO26262 guidebook

Functional Safety related WGs

Microcontroller standardization TF

Functional Safety related WGs

Guidebook (software, Microcontroller), demonstration experiment

* JASPAR (Japan Automotive Software Platform and ARchitecture)
  [the establishment] September, 2004 (the establishment of the standardization consortium by three Japan car makers)
  [activity contents] the non-competition domains such as in-vehicle LAN elemental technology, middleware, the software base by cooperation in Japanese makers
  [Activity 2010- ] Functional safety WG newly established: Formulation and evaluation of the functional safety requirement about the automotive electronic platform
  [Activity 2011- ] “Evaluation of transient fault effect” newly added as one of the activities

Copyright © Hitachi Automotive Systems, Ltd. All rights reserved
A lot of requirements of ISO26262 are similar to those of conventional quality management. But some requirements not included in the conventional ways are added.

- It is required to show evidence of design and verification based on the viewpoint of functional safety. (Report information necessary for audit, etc. shall be submitted.)
- Not \( \bigcirc \times \) (yes or no) judgment but quantitative judgment is required. (Example: diagnostic coverage)
3. Automotive Functional Safety Standard ISO26262

Metrics evaluation complied by ISO26262

Hardware Architecture Metrics:
meters for the assessment of the effectiveness of the hardware architecture with respect to safety

<table>
<thead>
<tr>
<th>ASIL</th>
<th>SPFM</th>
<th>LFM</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASIL D</td>
<td>≥99%</td>
<td>≥90%</td>
</tr>
<tr>
<td>ASIL C</td>
<td>≥97%</td>
<td>≥80%</td>
</tr>
<tr>
<td>ASIL B</td>
<td>(≥90%)</td>
<td>(≥60%)</td>
</tr>
</tbody>
</table>

BaumKuchen Model representation
Methodology of Approach to ISO26262

1. Analyze gaps against one’s company’s conventional development process and extract the lacked parts (gap analysis).
2. Focus attention on “highly recommended” (++) or higher level in ISO26262 at gap analysis (consider “highly recommended” (++) to be covered in principle).
3. Keep conventional level if the level of the conventional process is higher than ISO26262 requirement. (The level may be lowered from the viewpoint of ISO26262. But do not lower the level consciously.)
Correspondence work for ISO26262 (man-hour increase):
Traditional Japanese spirit of fight with bamboo spears can not win global business race
⇒ Apply recent development technologies and development tools
   Achieve more efficient and higher quality development process

ISO26262 standard describes recommendation to apply various development technologies and tools

**ISO26262 MUST requirement**

- Requirements management & traceability management and support tools (as for safety)
- Quantification of test coverage and support tools

**ISO26262 WANT requirement**

- Formal verification and support tools
- Virtual ECU simulator (Virtual HILS)
What is virtual ECU simulator?

Application of the virtual ECU simulator:

System, control: Implementation-related evaluation (execute time, operation load) of the electronic control system, necessary operational precision, error influence, implementation cost

Hardware: Microcomputer design (or selection), ECU design, ASIC development

Network: Communication error injection, network delay, decentralized control

Software: Run time task analysis, CPU load factor evaluation, the OS, middle software performance evaluation, FMEA test, exhaustive timing test (interrupts), HILS substitute

Tools (Example): Synopsys Inc./CoMET, GAIO TECHNOLOGY CO., LTD./No.1 System Simulator, etc.
An application example: Virtual HILS (vHILS)

- Target product system: ADAS controller
  Ranging with radar and Keeping safe distance against proceeding vehicles ahead (ACC function), etc.

**Virtual HILS (vHILS)**

1. ECU Model
   - ADAS ECU

2. CAN Model
   - CAN Bus Monitor

3. Vehicle Model
   - Engine
   - Body
   - HMI
   - Sensor

4. Event Processor
   - Display
   - Input
   - Test Specs

Note: Conventionally HILS with real ECU is used
HILS: Hardware-in-the-loop simulator

4. Virtual ECU application technology

The processing throughput by 3 parallel computing → evaluated result: equal to a real machine

more than a real machine to be feasible by N parallel processing

ADAS: Advanced Driver Assistance Systems
ACC: Adaptive Cruise Control
4. Virtual ECU application technology

The future of the software verification: V2Cloud

- Cloud computing for software verification
  - Large-scale VM environment: Facilitates sharing and management of the simulation
  - Complete automation: Scalable environment
  - Without having fixed assets, it is possible to enjoy the necessary target system configuration and test performance when needed

Test vectors described in a spreadsheet

<table>
<thead>
<tr>
<th>test vector</th>
<th>result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Engine Test</td>
<td>○</td>
</tr>
<tr>
<td>Brake Test</td>
<td>○</td>
</tr>
<tr>
<td>Body Test</td>
<td>×</td>
</tr>
<tr>
<td>Network Test</td>
<td>○</td>
</tr>
<tr>
<td>Fail Test</td>
<td>×</td>
</tr>
</tbody>
</table>

Expected (example): Massive regression tests or fault injection tests

HILS: several days → parallel VHILS on V2Cloud: one night only!
[Appendix] References

- JMAAB  http://jmaab.mathworks.jp/
- ISO26262
- Y. Ito et al, ”A Model Based Software Validation for Automotive Control Systems”, International Conference on Control, Automation and Systems (ICCAS), pp.102, 2010