



Application of MBD to Development of ECU Prototype for EPS

KOBAYASHI Masayuki

1 Introduction

Conventionally, most of the embedded control systems have been developed, using a document-based design and hand-coding. However, the high quality, low cost, and shorter development lead-time are expected, while increasing demand for higher performance and more diversified functionality in electronic control systems mainly in automotive industry. Since efficiency improvement in product development is expected also for one of KYB's major electronic control systems, the electric power steering (hereafter referred to as EPS), innovative process and methods differing from that of the past are needed.

In recent year, The model-based development (hereafter referred to as MBD), which enables the visualization of design and implementation, has drawn attention as a development method of embedded system. ^{Note 1)} This method in fact has been widely used and established a considerable track record in the automotive industry.

This document describes our approaches for MBD

implementation in ECU prototype development for EPS.

Note 1) An abbreviation of "Model Based Development."

2 Moves to MBD Implementation

2.1 Issues in a previous development method

Specifications are always required in the development of embedded systems, however it is difficult to make users fully understood the the contents of document. For example, there is a risk of experiencing unwanted iterations when some requirements are misunderstood by a user, resulting in a failure detected in test process. Actually, this type of problem had been seen frequently in a manufacturing floor, and solved by spending numerous resources. However, the present method is no longer able to sufficiently respond the demand for improved efficiency in product development today.

2.2 MBD Overview

MBD is the development method that uses the CAE tool with it's simulation capabilities ^{Note 2)} during the development process of embedded control systems in

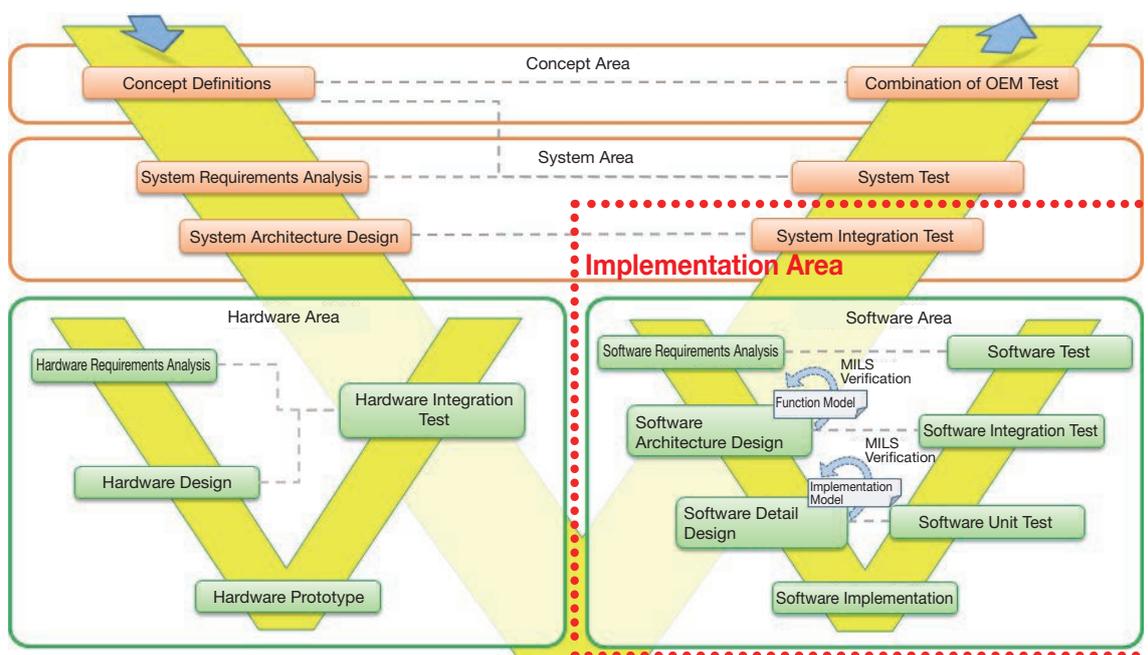


Fig. 1 V-Shape model of development life cycle

order to improve quality and development efficiency thoroughly in the development life cycle as illustrated in Fig. 1. The left half of V shape illustrates the design phase while the right half illustrates the test phase. Function model is created/verified in software architectural design. Implemented model is created/verified in a detailed design. In this development, focusing on the software, the system integration test was conducted as software verification.

Features and advantages of MBD implantation are described as follows:

(1) Clarification of specifications

An intuitive, easy-to-understand specifications can be created by using common language, "model" in place of two different words such as "specification" and "function." This also helps communication with overseas affiliates.

(2) Front Loading

The validity of specification is verified in each simulation since a specification is feasible model. This is called MILS,^{Note 3)} and the combination of control model and plant model^{Note 4)} enables an operation check for the developing product even from its design phase. A risk of experiencing unwanted iterations can be reduced by giving special emphasis to the upstream processes of development.

(3) Automatic Code Generation

Since source codes can be automatically generated from the model, a consistent level of quality, such as readability and execution efficiency, can be maintained with regards to human errors and variation in skills of programmers. Furthermore, the manpower required for coding can be significantly reduced. The specification and software prepared in equal without fail by code generation facilitates the management.

(4) Reuse

The past developed models are stored in library so that they can be easily reused at the specifications level. As a result, the accumulation of expertise and asset enhancement can be achieved as well as improved development efficiency. The above explanations can be applied to the plant model as well as the control model.

Note 2) Defines a series of processes, ranging from design through development, operation, and maintenance.

Note 3) An abbreviation of "Model In The Loop Simulation." Simulation performed by combining a control target model and a controller model .

Note 4) A physical model in which the motion of the control target such as a motor is substituted into the motion equation.

3 Software Area

In order to limit the effect of developing model to an application part for this software, AUTOSAR^{Note 5)}-compliant software components was used (Fig. 2).

Application Layer is the application part, where the model-based development is applied.

BSW Layer realizes the connection between software

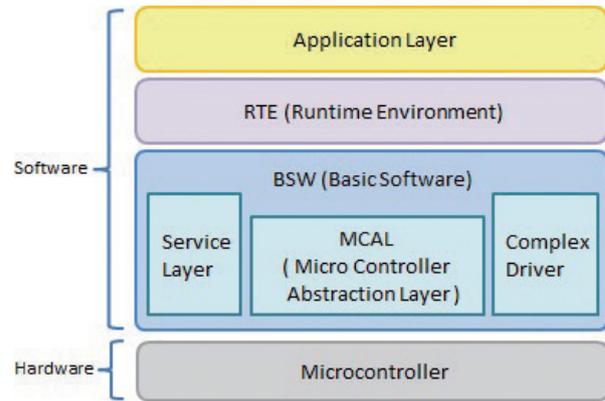


Fig. 2 AUTOSAR Configuration

and hardware, and consists of Service Layer and MCAL^{Note 6)} that provide OS services, including complex drivers that is required for using sophisticated functions.

RTE is a communication layer that realizes the connection between BSW Layer and Application Layer.

Having layer components as described above, applications is treated as one reusable module, and the development can be performed without awareness of Hardware.

Note 5) An abbreviation of "Automotive Open System Architecture." The organization and specifications that standardizes automotive software platforms.

Note 6) An abbreviation of "Microcontroller Abstraction Layer." Software module that provides an access to inside of microcontroller.

4 Developing Process

The work carried out in this development based on the development process is described as follows.

4.1 Requirements Analysis

Extracts requirements from the concept to be developed, and reflects the specific methods for realizing requirement to the software specifications (Fig. 3).

4.2 Function Model

Converts the function into a model based on the output of requirements analysis in the architecture design (Fig. 4). MILS is performed in this function model in order to verify/confirm the feasibility of the system operation, including fail-safe and requirements. To increase readability of model, the modeling guidelines were created in modeling process to ensure the description rule based on MAAB^{Note 7)}.

For modeling tool, the MathWorks MATLAB[®]/Simulink[®]^{Note 8)} that has already been used in our company was used . The model which has been developed by our company was used for plant model.

Note 7) An abbreviation of "Mathworks Automotive Advisory Board". Guidelines which provides the rules such as description rules for Mathworks product.

Note 8) Graphical environment for algorithm development and system simulation. MATLAB[®], Simulink[®] are registered trademarks of the MathWorks.

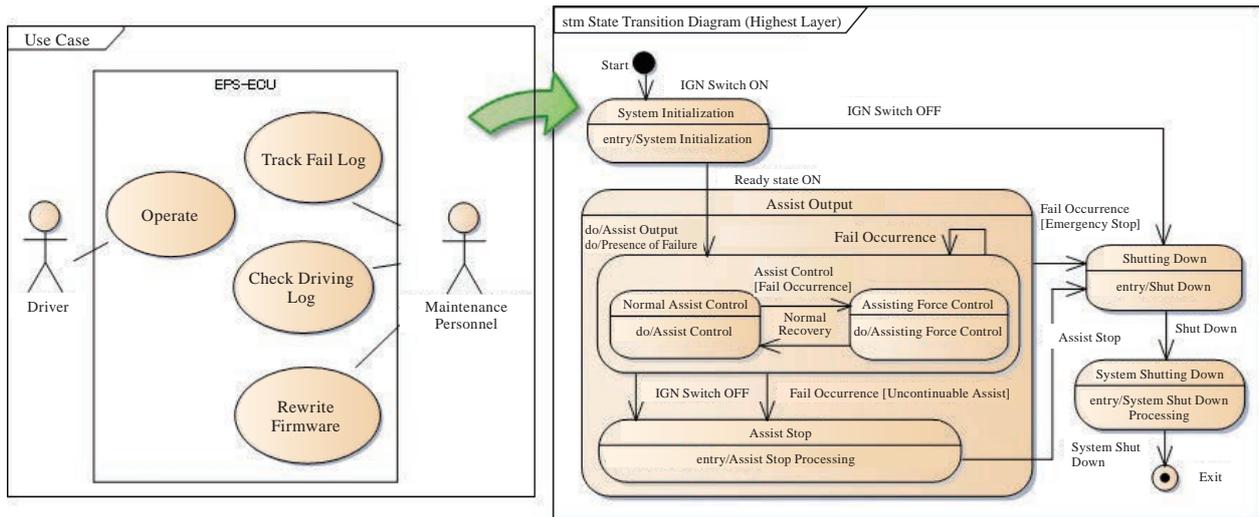


Fig. 3 Software requirements analysis

4.3 Implementation Model

Modifies the function model consciously of the memory performance of the program with an ECU equipped. The discretization of continuum model and the separation of individual component, the modification including the optimization of the variable type are performed. Then, the equivalency between function model and implementation model were verified by Back-to-Back test (Fig. 5).

4.4 Implementation

Provides code generation for implementation model. For generated C code, the conformance of MISRA-C^{Note 9)} rules are checked by using a static analysis tool and appropriate corrective measures are implemented for nonconforming items.

Note 9) Coding standard for securing the security, portability, and reliability of software (C language).

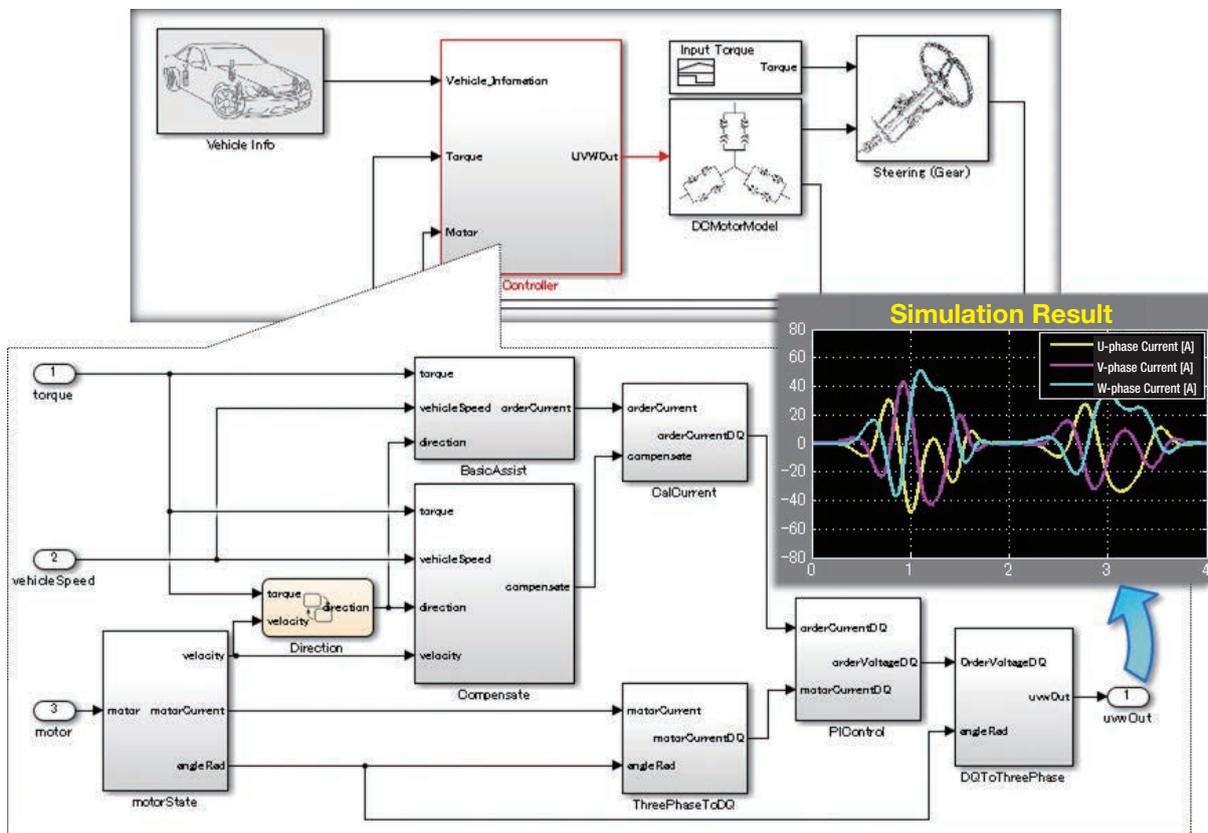


Fig. 4 Function model created by MATLAB®/Simulink®

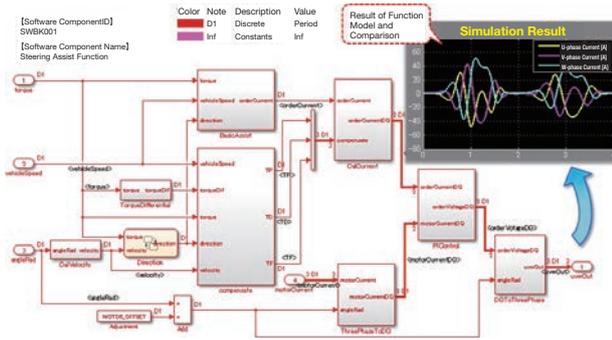


Fig. 5 Implemented model created by MATLAB®/ Simulink®

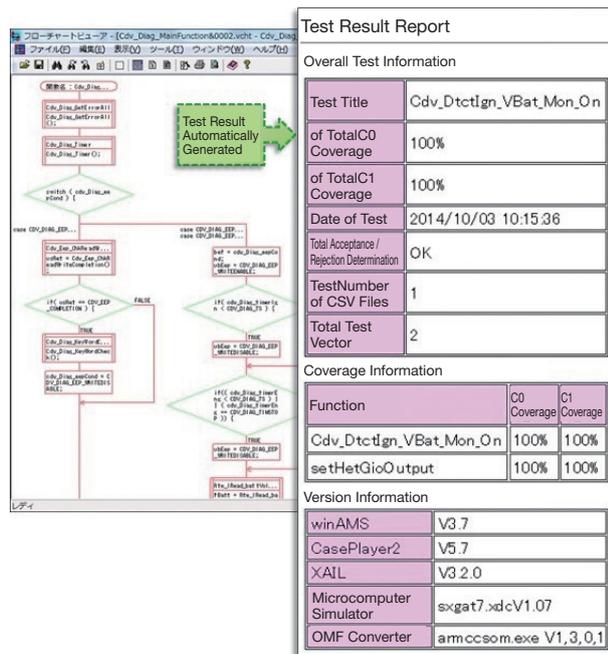


Fig. 6 Coverage Measurement

4.5 Test

Tests required during this development is described as follows:

4.5.1 Software Unit Test

Conduct the unit test for each software unit by using microcomputer simulator. A consistency with the detailed design is verified by Back-to-Back test and coverage measurement (Fig. 6). Also, an ISO26262^{Note 10)} compliant unit test tool is used to automatically generate the evidence.

Note 10) Specification of safety requirements for all automotive electronic and electrical systems.

4.5.2 Hardware integration test

Software units are rationally integrated to conduct a test using the microcomputer simulator in combination with test program. The proper operation of integrated software is confirmed, and its compatibility with the software architecture design is verified (Fig. 7).

4.5.3 Software Test

A sensor and a control target are combined with ECU equipped with the integrated software to conduct a test. Also, the memory usage and CPU shares in a

microcomputer are measured to verify the consistency with software requirement.

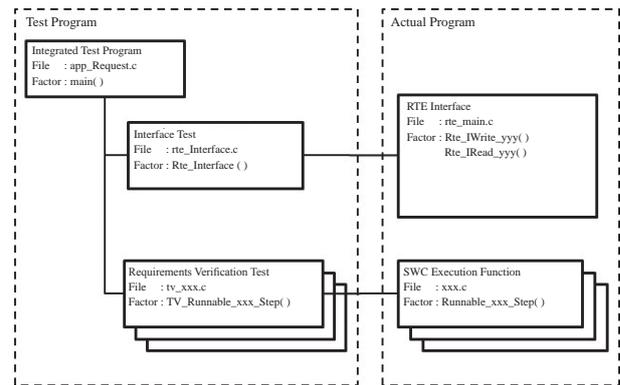


Fig. 7 Program configuration in software integration test

4.5.4 System integration test

Provides rational integration of the individual components of the system Besides the basic functions, the robustness was evaluated by using the bench test device, which is equipped with a steering and gear box (Photo 1). The system operating waveform generated in the steering operation of bench test device is shown in Fig. 8.

Its compatibility with system architecture design is verified, and the completion of system is confirmed from the evidence of test result.

5 Summary and future tasks

Using the model in the upstream process enables the early verification of required specification validity in the design phase, resulting in quality improvement. Also, the common understanding among project members was successfully improved and the communication in the development process was facilitated by utilizing the model as communication tool. Consequently, the work efficiency has been improved, and the benefits of using MBD in a multiple people development project were confirmed. Efforts were also made to the preparation of the operation environment such as development of guidelines and improvement in fundamental activities for full implementation of MBD. We would like to horizontally spread expertise that can be adopted in development of the EPS-derived products and other project.

However, there are some obstacles. Since multiple tools are often used for MBD, the basic overall knowledge about each tool must be acquired. Therefore, it may be a time consuming to train the MBD engineers. Also, the environment construction such as HILS^{Note 11)}, and the overall development process, including development of valuable human resources, require considerable funding. Therefore, the hurdle for utilizing this technology is fairly high.

Note 11) An abbreviation of "Hardware In The Loop Simulation." Simulation using an actual ECU with a model that simulates an actual vehicle.

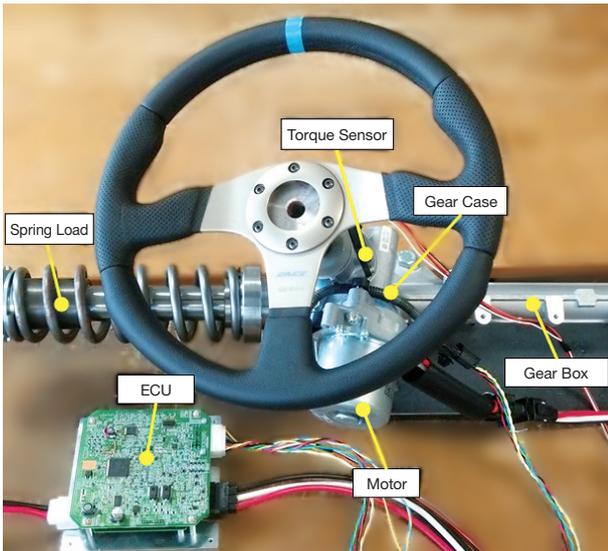


Photo 1 Bench test device

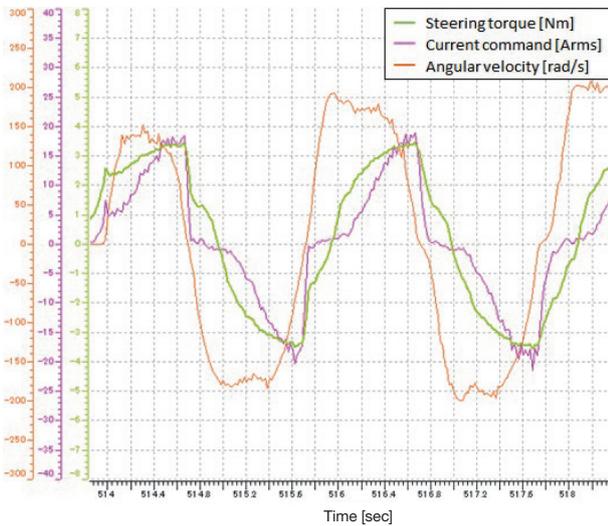


Fig. 8 The operating waveform when rotating the steering by 90° to the left and the right

6 In Closing

In recent years, remarkable progress has been made in automotive technology, and the vehicles equipped with driver assistance systems are no longer unusual. Most driver assistance systems, typically the lane keeping assist, are closely related to EPS. When function of "steering assist + α " is added to EPS, the complexity of systems will be increased, leading to bloated software. Under these circumstances, it would be a necessary consequence that MBD is more commonly used in standard processes.

Besides MBD, there are always new topics in automotive industry such as AUTOSAR and ISO26262, which are applied in our project, and the fault tolerant^{Note 12)} design. To respond to such changes in needs, and standards, we, as engineers, should always pay careful attention to the industry trend and prepare the appropriate environment and system.

Note 12) This enables a continuous operation without stopping in the presence of faults in some of its components.

Author



KOBAYASHI Masayuki

Joined the company in 2010.
Development Sect., Electronics
Technology Center, Engineering Div.
Engaged in software development.