

パワートレインシステムの複雑化に対応する モデルベース開発技術

Model-Based Development Technologies to Support Increasingly Complex Power Train Systems

荒 木 秀 和*

Hidekazu ARAKI

船 越 允 維*

Masayuki FUNAKOSHI

齋 藤 佳奈子*

Kanako SAITO

山 本 健 介*

Kensuke YAMAMOTO

大 口 良 輔*

Ryosuke OHGUCHI

要 旨

パワートレインシステムの複雑化に対応し、開発効率および品質を向上させるためには、要求と検証項目の明確化、要求の構造化とインタフェースの明確化、要求のトレーサビリティ確保が有用である。

本研究では、準形式表記である Systems Modeling Language を用い、独自にガイドラインを策定することで、この三つの項目を達成した。

また、Systems Modeling Language で記載した要求モデルは、パワートレイン電子制御ユニットのプラットフォーム領域に対する要求仕様としてサプライヤにも発行する。このため、要求の妥当性だけでなく、要求仕様として記載に問題ないかを効率的かつ精度良く検証するためのタスクを定義し、2015 年モデル量産機種開発に部分適用した。

ABSTRACT

In order to support power train systems as they become increasingly complex, as well as to raise the efficiency and quality of development, it is useful to (1) clearly define requirements and verification items, (2) structure requirements and clearly define interfaces, and (3) assure the traceability of requirements.

The present research project used Systems Modeling Language, which is a semi-formal notation. By formulating the project's own original guidelines, the above three items were achieved.

Requirement models written in Systems Modeling Language are also published to suppliers as requirement specifications for the platform area of the power train electronic control unit. Therefore the tasks involved in efficiently and accurately verifying not only the validity of the requirements, but also whether or not problems exist in how they are written as requirement specifications, were defined and applied in parts to the development of 2015 model year mass-production models.

1. はじめに

近年、自動車のパワートレイン制御システムは、大規模化、複雑化の傾向にある。開発効率向上と品質確保の両立に向け、自動車業界全体で以下の取り組みがなされている。

一つ目は、Fig. 1 に示すパワートレイン電子制御ユニット (PT-ECU) 開発プロセスの仕様設計段階で制御仕様の妥当性を確認してから、実装工程に移行することにより、

手戻りの小さい制御開発を実現する取り組みである。仕様設計段階で制御仕様の妥当性を確認するには、制御要求を Fig. 2 のように階層化し、各階層の制御要求が妥当かを検証する必要がある。

二つ目は、仕様設計段階で制御仕様の妥当性を確認するため、シミュレーションを活用する取り組みである。ソフトウェア作成前の仕様設計段階でシミュレーションを実施するには、制御対象と制御アルゴリズムをモデル化する必要がある。また、制御アルゴリズムがどの要求と紐づいて

*四輪 R&D センター

おり、どの検証項目を満たせば妥当と判定できるのかという、要求、検証項目、制御アルゴリズム間のトレーサビリティ確保が必須となる。

三つ目は、自動車向けソフトウェアの標準規格 Automotive Open System Architecture (AUTOSAR)⁽¹⁾に準拠した形で、制御ソフトウェアを設計する取り組みである。AUTOSARでは、Fig. 3に示すソフトウェア構造とともに、構造間のインタフェース (I/F) が規定されている。規格に準拠した形でソフトウェア構造や構造間 I/F を設計することで、制御ソフトウェアの再利用性を高めることができる。しかし、弊社の既存ソフトウェア構造と AUTOSAR のソフトウェア構造には差がある。そのため、PT-ECU サプライヤが AUTOSAR に準拠した形で

ソフトウェア実装する場合、弊社のソフトウェア構造と AUTOSAR 規格の適合で多くの時間が掛かるという課題があった。この対策として、AUTOSAR 規格を考慮した形で要求を構造化し、構造間 I/F を決定する、という方法が有効と考えた。

本研究では上記の取り組みを踏まえ、2016 年モデル以降の量産機種開発で制御ソフトウェアの再利用性を高め、手戻りの小さい制御開発を実現するために、下記の手法の構築を検討した。

- (1) Systems Modeling Language (SysML) を用いた、制御要求の階層化手法と妥当性検証手法
- (2) Model in the Loop Simulation (MILS) による要求の妥当性検証手法
- (3) AUTOSAR 規格を考慮したソフトウェア構造化手法

2. SysML を用いた制御要求の階層化手法と妥当性検証手法の構築

システムを実現する際に効果を発揮するモデリング言語である SysML は、システムの仕様化、分析、設計、妥当性確認や検証に利用できるため、現在、自動車、航空宇宙、通信分野などで幅広く活用されている⁽²⁾。

本研究では、Fig. 4 のプラットフォーム領域に含まれる制御要求に対し、SysML を用いた制御要求の階層化手法と、制御要求の妥当性検証手法を構築した。

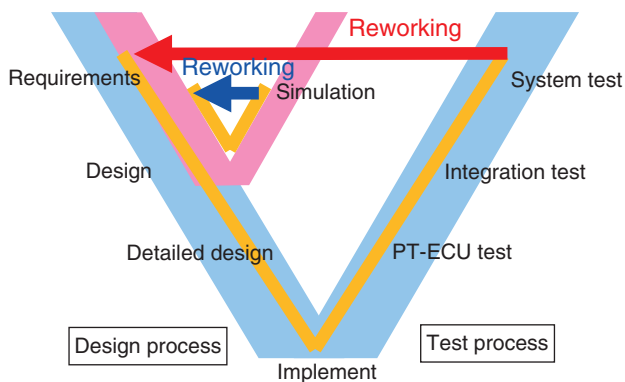


Fig. 1 V-model

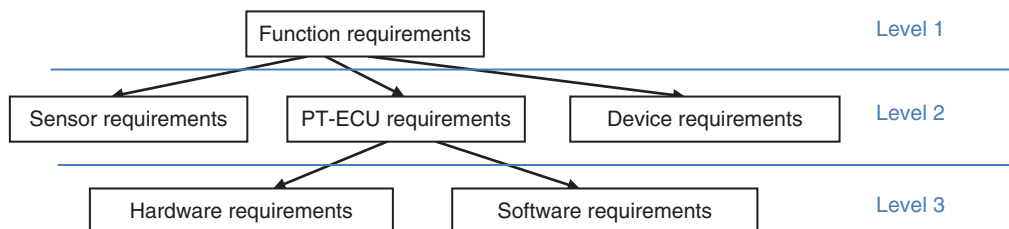
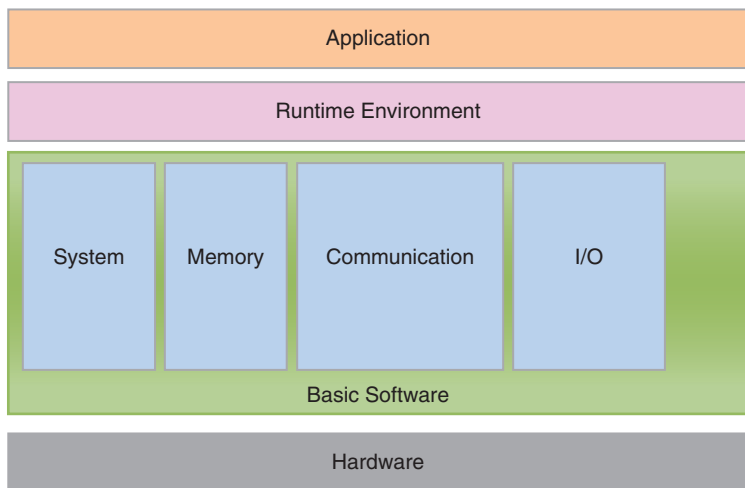


Fig. 2 Hierarchy of control requirements



I/O: Input Output

Fig. 3 AUTOSAR software architecture

2.1. SysML を用いた制御要求の階層化手法

Fig. 2の階層化をおこなうにあたり、各階層を下記のとおり定義した。

階層1は、実現する機能本来の制御要求、および、その機能と他機能との間のI/Fが明確となるレベルである。PT-ECU、センサ、デバイスといった構成要素が見直されても、本階層の制御要求、およびI/Fは変わらない。

階層2は、どういうPT-ECU、センサ、デバイスを使ってどう検知、操作するかが明確となるレベルである。本階層で、PT-ECUで実現すべき制御要求、センサで実現すべき制御要求、デバイスで実現すべき制御要求が明確となる。また、PT-ECUとセンサ、デバイス間のI/Fも本階層で明確化される。

階層3は、ソフトウェアで実現する制御要求とハードウェアで実現する制御要求、および、ソフトウェアとハードウェアでやり取りされるデータが、全て確定するレベルである。

各階層の構造は検証のしやすさを考慮し、出力信号数が最小となるよう定める。例えば、Fig. 5に示す構造Aは四つの入力を持っている。各入力がON, OFFで表される場合、テストパターンは $2^4=16$ とおり存在する。構造Aを構造A-1と構造A-2に分割できれば、テストパターンは $2^2 \times 2=8$ とおりまで削減でき、検証が容易となる。

各構造に割り当てられる制御要求には、構造の動作要求を自然言語で記述する。制御要求を網羅的に抽出できるよう、Table 1に示す七つのカテゴリタイプを定めた。

Table 1 Category types

No.	Category type
1	Operation
2	Safety
3	Market value
4	Regulation
5	Design policy
6	Diagnosis
7	Quality

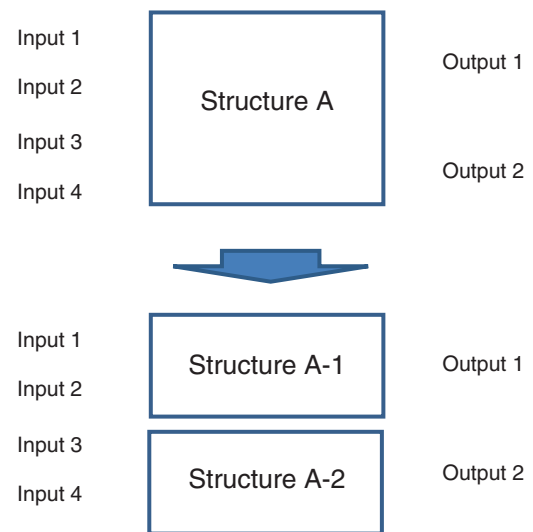


Fig. 5 Division of structure

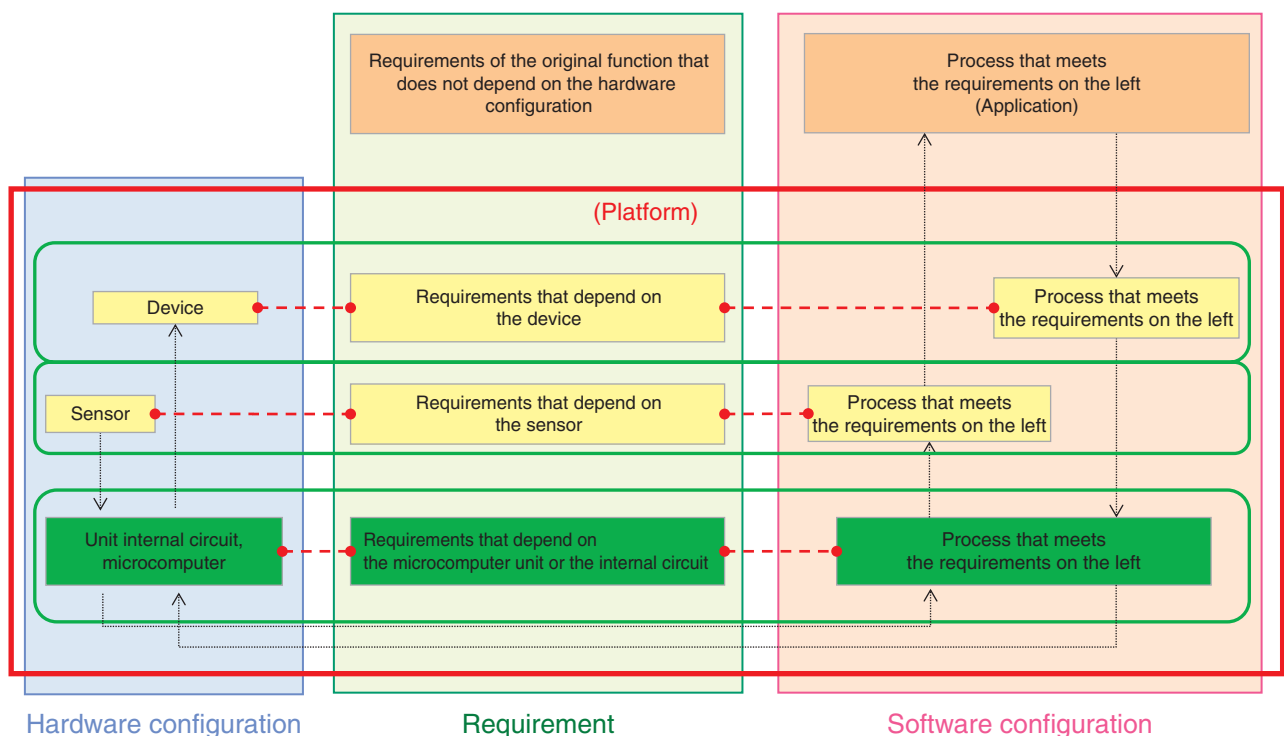


Fig. 4 Platform area

Table 2 Listed items of verification items

No.	Item	Provisions
1	Preconditions of tests	Required test conditions for verification
2	Parameters for test condition	Parameters and their values that correspond to the required test conditions for verification
3	Pre-state	State of the function before testing
4	Operating parameters and volume	Parameters that trigger transition to post-state, and the volume to operate
5	Post-state	State of function after testing
6	Measurement parameters	Measurement parameters that are subject to verification
7	Judgment criteria	Expected value of measurement parameters

2.2. 制御要求の妥当性検証手法

制御要求が満たされているかどうかの判定基準を明確化するため、各制御要求に対し、検証項目を一つ以上作成する。検証項目は、仕様設計段階で MILS や試作機による制御仕様の妥当性検証をおこなう際に判定基準として使用されるほか、実機テスト段階で一致性検証の判定基準としても使用される。将来の検証自動化を想定し、検証項目作成について Table 2 に示す七つの記載項目を定めた。

以上を踏まえ、各階層につき 2.2.1. 項から 2.2.4. 項に示す Diagram を作成する。

2.2.1. Criterionl Requirement Diagram 作成

Fig. 6 に示すとおり、Criterionl Requirement Diagram (CRD) では、構造、制御要求、検証項目と、それぞれの紐づけを記載する。

2.2.2. Block Diagram 作成

Fig. 7 に示すとおり、Block Diagram (BD) では、その階

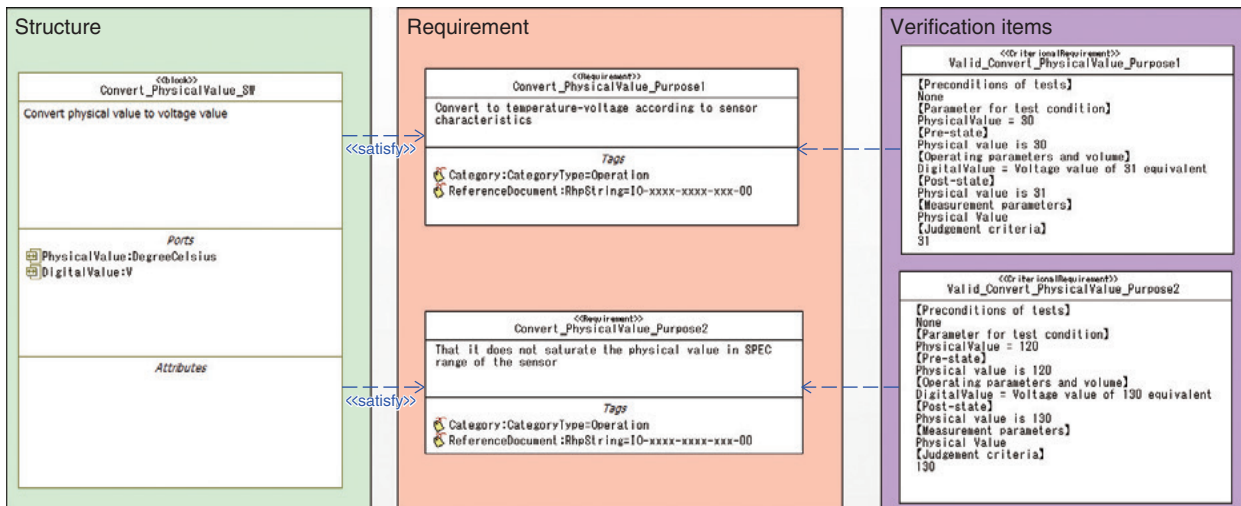


Fig. 6 Criterionl requirement diagram

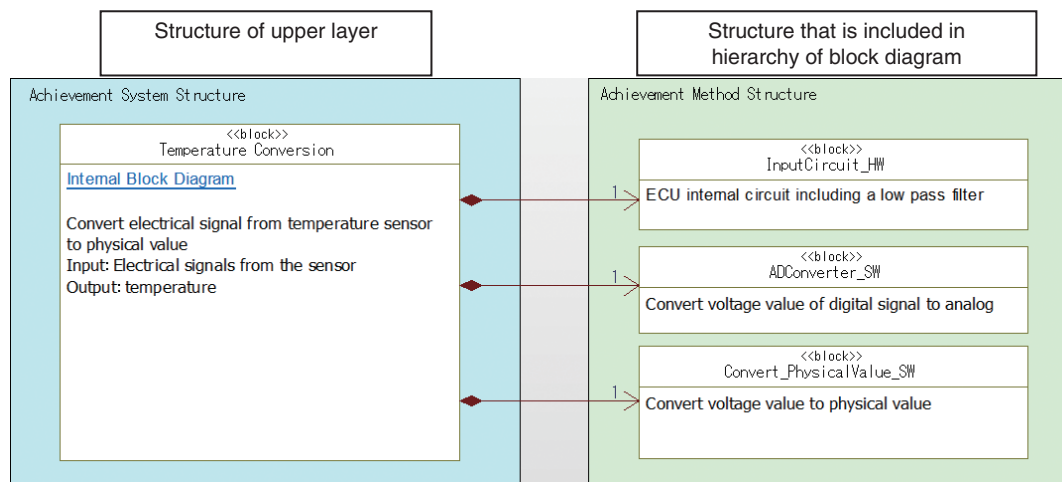


Fig. 7 Block diagram

層に含まれる構造と上位構造との紐づけを記載する。よって、最上位階層である階層1ではBDを作成しない。

2.2.3. Internal Block Diagram 作成

Fig. 8に示すとおり、Internal Block Diagram (IBD) では、下位構造間のデータの流れを記載する。

これにより、仕様変更に伴う影響分析、安全分析において、機能内部で完結する変数なのか、他機能とのI/Fなのかの判別が容易となる。

2.2.4. Requirement Diagram 作成

Fig. 9に示すとおり、Requirement Diagram (RD) では、その階層に含まれる制御要求と上位階層の制御要求との紐づけを記載する。よって、最上位階層である階層1ではRDを作成しない。

階層化の定義、要求抽出のためのカテゴリタイプ、検証項目作成時の記載項目、およびDiagramの記述方法を定め、階層間の要求の紐づけ、階層間の構造の紐づけ、各階層の構造と要求と検証項目の紐づけを明確化することにより、制御要求の妥当性検証が可能となる。

本研究では、SysML記述内容をシステム設計要求仕様とすることにより、制御要求の階層化と妥当性検証を2016年モデル量産機種に適用できるよう、2015年モデル量産機種への先行適用で、課題の抽出と対策を実施した。

3. MILS による要求の妥当性検証手法

開発の手戻りをなくし、開発効率を向上させるためには、PT-ECUのハードウェア開発やソフトウェア開発の前に制御要求の妥当性を検証し、要求の精度を高める必要がある。その手法として、SimulinkモデルでのMILSによる動作検証を実施する。

3.1. プラットフォーム検証用 Simulink モデルの作成

プラットフォームの実機動作検証は、センサからの入力と、デバイスへの出力を用いてPT-ECUの動作を検証するために、Fig. 10に示す構成で実施する。

MILSとは、実際のセンサ、デバイスをモデル化し、机上で電子制御ユニットのソフトウェア動作を模擬する検証手法である。

本研究では、プラットフォームのMILSを実施するにあたり、Fig. 11に示すSimulinkモデルを作成した。

図中のプラントモデルは、センサ、デバイス等に対応する。コントローラモデルは、制御要求に対応する。テストモデルは、他システムからの入力部に対応する。Input Output (I/O) モデルは、ECUの入出力回路に対応する。

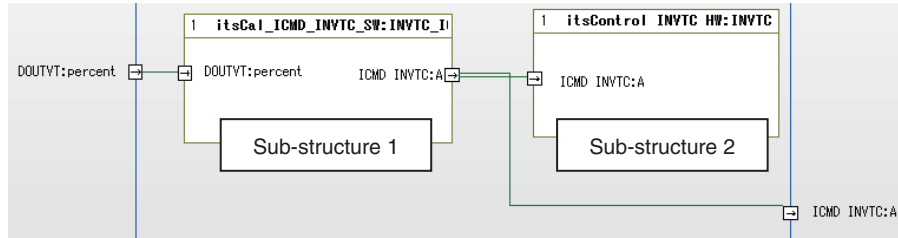


Fig. 8 Internal block diagram

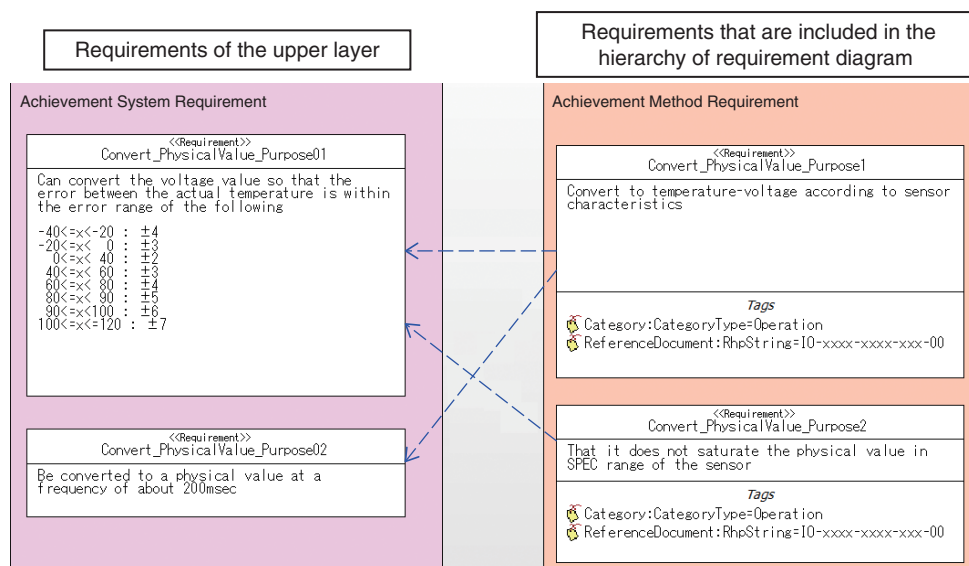


Fig. 9 Requirement diagram

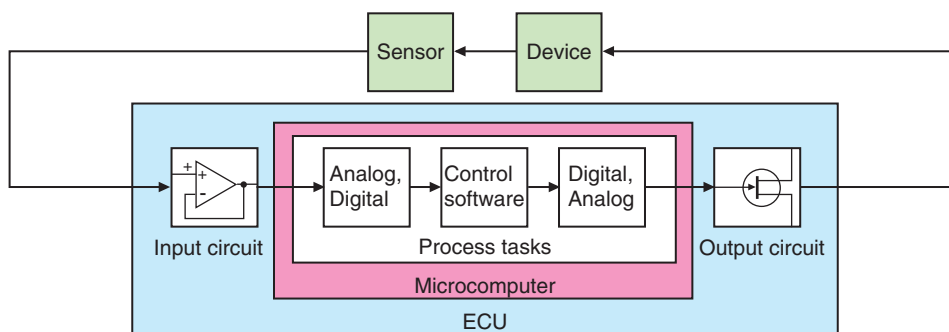


Fig. 10 Real-world environment

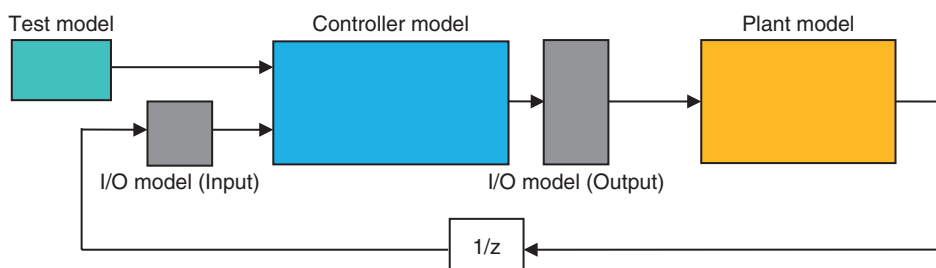


Fig. 11 Virtual environment

3.2. テストパターンの作成と検証

テスト対象であるコントローラモデルは、センサ故障検知部、デバイス故障検知部、センサコントロール部、デバイスコントロール部で構成されており、各部について、検証が必要である。プラットフォームの検証は、基本動作検証と故障検知検証を2本立てで実施する。

- (1) 基本動作検証では、他システムからの入力値（目標値）を Fig. 12 のように動的に変更させ、入力値に対する PT-ECU 出力部の出力値が、検証項目の記載に合致する妥当な値であるかを検証する。
- (2) 故障検知検証では、センサ、デバイスの断線、地絡、天絡等の故障をモデル上で擬似的に発生させて、プラットフォームの故障状態検知ステータスが検証項目の記載に合致する妥当な値であるかを検証する。

Simulink モデルを用いた MILS により、プラットフォームのシステム機能単位で、要求仕様の妥当性検証ができる。

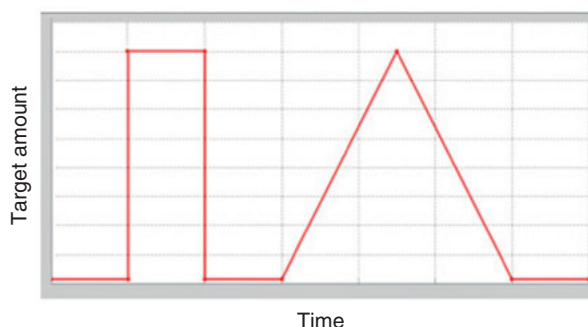


Fig. 12 Simulation test pattern

本研究では、2016 年モデル以降の新規デバイス開発に対して MILS による妥当性確認を適用できるよう、2015 年モデルの新規デバイス開発に先行適用し、MILS の有用性と課題を確認した。

4. AUTOSAR 規格を考慮したソフトウェア構造化手法

Fig. 3 に示す AUTOSAR 構造につき、各部の概要を以下に示す。

アプリケーションは、PT-ECU ハードウェア非依存のソフトウェアであり、自動車メーカーの競争領域と位置づけられる。

Runtime Environment は、センサ、デバイス、PT-ECU 内部回路、マイコンといったハードウェア構成が変わっても、アプリケーション間やアプリケーションと Basic Software (BSW) の間の I/F が変わらないよう、接続を抽象化するソフトウェアである。

BSW は、マイコンや PT-ECU 内部回路の変更により、プログラムが変わるソフトウェアである。ハードウェア構成に合わせて抜き差しすることにより再利用性を高められるよう、内部機能モジュールとして構成される。

4.1. プラットフォーム機能の分類

弊社の既存ソフトウェア構造と AUTOSAR 規格の適合で多くの時間が掛かるという課題をクリアするため、プラットフォーム機能を、Fig. 13 に示す 6 グループに分割する。

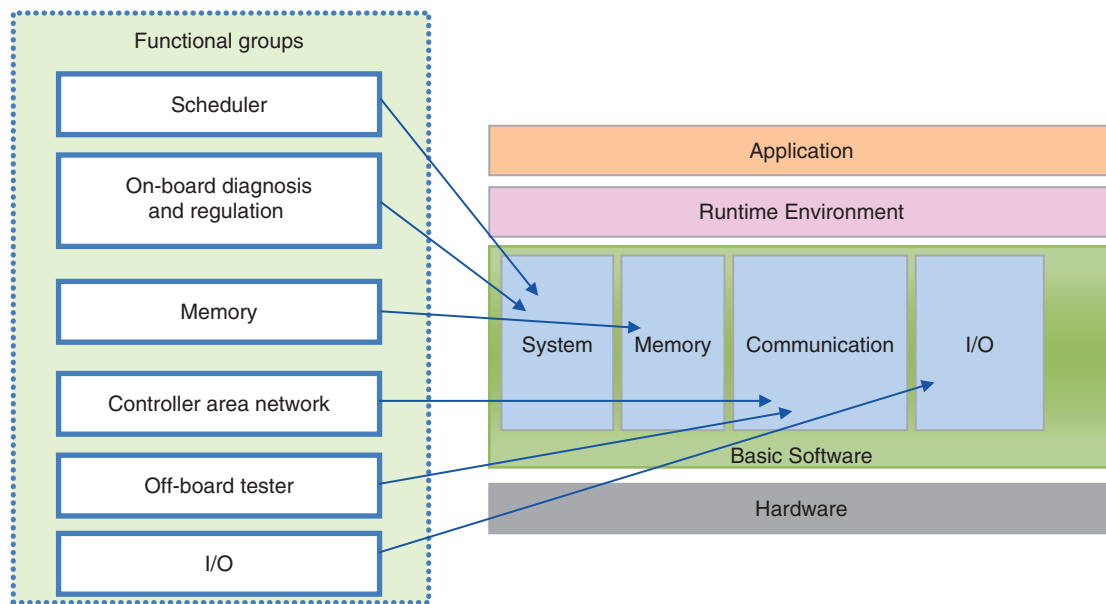


Fig. 13 Linkage of functional groups and AUTOSAR structure

各グループの概要を以下に示す。

図中の Scheduler は、PT-ECU のタスク制御をおこなう。On board diagnosis and regulation は、車載コンピュータによる自己診断と法規対応データの生成をおこなう。Memory は、不揮発データ管理をおこなう。Controller area network は他ユニットとの送受信をおこなう。Off board tester は、外部診断機との送受信をおこなう。I/O は電子制御ユニットの入出力をおこなう。

4.2. AUTOSAR 規格を考慮した I/F

各グループに含まれるプラットフォーム機能のソフトウェア仕様作成にあたり、アプリケーションに公開する I/F の意味を、各グループに対応する AUTOSAR の機能モジュールがアプリケーションに提供する I/F と一致させる。また I/F 名を、自動車業界で標準的に使用される略語の組み合わせで生成する。

4.1. 節と 4.2. 節の取り組みにより、ハードウェア構成に合わせたソフトウェアの抜き差しが容易となるだけでなく、PT-ECU サプライヤが AUTOSAR にのっとってプラットフォームのソフトウェアを実装する場合において、アプリケーションとの I/F 適合作業の負担を軽減することができるなど、ソフトウェアの再利用性を高めることができる。

本研究では、2016 年モデル量産機種でプラットフォーム機能の全ソフトウェア構造を新構造に置き換えられるよう、新構造に対応したソフトウェア仕様を作成し、2015 年モデル量産機種に部分適用した。

5. 2015 年モデル量産機種開発への適用

5.1. SysML を用いた制御要求の階層化と妥当性検証

SysML を用いた制御要求の階層化と妥当性検証を量産機種開発に適用するため、2.1. 節で記載した階層の定義、および、2.2. 節で記載した Diagram 作成方法についてガイドラインを作成した。

また、SysML は、システム設計レベルで情報を関連付ける、トレーサビリティを確保するという目的に対して高い優位性を発揮するが、類似した情報が分散してしまい、特定の種類の情報だけを抽出することが容易でないという性質がある。これにより、検証で情報の過不足を効率的に確認できないという課題が判明したため、システム設計要求仕様の検証実施前に必要なタスクを定義し、検証の効率化を図った。

5.1.1. 検証提示資料

検証対象情報を効率的に表示するため、仕様作成者が事前に HTML レポートを作成する。HTML レポートには、検証に必要な図、要求項目、構成要素、検証項目等が抽出、記載される。

HTML レポートは、検証に必要な情報、またその表示方法を規定するテンプレートを事前に作成しておくことで、誰でも同じフォーマットで自動作成することができる。また、HTML レポートでは関連情報をハイパーリンクでつなぐことも可能であり、検証をより効率的に進めることができる。

HTML レポートの応用例として、Fig. 14 に示す I/F レポートが挙げられる。I/F レポートでは、各設計 Block の入出力ポートをレベル別に表形式で表示する。この表の利用により、SysML 形式では同時に表示できない複数 Block の入出力を、網羅的に検証することができる。

この HTML レポートを要求仕様検証で使用することにより、検証者への要求仕様提示に割かれる時間が削減され、効率的な検証が可能となった。

Interface of Platform_IAT11

This table is generated from L3 flowports

Name	In/Out	Description	Unit	OwnerBlock
IntkAirT1B1_st	In	IAT11 status information	Integer	FI-control
IntkAir_t1B1	In	Engine intake air temperature (IAT11)	DegreeCelsius	FI-control
IAT11AD	Out	IAT11 sensor input voltage value after the AD conversion	V	IAT11_calc
IntkAirT1B1_st	Out	IAT11 status information	Integer	IAT11_calc
IntkAir_t1B1	Out	Engine intake air temperature (IAT11)	DegreeCelsius	IAT11_calc

This table is generated from L4-1 flowports

Name	In/Out	Description	Unit	OwnerBlock
IntkAir_t1B1	In	Engine intake air temperature (IAT11)	DegreeCelsius	IAT11_DetectRangeFail
IAT11AD	Out	IAT11 sensor input voltage value after the AD conversion	V	IAT11_Conversion
IntkAir_t1B1	Out	Engine intake air temperature (IAT11)	DegreeCelsius	IAT11_Conversion
Obd10D_st	Out	IAT11 OutOfRange status information	Integer	IAT11_OBD_Common
Obd10_st	Out	IAT11 status information	Integer	IAT11_OBD_Common

This table is generated from L4-2 flowports

Name	In/Out	Description	Unit	OwnerBlock
IntkAir_t1B1	In	Engine intake air temperature (IAT11)	DegreeCelsius	OutOfRange_Lo_Detection_SW
IntkAir_t1B1	In	Engine intake air temperature (IAT11)	DegreeCelsius	OutOfRange_Hi_Detection_SW
IAT11AD	Out	IAT11 sensor input voltage value after the AD conversion	V	IAT11_ADC_SW
IntkAir_t1B1	Out	Engine intake air temperature (IAT11)	DegreeCelsius	IAT11_PhyConv_SW
Obd10DH_st	Out	IAT11 OutOfRange High status information	Integer	OBD_Common_SW
Obd10DL_st	Out	IAT11 OutOfRange Low status information	Integer	OBD_Common_SW
Obd10H_st	Out	IAT11 Limit High status information	Integer	OBD_Common_SW
Obd10L_st	Out	IAT11 Limit Low status information	Integer	OBD_Common_SW

Fig. 14 Interface (I/F) report

5.1.2. 要求仕様のスタイルチェック

要求仕様品質向上のため、策定した要求分析ガイドラインに準拠した書式やフォーマットで要求仕様が作成されているかを検証する必要がある。しかし、人による目視確認では抜け漏れが発生する可能性がある上、莫大な検証時間が必要となる。そこで、独自に開発したツールを使い、仕様作成者が事前に要求仕様の書式やフォーマットを簡易チェックし、その結果 (Fig. 15, Fig. 16) を提示する運用とした。検証前にスタイルチェックをおこなうことで、設計上想定していない関連付けや矛盾を取り除いた要求仕様書で検証を実施することができる。これにより要求仕様内容の妥当性チェックに重点を置くことが可能となった。

上記の課題抽出と対策により、SysML を用いた制御要求の階層化と妥当性検証を 2016 年モデル量産機種に適用できる見通しが得られた。

5.2. MILS による要求の妥当性検証

MILS による要求の妥当性検証手法をおこなうため、3.1. 節に示すプラットフォーム検証用 Simulink モデルの作成、および、3.2. 節に示すテストパターンの作成と検証について、ガイドラインを作成した。

また、2015 年モデル量産車に適用予定の新規デバイス開発に対し、MILS による要求の妥当性検証を実施し、下記の知見を得た。

(1) PT-ECU のハードウェア、ソフトウェア、および実デ

バイスがない開発序盤において、制御周期を仮決定する、どのような異常動作が起こり得るかを見極める、といった用途で効果を発揮する。

(2) 検証の確からしさは、MILS で使用したプラントモデルの作りこみに依存する。このため、現状、制御周期の本決定や、異常判定等のしきい値決定には、実機での検証が必須である。MILS と実機でそれぞれ、どのような検証をおこなうべきかについては、今後も引き続き検討していく必要がある。

本取り組みにより、MILS の有用性と課題が明らかとなり、2016 年モデル以降の新規デバイス開発に対し、MILS による妥当性確認を適用できる見通しが得られた。

5.3. AUTOSAR 規格を考慮したソフトウェア構造化

4 章に示す機能分割方針、および I/F 決定方針に従い、On board diagnosis and regulation, Memory, Controller area network, Off board tester の 4 グループにつき、新構造に対応するソフトウェア仕様を作成した。

また、Scheduler と I/O の 2 グループについても、従来から機能要求が変わる部分について、新構造に対応するソフトウェア仕様を作成した。

上記の新構造対応ソフトウェア仕様を 2015 年モデルの量産機種に適用することにより、プラットフォーム機能の全グループについて、2016 年モデル量産機種で新構造に置き換えられる見通しが得られた。

Project	File	Location	D:\Platform\Platform_Brake.rpy
Result	No	Check point	Result details
OK	1-8	Does the model have a comment named "Diagram" in the package "Index".	
Failure	1-8-1	Does the model have hyperlinks to all diagrams except internal block diagram in the description of the comment named "Diagram".	Hyperlink to diagram does not exist L4-1::L4-1_CRD_BrakeNoSwDet_ECU L4-1::L4-1_BD L4-1::L4-1_RD_Diagnosis L4-1::L4-1_RD_Operation L4-1::L4-1_CRD_BrakeNoSwDet_ECU L4-1::L4-1_CRD_BrakeNoSwErrorDet_ECU L4-1::L4-1_CRD_BrakeNoSwErrorDet_ECU L3::L3_CRD_BrakeStDet
CHK	2	Does the model have a block.	block exist.
OK	2-1	Are the names of blocks in the model all different.	
OK	2-2	Do the names of blocks in the model all start with capital letters. Also, does the model use only alphanumeric text and the underscore character.	

Fig. 15 Style check sheet

[illegible]

Fig. 16 Requirement-block connection check

6. まとめ

下記手法を構築し、2015年モデル量産機種開発に適用することで、今後、手戻りの小さい制御開発を実現するための仕組みを整えることができた。

- (1) SysML 形式を用いた制御要求の階層化手法と、制御要求の妥当性検証手法
- (2) 仕様設計段階での、Simulink モデルを用いた MILS による妥当性確認手法
- (3) AUTOSAR を考慮した、プラットフォームのソフトウェアアーキテクチャ構造化手法

参考文献

- (1) <http://www.autosar.org/>
- (2) 独立行政法人 情報処理推進機構：序文. システムエンジニアリングについて，モデルベースシステムズエンジニアリング導入の手引き，p.13-36 (2013)

■ 著 者 ■



荒木秀和



齋藤佳奈子



大口良輔



船越允維



山本健介