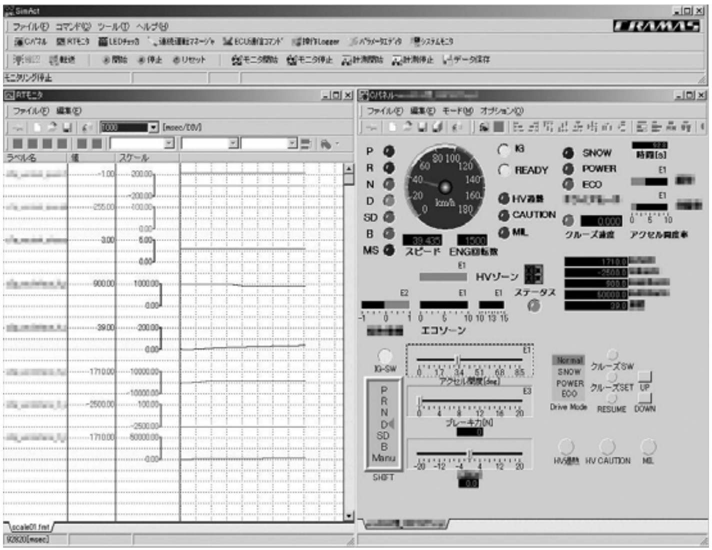
Use of SILS to Development of Control Application Specifications

Kenji HOSOKAWA
Mitsuo YOSHIDA
Tomohiro MATSUO
Tsuyoshi TAKATORI
Junji TAKAHASHI



Abstract

Conventionally, HILS (Hardware In the Loop Simulator) was used for verification of control application specifications. However, ECU (Electronic Control Unit) was rarely completed by the time of specification development. In that case, we modified a similar existing ECU and used it, but that increased the verification preparation man-hours.

Under these circumstances, due to the more intensive development competition among car manufacturers, we need to shorten more the development period of products to respond swiftly to market changes. To reduce the specification development man-hours, we upgraded the HILS to SILS (Software In the Loop Simulator) equipped with virtual ECUs that replace ECU hardware of HILS.

We used the SILS to the development of eco-driving-indicator application specifications. This paper describes the SILS employing the simulation method that does not require ISS and signals for the control application operation are sent to / received from the plant model in RAM values.

1 Introduction

Responding to the increasing demand for the eco-friendly and safe vehicles, vehicle control systems have become more complicated and larger in size year after year. Amid more intensive development competition among car manufacturers, shorter product development period has become a serious challenge for us to respond swiftly to the market change.

In order to deal with the shorter development period, many companies have improved their verification tools used for development of control application specifications and developed SILS (Software In the Loop Simulator) that enables applications to be verified by software simulation instead of testing them physically.

We used our developed SILS to the development of control application specifications and we will explain its technology and the verification results in this paper.

2 Background to Introduction of SILS to Specification Development

2.1 Changes of Simulators for Specification Development

Generally, in the development of control application specifications, ① specification design, ② software coding and ③ verification are conducted repeatedly, like running on a loop, to improve the application (**Fig. 1**).

Hereinafter, "verification" refers to the verification in specification development.

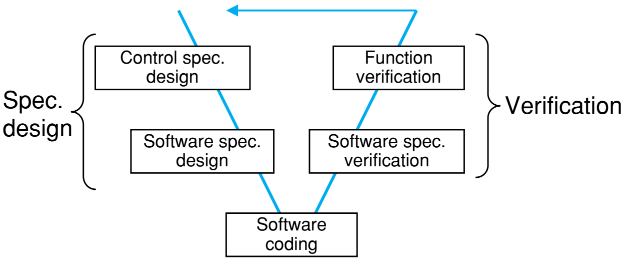


Fig.1 Verification Loop of Specification Development

Now, we take a brief look at the changes of "③verification" tools. In the era before a simulation tool was developed, specifications were verified using an actual vehicle. Since an actual vehicle needed to be prepared and an ECU (Electronic Control Unit) to be created for the evaluation, it took much time to design and verify the specifications according to the verification loop. In addition, total safety had to be confirmed before the verification because of the possible risk of breakdown or run-away of the vehicle during the verification (**Fig. 2**).

Later, to solve these problems, HILS (Hardware In-the Loop Simulator) tools were developed to simulate a vehicle under test on PC. It slashed man-hours for verification, enabled efficient verification on PC and eliminated the risk of damage to the vehicle. FUJITSU TEN produces HILS system under the name of CRAMAS (ComputeR Aided Multi-Analysis System) and sells it to car manufacturers and their supplier group companies.

The HILS simulation requires an ECU. However, the ECU was rarely completed by the time of the control application specification development. Conventionally, we modified a similar existing ECU and used it, but that increased man-hours in verification preparation.

Under these circumstances, as the shorter development period was requested, SILS, which software simulates all pieces of hardware such as the ECU and the bench system for exchanging signals with the ECU was developed. It consists of software only. As a result, it is cheaper than HILS and each engineer can use his / her SILS. Also, the SILS shortens the man-hours for preparing hardware system such as an ECU, wiring harness, an instrumentation board, etc.

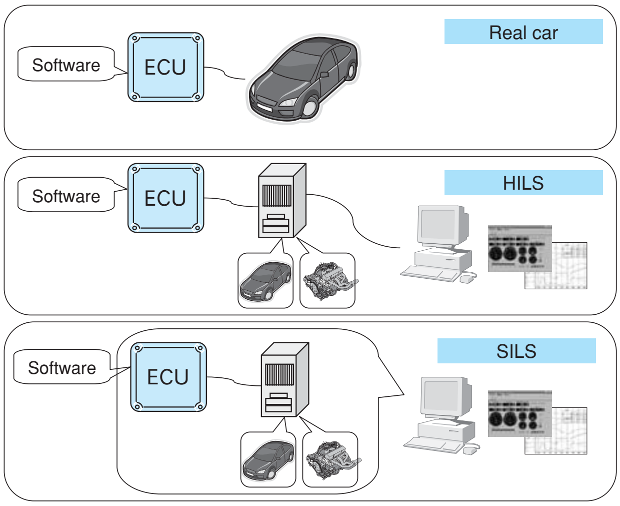


Fig.2 Changes of Verification Tool

2.2 Types of FUJITSU TEN SILS

Applying the technology used for CRAMAS, we developed the following three types of the SILS according to the intended purpose. Users choose and use an appropriate SILS type based on required speed and/or accuracy.

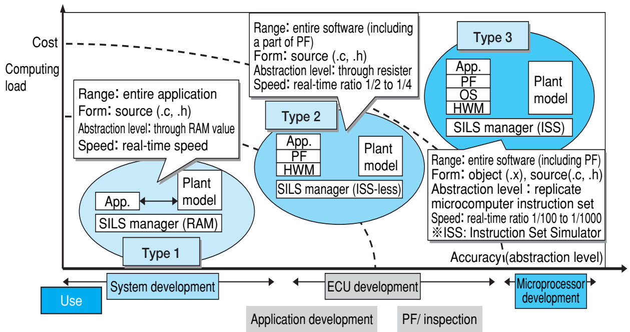


Fig.3 Types and Characteristics of Our SILS

Among them, the SILS type 1 is used for specification development this time because its computational load is so low that it is suitable for execution of a stand-along application.

Only for your reference, we use the SILS type 2 for when simulating the entire ECU software and the SILS type 3 for when simulating a microcomputer with high precision.

3 Outline of Eco-driving-Indicator

This section describes the outline of the "eco-driving-indicator," a control application of which specifications were developed with SILS this time.

This application has the function of supporting eco-driving to reduce CO_2 emission and fossil fuel consumption as measures against recent global warming. Car manufacturers have been introducing the application to their vehicles one after another.

The fuel is consumed much when a driver starts a car or is cruising. This application helps a driver avoid unnecessary behaviors (abrupt acceleration by suddenly pressing down on the accelerator and speed change by changing the pressure onto the accelerator).

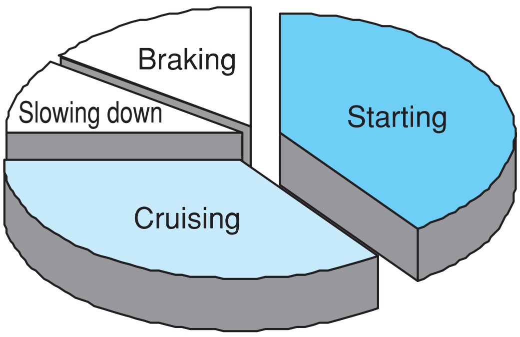


Fig.4 Driving Modes in a city and Fuel Consumption Ratio
(From the website of The Energy Conservation Center, Japan)

From the driving speed and the level of the pressed accelerator, this application judges whether or not the driver drives within the appropriate speed range in eco-driving technique. It supports eco-driving by showing the judgment result on a display device such as a lamp.

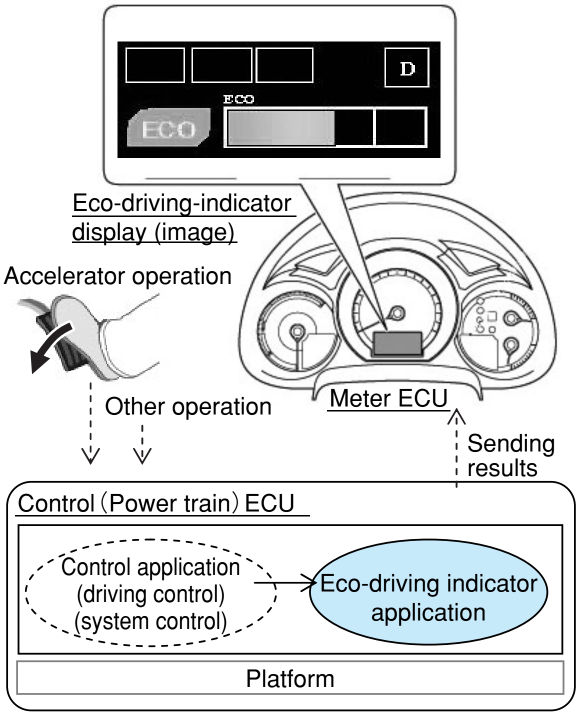


Fig.5 Outline Drawing of Eco-driving-Indicator

FUJITSU TEN developed the specifications of this application jointly with Toyota Motor Corporation and we used the SILS for the development.

If we had developed the specifications using a conventional method, we would have modified an existing ECU similar to this system and used it along with HILS. However, we decided to use the SILS to challenge a new development method without using hardware and also for the reasons ①, ② and ③ listed below.

①Shorter development period

②Design and verification of various specifications in a short time

③Inexpensive verification tool for multiple engineers to design and verify specifications simultaneously

4 Outline of SILS

Here, we explain the outline of the SILS that was used to help the specification developing engineers develop the specifications this time.

Fig. 6 shows the components of the entire system.

The exclusive platform (PF) and event manager that controls time are integrated in the user application, and it is built and executed in VC (Microsoft Visual C++) as one executing unit.

The executing unit is connected to Virtual CRAMAS DLL (Dynamic Link Library) and it receives and sends input / output signals from / to the plant model and CRAMAS GUI (Graphical User Interface) through the DLL.

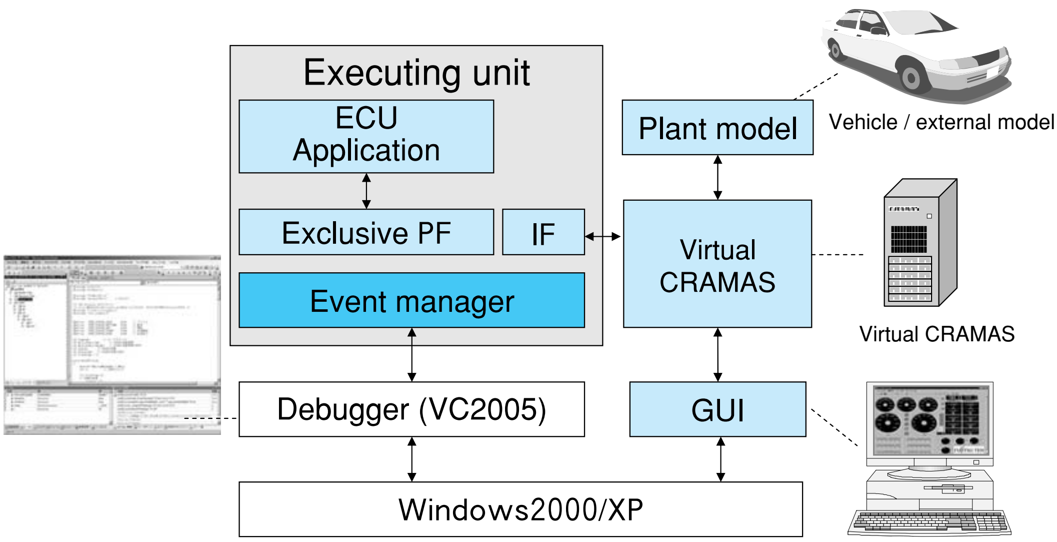


Fig.6 SILS Block Diagram

The following is the explanation of each component.

①ECU application

It is the software itself for which specifications are developed.

This application does not have to be modified, in principle.

②Exclusive PF / IF

It is the interface that allows the ECU application to run on the SILS. It provides the system that eliminates the need for changes in the ECU application.

③Event manager

It is a simulation manager developed by FUJITSU TEN.

It was developed with the aim of having no black box in the system in order to promptly handle problems such as specification defects in the advance application develop.

The event manager controls simulation time, which enables routine / non-routine processes of applications to be executed at high speed without any time conflicts (**Fig. 7**).

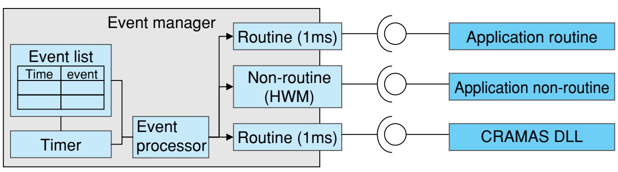


Fig.7 Event Manager Configuration Diagram

The following three are embedded in the event manager.

- Timer block that is the absolute time of the SILS simulation
- Inventory list that controls events and their starting time in pairs
- Event processor that processes those events

Since the event processor controls complicated starting time requested by routine / non-routine processes, even interrupt became to be simulated.

④Debugger (VC2005)

Among the standard functions offered in the non-embedded debugging environment are stopping or single-stepping of a program by breakpoint, and rerunning of a program after a RAM value is changed. However, HILS uses hardware so that even if the program stops, the hardware other than the microcomputer continues working and it cannot maintain the conditions of when the program is stopped until it restarts. Therefore, the verification cannot be continued.

However, when using the SILS, various VC debugging functions including breakpoint are all available. When a program stops, the hardware stops simultaneously so that the program can be restarted in the same conditions as those of when it stops, as if time would have stopped.

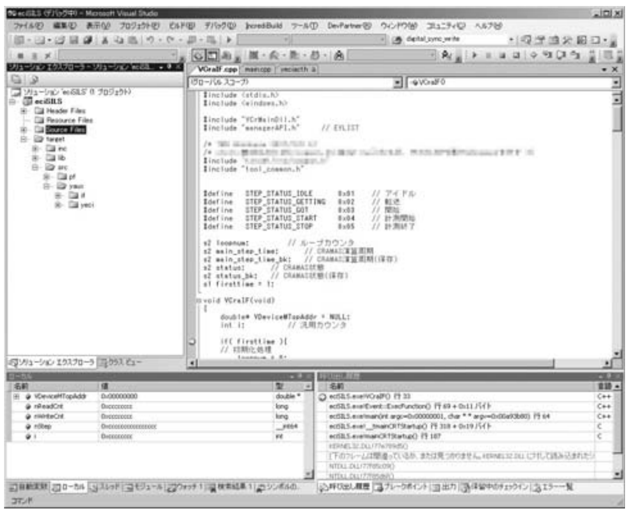


Fig.8 VC Debugger Image

⑤Plant model

A vehicle and other hardware that the application controls are connected as the plant model.

⑥Virtual CRAMAS / GUI

It starts / ends simulation or inputs / outputs data.

Fig. 9 shows an image of an actual simulating screen.

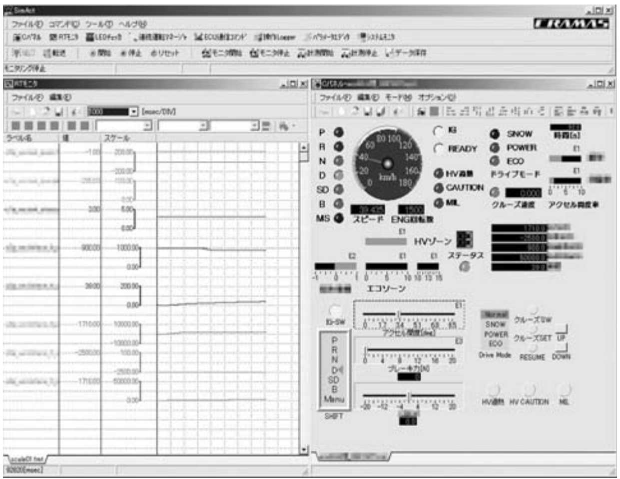


Fig.9 Image of Actual Screen

5 Use Case Example of SILS

In this section, we introduce the use case example of the SILS used for the eco-driving-indicator specification development.

5.1 Use in Eco-driving-Indicator Specification Development

The SILS type 1 that we used this time receives and sends data from / to the plant model in RAM value. Therefore, an interface layer is formed between the application and the plant model for the traditional HILS (**Fig. 10**).

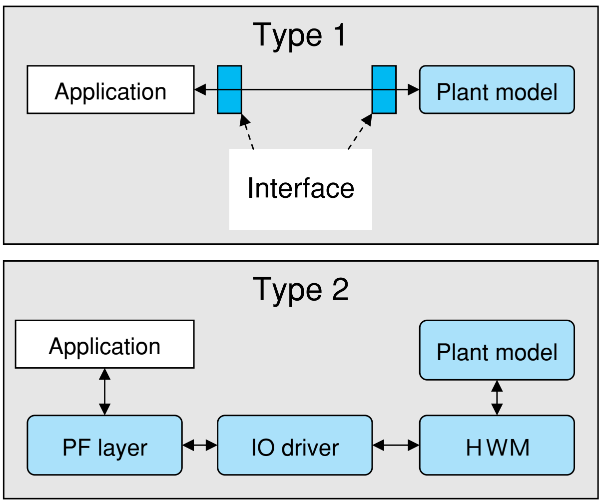


Fig.10 Difference between Type 1 and Type 2

We developed the eco-driving-indicator specifications for multiple intended vehicles and they have their own interfaces. However, that offered an opportunity for us to verify the specifications' compatibility with all those interfaces. For example, when adding signals to the application as input from the model plant, the specification developing engineer can easily add them by changing only the interface without creating HWM (HardWare Model) that requires hardware knowledge. Moreover, as shown in **Fig. 11**, the specifications can be verified effectively because multiple engineers can verify them simultaneously.

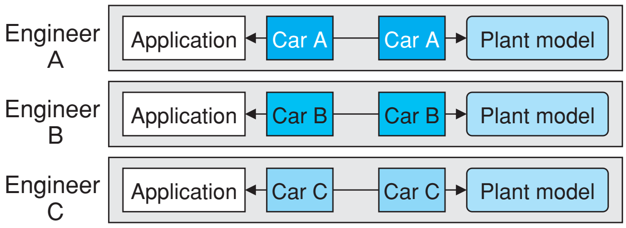


Fig.11 Utilization of Interface Layer of Type 1

In the next couple of sections, we introduce the SILS tools used for the eco-driving-indicator specification development.

5.2 GUI

The following three segments on the main screen are used.

(1) Execution control bar

It is used to give commands of overall SILS operation such as start / end of simulation / measurement.

(2) RT (Real Time) monitoring segment

It displays random RAM signals in chronological order.

It monitors the ever-changing waveform in real time (virtual-time simulation).

(3) C (Control) panel segment

It is the interface for inputs / outputs to the application.

A user can create it by combining GUI components arbitrarily.

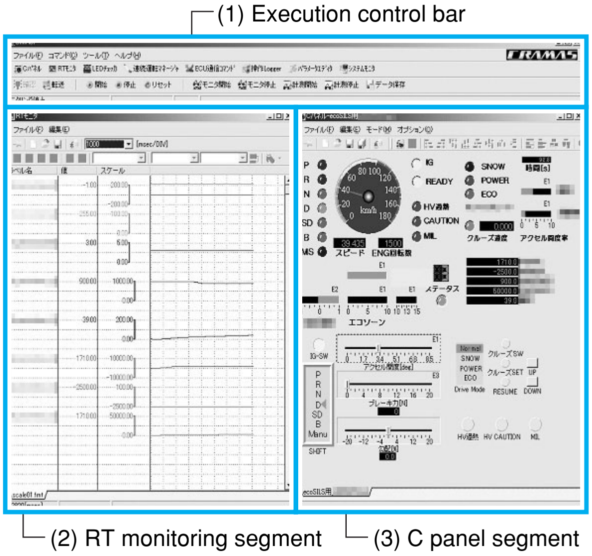


Fig.12 SILS GUI Screen

We now explain the eco-driving-indicator screen of (3) C panel.

(A) Output segment

It displays various values including the vehicle speed, RPM of the engine, the state of display lamps, eco-driving-indicator, RAM physical values that the specification developing engineer prefers to monitor. We created some virtual eco-driving-indicator displays based on the actual ones of the intended vehicles and chose one of them for the verification in order to realize the environment similar to the actual eco-driving-indicator of each of those vehicles.

(B) Input segment

It is equipped with indicators of the shift position, accelerator opening degree, brake force, various mode switches, etc.

The virtual vehicle is controlled by operating the input segment, monitoring the above output segment that changes in real time.

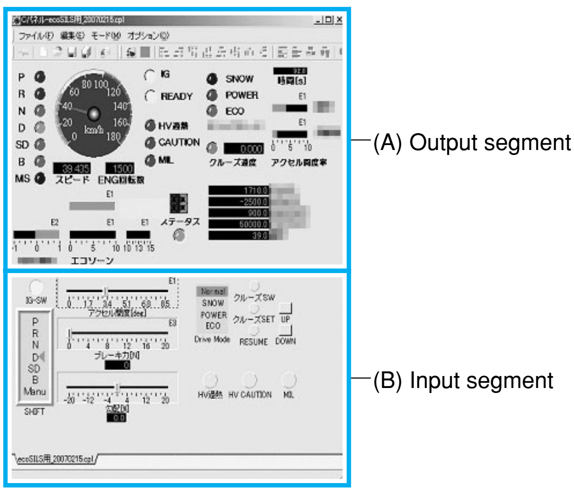


Fig.13 GUI (Input / Output Segments)

5.3 Plant Model

The plant model includes the block below that affects eco-driving-indicator control.

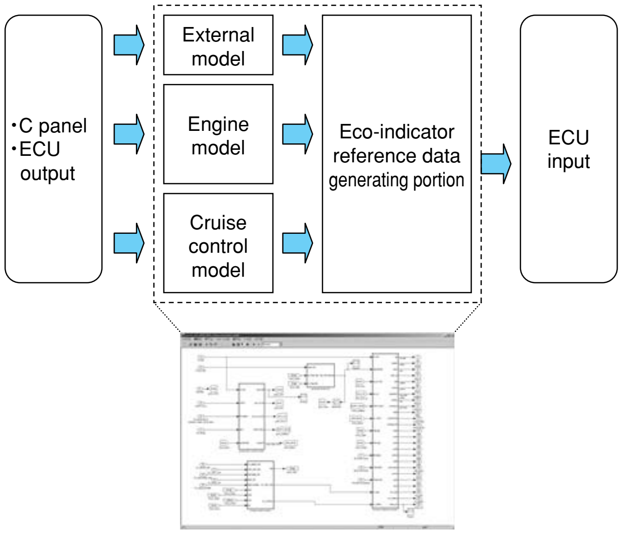


Fig.14 Block Diagram of Plant Model

・Engine model

Monitoring the current vehicle speed and RPM of the engine, this model changes the vehicle conditions with the accelerator opening degree, brake force, etc.

・External application models

These are other application models that affect the eco-driving-indicator such as cruise control.

・External model

This model computes the external environment such as slope, friction coefficient of the road, etc., that affects the vehicle behavior.

6 Result Confirmation

This time, the specifications were developed before the ECU was completed. When it was completed and the application was installed in it (**Fig. 15**), no problems were found in the application function and the application was installed only by adjusting the interface to the actual ECU.

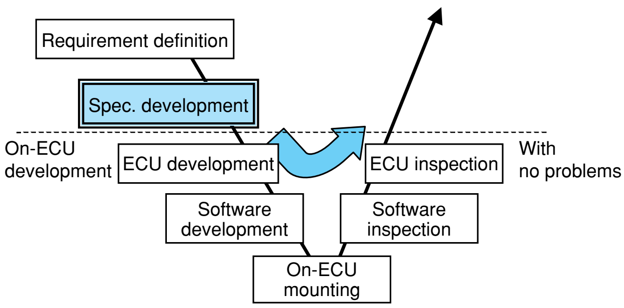


Fig.15 V-model Process from Specification Development to ECU Inspection

This proves that the SILS provided the functions equivalent to the actual vehicle and hardware even without using the ECU. It is a significant achievement.

In particular, mismatches and conflicts among individual specifications can be found only by monitoring actual vehicle behaviors in many cases. By using the SILS, we could find problems in the early stage, which gave us enough time to examine them with the customer and taking measures against them. That is another accomplishment.

As for the realization of a inexpensive verification tool, since VC2005 (Express Edition) was provided by Microsoft for free and other components were made in-house, the SILS was created at very low cost. Therefore, each specification developing engineer can have his / her SILS and so one of our objectives of providing a reasonable verification tool for each engineer was achieve.

7 Conclusion

The eco-driving-indicator system developed this time was relatively smaller than other power train control systems so that its SILS could easily be created. In order to use SILS for development of other control systems in the future, it will be the key as to how fast and how accurately the SILS can be created. Therefore, issues and problems that became evident in this development need to be solved completely.

In this challenge, we acquired the skills to prepare development simulator SILS in a short time and to develop high-quality application specifications even without using an actual vehicle and hardware.

In order to cope with growing control systems and to meet the request for shorter development period, it is vital to innovate, use and share simulation tools like this one, actively. We hope to continue our work to complete this SILS as a standardized tool for specification development.

Profiles of Writers

Kenji HOSOKAWA

Entered the company in 1990. Since then, has engaged in the development environment technology and the development of design assistant tools. Currently, in the System Development Department, Research & Technology Development Division, AE Group.

Mitsuo YOSHIDA

Entered the company in 1997. Since then, has engaged in the development of in-vehicle electronic devices and the process construction of the next-generation software development. Currently, in the Advanced Design Process Planning Department, Control System Development Division, AE Group.

Tomohiro MATSUO

Entered the company in 2001. Since then, has engaged in the development of in-vehicle electronic devices. Currently, in the System Development Department, Research & Technology Development Division, AE Group.

Tsuyoshi TAKATORI

Entered the company in 1993. Since then, has engaged in the development of in-vehicle electronic devices. Currently, the Team Leader of System Development Department, Research & Technology Development Division, AE Group.

Junji TAKAHASHI

Entered the company in 1984. Since then, has engaged in the development of in-vehicle electronic devices. Currently, the Department General Manager of System Development Department, Research & Technology Development Division, AE Group.