

Application of Model-based Development to MICU

Noboru UJIGAWA*

Shuhei KINOSHITA*

Masayuki YAMAZAKI*

ABSTRACT

The introduction of model-based development is being promoted to increase specification quality in the development of software for body electrical parts. This introduction requires model-based development processes and development tools. Therefore, a model-based development process was constructed that covers from analysis of requirements in the early stages of development to definition of the methods for achieving those requirements. In addition, the tools needed to efficiently promote development were developed. The constructed process was applied to trial development of a Multiplex Integrated Control Unit, and resulted in a 35% reduction in incompleteness (undefined, conflicting or ambiguous specifications) compared to control specifications created using the previous development process.

1. Introduction

As the importance of electronic control technology in automobiles increases, the scale and complexity of software is increasing steadily⁽¹⁾. However, demands to increase quality, shorten the development time, and reduce the cost of electronic control units are also increasing. Amidst these circumstances, the introduction of model-based development (MBD) is recently being promoted to increase the quality of software mounted in vehicles⁽²⁾⁻⁽⁴⁾.

The control specifications for Multiplex Integrated Control Units (MICU) that perform integrated control of body electrical functions such as door locks, wipers, headlights, turn signal lamps and hazard lamps are presented to suppliers, and suppliers develop parts based on those specifications.

However, issues concerning the control specifications presented to suppliers included clearly defining the requirements and specifications that form the basis of the specification definitions, how to ascertain the compatibility of common use design specifications, and performing unambiguous design.

An MBD process was constructed and trial MICU development was performed to address the above issues. This paper describes the constructed process and developed tools, and the effects obtained from trial development.

2. Goals of Process Construction

The goals of constructing an MBD process were established as follows.

(1) Creating a system that enables separation of

requirements and specifications, analysis of requirements from multiple viewpoints, and detection of insufficient requirement definitions

(2) Creating a system that allows clear expression of the relationships between requirements and specifications, and association of the reasons and other information concerning those relationships.

(3) Creating a system that enables reduction of specification ambiguity by using a semi-formal specification description language

3. Construction of the Development Process

3.1. Process Overview

The constructed MBD process consists of analysis viewpoints and levels (Fig. 1).

3.1.1. Viewpoints

Analysis is performed from multiple viewpoints to enable detection of insufficient requirement definitions in the early stages of development. These analysis viewpoints are as follows.

- Behavior: The required behavior and state transitions of the system are defined.
- Structure: The static composition and structure of the system are defined.
- Requirement: The required contents of the system are defined.

3.1.2. Levels

(1) System definition: The subject of development (what) is clarified and the system for which requirements are to be

* Automobile R&D Center

defined is specified. The system scope, role, functions and relationships with other systems are defined in this level.

- (2) Requirement definition: The demanded behavior, structure and requirements of the specified system are defined.
- (3) Method definition: The specific methods (how) for realizing the system are defined. These methods are associated with the behavior, structure and requirements defined by the requirement definition, and each is assigned a reason (why).

3.1.3. Work product

Figure 2 shows the work product created for each viewpoint and level. These work products are created mainly using System Modeling Language (SysML)⁽⁵⁾ diagrams, MATLAB/Simulink and Stateflow⁽⁶⁾. Here, the minimum work product required for each analysis viewpoint and level is selected from the multiple diagrams defined by SysML.

3.2. Process Details

Figure 3 shows the detailed workflow for the process configured by the analysis viewpoint and level framework.

3.2.1. System definition

- (1) Definition of development scope: The system to be developed is specified using a use case diagram. In addition, the users and other systems related to the specified system are defined (Fig. 4).
- Definition of system users: Persons (e.g., drivers) and things (e.g., other systems) that use the system are defined as actors.
- Definition of system functions: Functions required of the system from the viewpoints of the actors are defined as use cases.
- Definition of relationship between system users and functions: Use cases and the actors that execute (use)

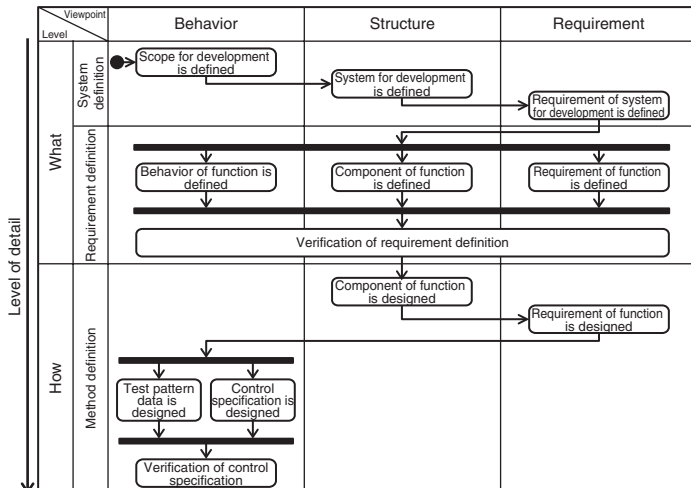


Fig. 3 Workflow of development

those use cases are associated.

- Definition of system boundaries: Use cases are enclosed by system boundary frames, and the system to be developed is defined.

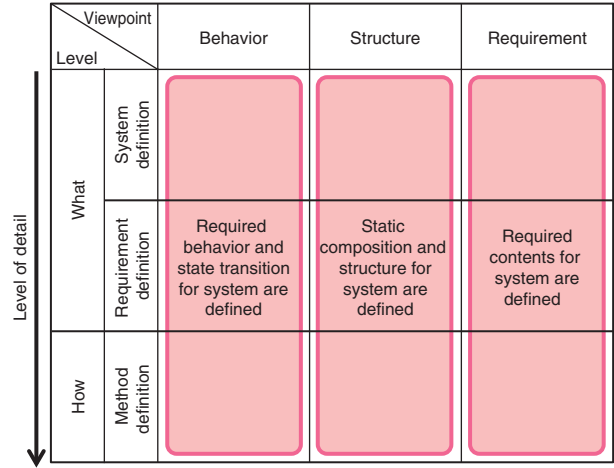


Fig. 1 Relationship between process and analysis viewpoints

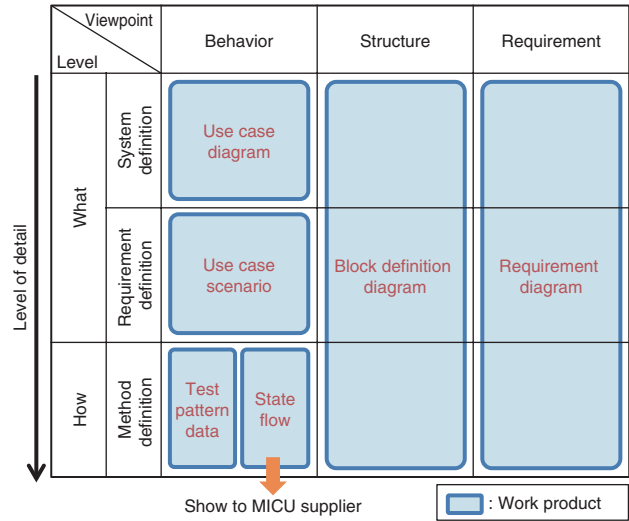


Fig. 2 Work product of each viewpoint and level

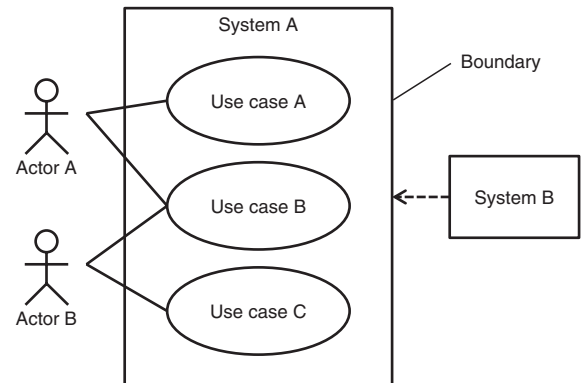


Fig. 4 Example of use case diagram

(2) Definition of system to be developed: The system to be developed is defined as a block definition diagram based on the use case diagram (Fig. 5).

- Boundary frames in the use case diagram are defined as the structure blocks in the block definition diagram, and information such as the system overview and body type is assigned.

(3) Definition of requirements of system to be developed: The requirements of the system to be developed are defined as a requirement diagram based on the use case diagram (Fig. 5).

- The use cases in the use case diagram are defined as the requirement blocks in the requirement diagram.

3.2.2. Requirement definition

(1) Analysis and definition of function behavior: Function behaviors are defined using scenarios based on the use case diagram (Fig. 6).

Scenarios define the interactions between the actors appearing in the use case diagram and the system, and the system behavior at that time. In this step the behavior is defined in various situations that can be conceived for the system.

• Definition of basic scenarios: The interactions between related actors and the system are defined as the basic scenario for each use case.

• Definition of expanded scenarios: Different patterns of precondition and post-condition, procedures leading to the post-condition, and actors for a basic scenario are derived and defined as the expanded scenario.

(2) Definition of function elements: The required system elements are defined in a block definition diagram based on the block definition diagram defined in the system definition process (Fig. 7).

The elements comprising the system are defined as structure blocks. In addition, the relationship between structure blocks is also defined. The definition process is as follows.

- The system elements needed to satisfy the requirements are defined as structure blocks from the viewpoints of input, control and output.
- The defined structure blocks are associated according to the relationship between upper and lower structures.
- The information exchanged between structure blocks

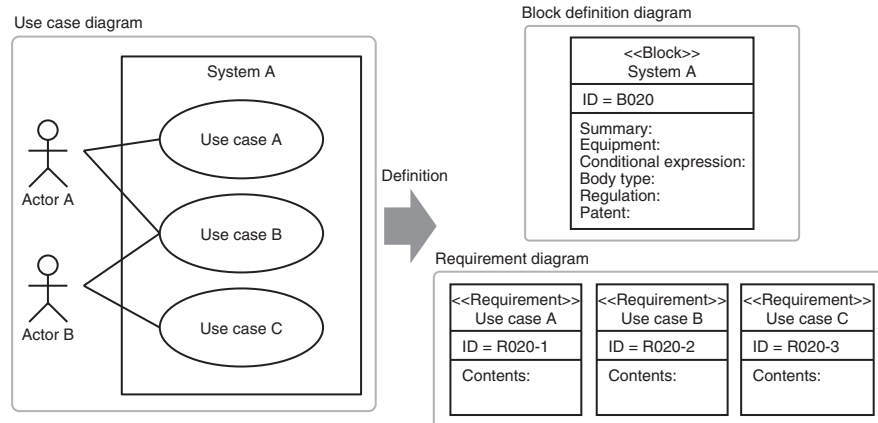


Fig. 5 Relationship of each diagram in system definition process

Scenario	
ID	UC020-1-1
Summary	...
Actor	Driver
Order	Contents
Precondition	Ignition is OFF
1	Driver turns on ignition
2	System outputs relay
3	End of Scenario
Postcondition	Relay is ON

Fig. 6 Example of use case scenario

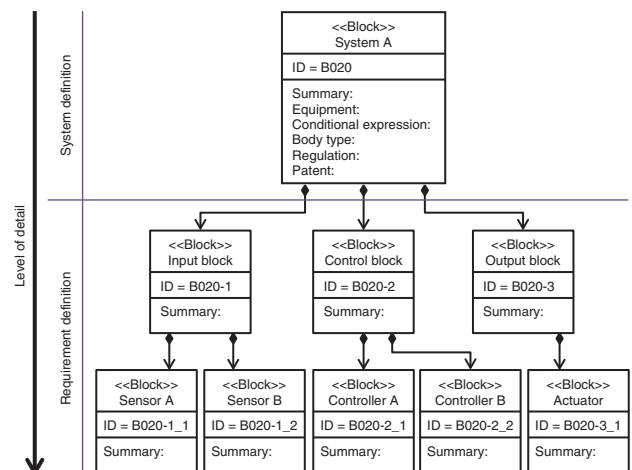


Fig. 7 Example of block definition diagram in requirement definition process

is defined in each structure block as input and output information (Fig. 8).

- (3) Analysis and definition of function requirements:
The requirements for each function are defined in a requirement diagram based on the requirement diagram defined in the system definition process (Fig. 9).
The requirements of each function are defined as requirement blocks, and the relationship between requirement blocks is defined. The definition procedure is shown below.
- The detailed requirements (e.g., operations, conditions, output, priority ranking in case of requirement conflict) of the use cases defined in the system definition process are extracted in consideration of system use scenarios, and defined as requirement blocks.
 - The defined requirement blocks are associated according to the relationship between upper and lower requirements.
 - The reason for each requirement is associated.
- (4) Verification of requirement definition: The work products created in the requirement definition process

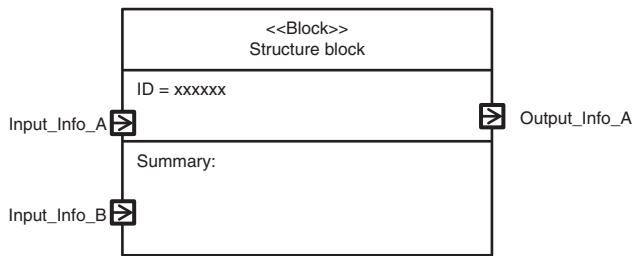


Fig. 8 Example of definition of input and output information to structure block

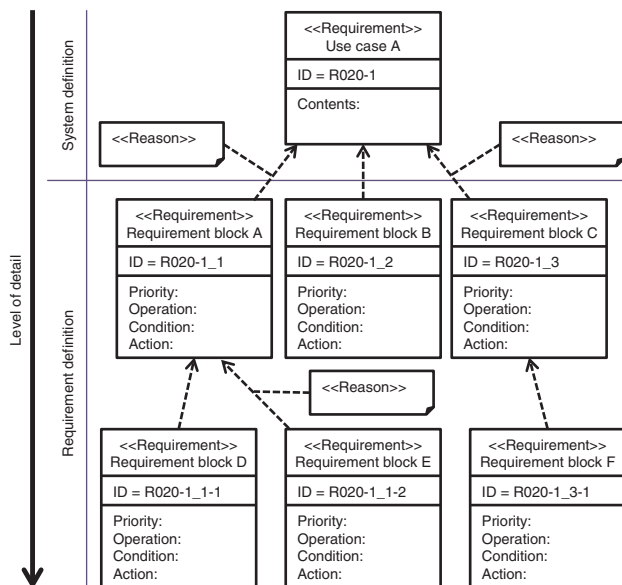


Fig. 9 Example of requirement diagram in requirement definition process

are compared as shown in Fig. 10 and verified from the following viewpoints.

- (a) Comparison of requirement diagram and scenarios
- Are the requirements needed to satisfy the scenario defined?
 - Are all scenarios associated with requirements?
- (b) Comparison of scenarios and block definition diagram
- Are the structures needed to satisfy the scenarios defined?
 - Is the input and output information needed to satisfy the scenarios defined?
- (c) Comparison of block definition diagram and requirement diagram
- Are the structures needed to satisfy the requirements defined?
 - Are all requirements associated with structures?
 - Is the input and output information needed to satisfy the requirements defined?

3.2.3. Method definition

- (1) Specification of function elements: The specific components are defined in a block definition diagram based on the block definition diagram defined in the requirement definition process (Fig. 11). The definition procedure is shown below.

- The specific components (e.g., switches, ECU, actuators) are defined as structure blocks for the structure blocks defined from the input, control and output viewpoints in the requirement definition process. At this time the information and evidence indicating the reason for selection are associated with the structure block. This associated information enables ascertaining of the compatibility of common

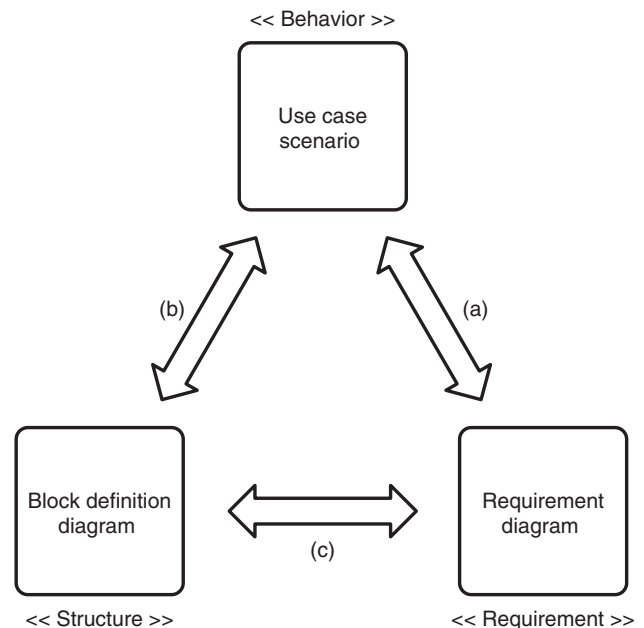


Fig. 10 Verification of work product

use design specifications in case of differential development such as minor changes or specification changes for each destination.

- Software elements are defined as structure blocks for the structure blocks defined from the control viewpoint in the requirement definition process.
- The defined structure blocks are associated according to the relationship between upper and lower structures.
- Electrical characteristics and other data are defined as attribute information for each defined structure block.
- The input and output information is defined for each structure block.

(2) Specification of function requirements: Methods are investigated and associated with the requirements in the requirement diagram based on the block definition diagram and requirement diagram defined in the requirement definition process.

Methods for the operations, conditions and outputs defined in the requirement diagram are defined in requirement blocks in consideration of restrictions arising from the hardware configuration and software processing methods, and these methods are associated with the requirements. In addition, the reasons for the methods are associated with the requirement blocks.

(3) Creation of test patterns: Test patterns are created on the basis of the scenarios defined in the requirement definition process and the input and output information defined in the block definition diagram (Fig. 12).

(4) Creation of control specifications: The Stateflow is

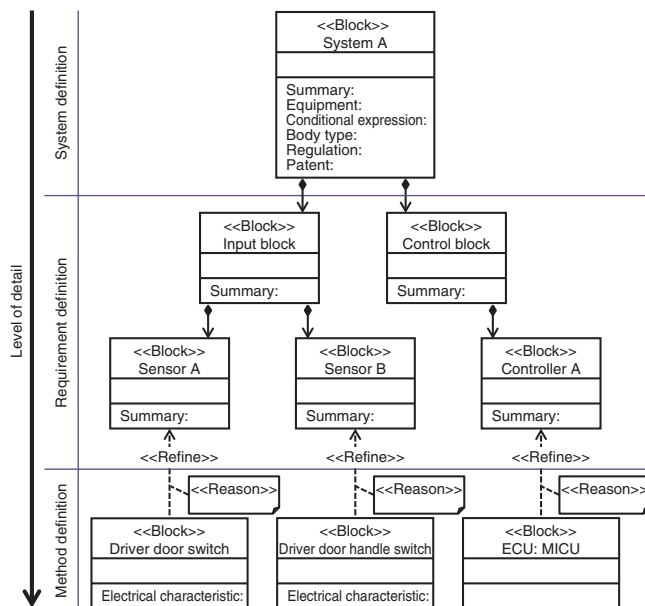


Fig. 11 Example of block definition diagram in method definition process

defined on the basis of the requirement diagram and block definition diagram created thus far (Fig. 13).

The detailed events required for control and the states that arise when events occur are defined using the states and transition conditions.

- Definition of states
Of the input and output information defined in the structure blocks of the block definition diagram, the output information is extracted as the states.
- Definition of transition conditions
Each condition described in the requirement blocks defined in the requirement diagram is defined as a condition for transition between states.
- States and transition conditions are added and deleted as necessary.

The Stateflow created here is based on description rules established for other process models as a result of hearings to collect requests from suppliers in order to present control specifications to suppliers.

(5) Verification of control: The Stateflow and test patterns are compared to verify the defined control specifications.

Time (sec)	Signal						
	Input					Output	
	Signal_A	Signal_B	Signal_C	Signal_D	Signal_E	Signal_F	Signal_G
0	0	0	0	0	1	0	0
1.00	0	1	0	0	1	0	0
2.00	0	1	0	0	1	0	0
3.00	0	0	0	0	1	0	0
4.00	0	0	0	0	1	0	0
5.00	0	0	0	0	1	1	0
6.00	0	0	0	0	1	1	0
7.00	0	0	2	0	1	0	1
8.00	0	0	2	0	1	0	1
9.00	0	0	2	0	1	0	0
10.00	0	0	2	0	1	0	0

Change point of value

Fig. 12 Example of test pattern data

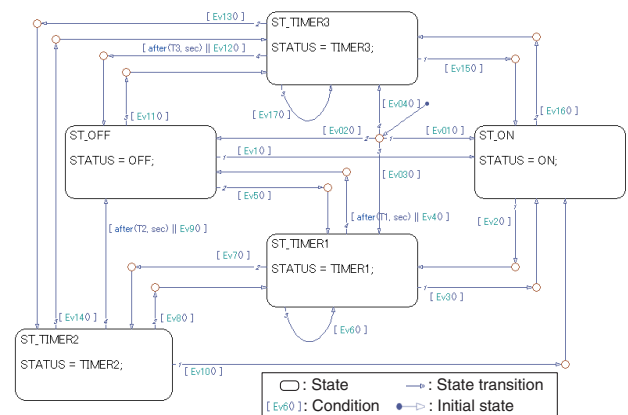


Fig. 13 Example of Stateflow

4. Construction of Development Tools

4.1. Synchronization of Input and Output Information

The input and output information of each block is defined in the structure blocks of the block definition diagram. However, many attributes such as name, direction, meaning, possible value, and initial value are assigned to input and output information, and it was a burden to describe the attributes for each input and output information in the model. In addition, complex procedures were also required to display the attribute information when examining the detailed description contents.

Therefore, a tool was developed that can edit the input and output attribute information and reflect the edited contents to the model using a list format user interface. This tool can also be used to output the model contents in list format (Fig. 14).

This tool enabled reduction of the work time required to assign input and output information to 1/5 that prior to tool introduction.

4.2. Automatic Assignment of IDs to Demands and Structural Elements

Unique IDs were assigned to requirement blocks and structure blocks. However, the number of blocks increases in proportion with the size of the system, so much work time was required to define the IDs one by one in the

model. In addition, repeated changes to models resulted in ID overlap.

Therefore, a tool was constructed that automatically assigns a unique ID to each block, enabling automation of the tasks of “Redefine all IDs,” “Define IDs only for elements (requirement and structure blocks) without ID definition,” and “Delete all IDs.”

This tool enabled complete prevention of ID duplication and shortened the work time required to assign IDs to 1/2 that prior to introduction.

5. Verification of the Process

Trial MICU development was performed in accordance with the MBD process to verify the constructed process. The number of specification checks from the supplier for the control specifications was used as the evaluation scale used to judge the advantages and disadvantages of the constructed process.

The condition was that if the number of specification checks classified as undefined and undescribed, conflicting, or ambiguous (multiple interpretations possible) in the breakdown of specification checks from the supplier could be reduced, the constructed process would be judged to have higher development quality than the previous development process.

The cooperation of the MICU supplier was obtained, and the control specification created using the MBD process was checked by the supplier from the same viewpoints as the previous development. As a result, the number of specification checks classified as undefined and undescribed, conflicting, or ambiguous decreased by 35%.

6. Conclusion

The newly constructed MBD process enabled organizing of the purpose of the system, the system configuration, and how each system element behaves, by analyzing the system from the three viewpoints of behavior, structure and requirements. Furthermore, the requirement quality could be increased by organizing and comparing each work product. In addition, association of requirements, reasons for requirements, and methods for achieving requirements enabled ascertaining of the compatibility of common use design specifications.

In the future, the constructed development process will be applied to mass-production development in order to steadily increase quality as well as continue to promote enhancements from the viewpoint of efficiency. In addition, efforts will also be made to establish rules governing the use of work products by multiple divisions, and to popularize the MBD process.

References

(1) Shasai sofuto kyodaikani tachimukau, Nikkei

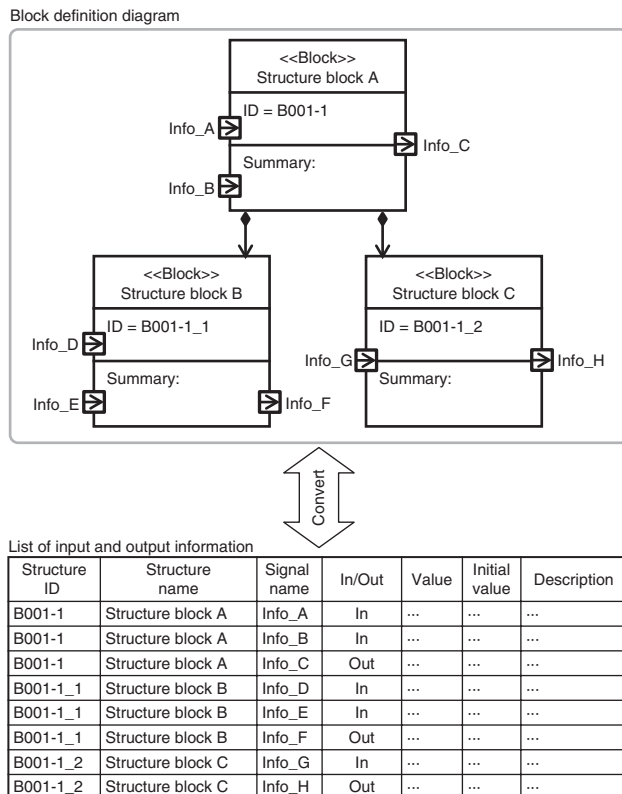


Fig. 14 Synchronous tool of input and output information

- Automotive Technology, 2007-11, Nikkei Business Publications, p. 82-83 (2007) (in Japanese)
- (2) <http://sec.ipa.go.jp/reports/20120323.html>, 2012/8/23
- (3) Mizutani, M., Yahata, Y., Kamei, Y., Hayase, K., Kaneda, M.: Application of model-based development to body-system mass-produced ECU, Mitsubishi Motors Technical Review, No. 22, p. 56-59 (2010)
- (4) Watanabe, S., Yoshino, H., Shionoya, Y., Seki, T.: New MBD Software Development Process for Electronic Control AWD System, Honda R&D Technical Review, Vol. 23, No. 1, p. 23-29
- (5) <http://www.omg.sysml.org/>
- (6) <http://www.mathworks.co.jp/products/>

■ Author ■



Noboru UJIGAWA



Shuhei KINOSHITA



Masayuki YAMAZAKI