

振舞いプロパティに基づく自動車組込みソフトウェアの 協調制御アーキテクチャ設計方法の提案と評価

青山 幹雄^{1,a)} 田邊 隼希²

受付日 2011年6月14日, 採録日 2011年11月7日

概要: 自動車組込みソフトウェアは多数の ECU (Electronic Control Unit) がネットワークを介して連携するリアルタイム分散処理アーキテクチャをとる. 高度な安全性や信頼性を実現するために, 複数の ECU 間で協調制御が必要となっている. しかし, ECU 間のネットワークを介した情報の相互作用に加え, ECU 制御に対する車体運動の反応を多数のセンサが取り込むことによる物理的相互作用が加わり, 協調制御の設計が困難である. 本稿は, 複数 ECU 間のグローバルな振舞いをプロパティとして统一的にモデル化し, プロパティを中心とする自動車組込みソフトウェアの協調制御アーキテクチャ設計方法を提案する. プロパティは自動車, 周辺環境, ユーザの特性とその状態を包括的に定義する. DSM (Design Structure Matrix) を拡張し, プロパティを非機能要求と関係づけて自動車全体の振舞いをモデル化する方法を提案する. モデルに基づくアーキテクチャ設計により, 非機能要求を満たす自動車全体の協調制御設計が可能となる. 提案設計方法を車両運動統合制御システムに適用し, その有効性を示す.

キーワード: ソフトウェアアーキテクチャ, 自動車組込みソフトウェア, リアルタイム分散処理, サイバーフィジカルシステム, フィーチャ競合

A Design Methodology and Its Evaluation for Automotive Cooperative Software Architecture Based on the Behavioral Properties

MIKIO AOYAMA^{1,a)} HAYAKI TANABE²

Received: June 14, 2011, Accepted: November 7, 2011

Abstract: It is required to cooperatively control automotive embedded software across a number of ECUs (Electronic Control Units) connected through networks. However, the complexity of interactions among ECUs makes it difficult to design the cooperative control. This article proposes a design methodology of architecture for automotive cooperative software based on the behavioral properties. We define the property as the attributes and states of the vehicles, surrounding environment and users. We propose the extended DSM (Design Structure Matrix) and associated models for modeling automotive behavior involving NFRs (Non-Functional Requirements). By designing the architecture based on the set of models, we can design the cooperative control across the distributed automotive software systems for meeting the NFRs. We apply the proposed methodology to the VDM (Vehicle Dynamics Management system) and evaluate the effectiveness of it.

Keywords: Software Architecture, Automotive Embedded Software, Real-Time Distributed Processing, Cyber Physical System (CPS), Design Structure Matrix (DSM)

¹ 南山大学情報理工学部ソフトウェア工学科
Department of Software Engineering, Nanzan University,
Seto, Aichi 489-0863, Japan

² 南山大学大学院数理情報研究科
Graduate School of Mathematical Sciences and Information
Engineering, Nanzan University, Seto, Aichi 489-0863,
Japan

^{a)} mikio.aoyama@nifty.com

1. はじめに

自動車組込みソフトウェアは多数の ECU (Electronic Control Unit) がネットワークを介して連携するリアルタイム分散処理アーキテクチャをとる. 安全で快適な走行,

排気ガスや燃料消費の削減などの環境性能の向上など、多様な要求を満たすために複数の ECU 間で協調制御が必要となっている [3]. しかし、ECU 間の相互作用はネットワークを介した情動的相互作用に加え、ECU 制御に対する車体運動の反応を多数のセンサが取り込むことによる物理的相互作用が加わり、きわめて複雑であるため、協調制御の設計は困難である。このため、異なる ECU が制御するシステム間で機能競合 (Feature Interaction) が発生し、不具合の原因ともなっている [11]. このようなシステムの設計は、物理コンピューティングシステム (Physical Computing System) [21], あるいは、CPS (Cyber Physical System) [13] とも呼ばれる新たな問題領域の 1 つであることから、新たなアプローチが求められている。

本稿では、複数 ECU 間にまたがる自動車全体のグローバルな振舞いに着目し、これをプロパティとして統一的にモデル化し、プロパティを中心とする自動車組込みソフトウェアの協調制御アーキテクチャ設計方法を提案する。プロパティは自動車、環境、ユーザーの特性とその状態を包括的に定義する。DSM (Design Structure Matrix) [4] を拡張し、プロパティと非機能要求と関係づけて自動車全体の振舞いをモデル化する方法を提案する。モデルに基づくアーキテクチャ設計により、非機能要求を満たす自動車全体の協調制御設計が可能となる。

2. 自動車組込みソフトウェアの課題

自動車組込みソフトウェアは、エンジンやブレーキなどの個別制御からそれらが協調した自動車全体の振舞い制御へと進化している。このため、自動車組込みソフトウェア開発には次のような新たな課題が提起されている。

(1) 自動車組込みソフトウェアの協調制御の振舞いとアーキテクチャ設計の複雑化

自動車組込みソフトウェアは多様である。エンジンやブレーキなどの制御ソフトウェア開発では、従来、機器ごとの個別制御を対象としてきた。これらの制御ソフトウェアの設計では、制御対象システムごとにフィードバック制御モデルを構築し、それに基づくモデルベース開発が中心である。この設計を支援する、MATLAB/Simulink などのツールが広く利用されている [17]. 一方、カーナビゲーションシステムなどの開発では、一般の組込みソフトウェアと同様、状態遷移に基づく設計が行われている。

現在の自動車組込みソフトウェアは、安全性の向上や排気ガス削減、燃費などの環境性能向上などの多様な要求を満たすために、エンジン、ブレーキ、トランスミッションなどの制御とカーナビゲーションシステムなどの異なる性質を持つ複数のシステムが協調する必要があるため、ネットワークを介した協調制御が導入されている [9], [20]. しかし、従来の個別に設計された制御システムを協調制御することは、異なるシステム間の振舞いの相互作用を制御する必要があ

る。このようなシステムは SoS (System-of-Systems) [10] の一種でもあるが、異種分散処理システムであることから、システム全体の振舞いを表現するモデルが確立しておらず、機能競合 (Feature Interaction) による不具合の発生原因ともなっている [11]. さらに、ECU 間の相互作用はネットワークを介した情動的相互作用に加え、ECU 制御に対する車体運動の反応を多数のセンサが取り込むことによる物理的相互作用が加わり、きわめて複雑である。

このため、従来のコンポーネント・コネクタに基づく構造的アーキテクチャモデルでは自動車組込みソフトウェアの協調制御アーキテクチャの設計が困難である。振舞いの相互作用に着目した新たなモデルとそれに基づくアーキテクチャ設計方法が必要である。

特に、従来は個々の ECU 内でフィードバック制御システムとして閉じたシステムとして扱ってきたが、現在は ECU 間の情動的かつ物理的協調が重要な要因となっていることから、他の ECU やセンサからの相互作用を取り込むことができる開いたシステムとして扱う必要がある。さらに、ECU 間の相互作用は、協調として意図された作用に加え、自動車の走行にともない物理的実体として相互に及ぼす物理的力や路面を介した作用などの意図されない相互作用が無視できない影響を及ぼす点で、従来のアーキテクチャ設計技術では扱うことが困難となる新たな課題を提示している。

(2) 自動車組込みソフトウェアの非機能要求設計の複雑化

自動車組込みソフトウェアは、安全性、信頼性などの非機能要求の厳格な達成が求められる。非機能要求はシステムの横断的特性であることからグローバルな性質の設計が必要となるため、自動車の協調制御ソフトウェアのようなリアルタイム分散処理システムでは SoS 全体の複雑な振舞いに対する非機能要求を満たす設計がきわめて困難となる。従来の、たとえば、NFR フレームワーク [8] などの非機能要求のモデル化方法は、このような分散処理システム全体の非機能要求を、複雑なシステムの振舞いとあわせてスケラブル、かつ、厳密にモデル化できるまでには至っていない。振舞いと非機能要求との関係をモデル化できる新たなモデル化方法とそれに基づく設計方法が求められる。

3. 関連研究

本稿に関連する研究を次の 4 つのアプローチにまとめて示す。

(1) ソフトウェアアーキテクチャのモデル化

従来の構造的アーキテクチャモデルに対して、異なる視点から補完するモデルが提案されている。インバリエントに基づくモデル [1] やコンサーンに関連づけるモデル [12] がある。

一方、自動車組込みソフトウェアのドメインでは、アーキテクチャの複雑さを軽減するために、車種などに非依存

なモデル化 (DAF) が提案されている [5]。しかし、具体的な設計方法の提案までには至っていない。

(2) 自動車組込みソフトウェアの開発技術

自動車ソフトウェアの開発においては、エンジンやブレーキなどの制御システムを対象として、モデルベース開発が広く普及している [17]。しかし、制御システムと状態遷移に基づく情報システムが協調するシステム全体の振舞いを表現するモデルとはなっていない。また、ECU 間の情報的相互作用に加え物理的相互作用を介した多様な協調的振舞いを表現することが必要である。これは、物理コンピューティングシステム (Physical Computing System) [21]、あるいは、CPS (Cyber Physical System) [13] とも呼ばれる新たな問題領域の 1 つである。ECU 間の多様な相互作用を統一的にモデル化する方法が必要である。

(3) DSM によるアーキテクチャの設計と評価

システムの相互作用の一般的なモデルとして DSM (Design Structure Matrix) が知られている [4]。DSM を用いて類似度の高い要素をまとめてモジュール化し、要素間の依存関係を簡略化する [2], [16]。モジュール性の評価方法と支援ツールも提案されている [15]。自動車組込みソフトウェアの設計においても、モジュール設計とモジュール性の評価へ DSM を応用する方法が提案されている [12]。しかし、システム間の協調制御によるグローバルで複雑な振舞いの設計への応用には至っていない。

(4) 振舞いと非機能要求のモデル化とアーキテクチャ設計

NFR フレームワーク [8] などの非機能要求を構造的にモデル化する方法が提案されている。非機能要求はシステムの横断的な振舞いと密接に関係するが、複数の制御システム間での振舞いの相互作用と関連づける非機能要求のモデル化、ならびに、それに基づくアーキテクチャ設計は確立されているとはいえない。

4. プロパティ中心アーキテクチャ設計方法

4.1 プロパティとプロパティ中心設計

プロパティは自動車、周辺環境、ユーザーの特性や状態である。振舞いをプロパティの変化として定義する (図 1)。さらに、振舞いの相互作用をプロパティ間の相互作用とし

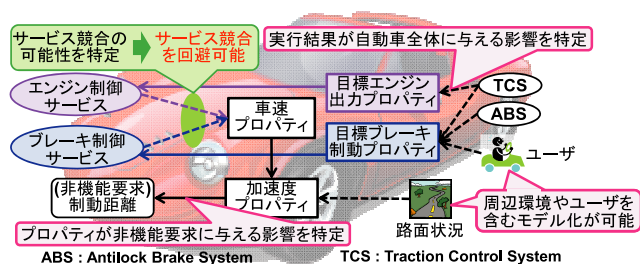


図 1 自動車の協調制御とプロパティによるモデル化

Fig. 1 Property model of cooperative control software for automotive.

て定義する。プロパティを制御する機能をサービスと呼ぶ。複数サービスがあるプロパティの一貫性を損なう変更をサービス競合と呼ぶ。

プロパティに基づく設計をプロパティ中心設計と呼ぶ。プロパティ中心設計ではサービスが影響を与えるプロパティを特定して設計する。プロパティの考え方は、オブジェクト指向における属性を分散処理ソフトウェア全体へ拡張したものといえる。

本稿では、システム間の相互作用に焦点を当てていることから、個別システム内の設計は、従来の設計方法が適用されるものとする。このような点から、プロパティは、従来のモデルを補完し、SoS (System-of Systems) や CPS のグローバルな性質をモデル化する技術として意義がある。たとえば、「ナビ協調」と呼ばれるサービスでは、カーナビゲーションシステムの地図情報に基づき、坂道の勾配に基づき、エンジンやトランスミッションの制御を行い燃費や走行性の向上を図っている。従来は、このようなエンジンなどの制御システムとカーナビゲーションシステムなどの情報システムとは疎結合であったが、現在は、カーナビゲーションシステム、さらには、テレマティクスサービスなどの豊富な情報を活用した高度な協調制御が必要となっている。また、ハイブリッド車、あるいは、電気自動車では、駆動システムと回生ブレーキなどをシームレスに制御するために、いっそう高度な協調制御が必要となっている [20]。

このように、設計対象の範囲に応じて、プロパティにはコンテキストとして扱われるべき情報も含まれることに留意願いたい。

4.2 プロパティの分類

自動車を例に、ユーザ、自動車、周辺環境の特性に着目してプロパティを分類した (図 2)。

- (1) ユーザプロパティ: ユーザの振舞いをその特性と状態によりモデル化する。ユーザプロパティは、ユーザの操作などの取りうる状態や値を持つ。
- (2) 自動車プロパティ: 自動車の特性とその状態を表す。自動車のサービスはアクチュエータの振舞いを表すプロパティを制御する。

プロパティは目標値と現在値の 2 つの属性を持つことが

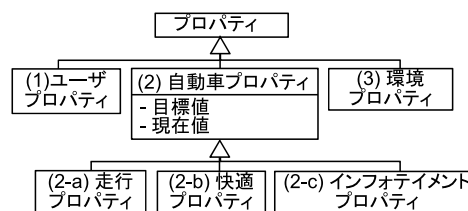


図 2 プロパティの分類

Fig. 2 Classification of behavioral properties.

できる。特に、自動車プロパティでは、制御システムの目標値への追従制御が必要となる。ユーザプロパティでは、ユーザ状態の表現のみのプロパティもあると考えており、このような場合、目標値は必要とされない。

サービスはプロパティの現在値が目標値となるように制御する。自動車プロパティは制御するアクチュエータの特性から、さらに、次のように分類できる。

- (2-a) 走行プロパティ：エンジンなどの自動車の走行に関するプロパティである。
 - (2-b) 快適プロパティ：エアコンなどの自動車の快適環境の状態に関するプロパティである。
 - (2-c) インフォテイメントプロパティ：カーナビゲーションシステムなどインフォテイメントに関するプロパティである。
- (3) 環境プロパティ：周辺環境の特性とその状態を示し、サービスが制御不可能なプロパティである。環境プロパティは外気温のようにセンサで測定可能なプロパティと、路面状況のようにセンサで測定不可能なプロパティがある。

4.3 プロパティのモデル化

自動車の協調制御の設計では、制御システムと情報システムとの協調をモデル化する必要がある。さらに、多様なセンサやアクチュエータが組込まれているため、それらを取り込む情報も多様となっている。これらの情報を表現するプロパティの種類や特性も異なる。

このようなプロパティ間の関係は、プロパティの型に基づく階層構造と協調制御によるプロパティ間での振舞いの影響伝播の2つの視点から、次のようにプロパティ構造モ

デルとプロパティ振舞いモデルでモデル化する。このプロパティモデルのメタモデルを図3に示す。

- (1) プロパティ構造モデル：プロパティとその間の構造的依存関係を階層構造化する。
- (2) プロパティ振舞いモデル：プロパティに基づき自動車全体のグローバルな振舞いをモデル化する。これは、次の振舞い設計表と依存関係モデルからなる。
 - 1) 振舞い設計表：プロパティとサービス、サービスと非機能要求間の影響をモデル化する。振舞い設計表はサービスからプロパティへの影響を表すサービス-プロパティ振舞い設計表 (S2P) とプロパティからサービスへの影響を表すプロパティ-サービス設計表 (P2S) に分けられる。
 - 2) 依存関係モデル：プロパティ間とプロパティと非機能要求間の振舞いの依存関係を拡張 DSM でモデル化する。

4.3.1 プロパティ構造モデル

プロパティ間とプロパティと非機能要求との関係をプロパティの型に基づき構造化し、クラス図で示す。センサやアクチュエータの構成の違いを抽象化して、プロパティの静的構造を理解するためのモデルである。図4に示す例では、具体的な仕様書に基づき抽出したプロパティを抽象化し、ある階層内での関係を示す。

4.3.2 拡張 DSM による依存関係モデル

従来、DSM [4] は要素を行、列にとり要素間の依存関係を示すために用いられている。

本稿では従来の DSM を拡張してプロパティ間の依存関係に加え、非機能要求とプロパティ間の依存関係を表現可能な拡張 DSM を提案する (図5)。

拡張 DSM は、プロパティ間の影響とあわせてプロパティと非機能要求間の影響を表現する。すなわち、サービス実行のプロパティへの影響に加えプロパティを介した非機能要求への影響も表現する。特に、サービスの実行によるプロパティの変化は他のプロパティにも伝搬し、伝播の連鎖を通して非機能要求へ影響する点に着目する。このプロパティ間の影響伝搬と非機能要求への影響を表現するモデルである。

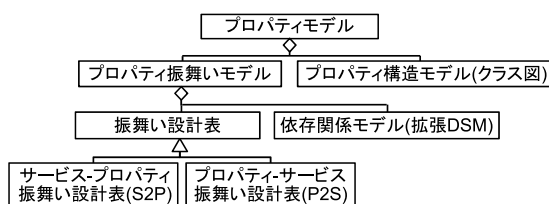


図3 プロパティモデルのメタモデル

Fig. 3 Metamodel of behavioral properties.

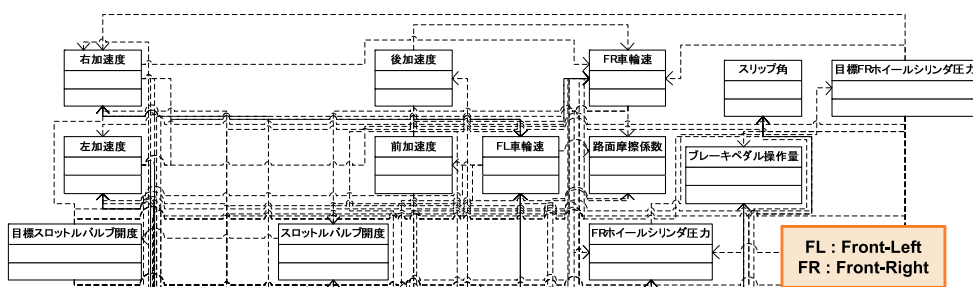


図4 プロパティ構造モデル (一部抜粋)

Fig. 4 An example of property structure model (excerpt).

プロパティをDSMの行と列に対応づけると交点の要素はプロパティ間の影響を表す。本稿では、図5に示すように、DSMの列を拡張して非機能要求を定義することで、プロパティの非機能要求への影響を表現する。行と列の要素間に影響がある場合はXにより表す。図ではプロパティ間の影響として、AからB、BからAとCへの影響を表す。さらに、プロパティから非機能要求への影響として、プロパティA、B、Cから非機能要求P、Q、Rに、それぞれ影響することを表す。

ハイブリッド車の仕様書[24]を例に作成した拡張DSMを表1に示す。スロットルバルブ開度に着目すると、スロットルバルブ開度が目標スロットルバルブ開度から影響を受け、スロットルバルブ開度は車輪速に影響することが分かる。また、ブレーキペダル操作量に着目すると目標ホイールシリンダ圧力に影響することが分かり、加えて、非機能要求の制動力支援とブレーキスリップ率に影響することが分かる。

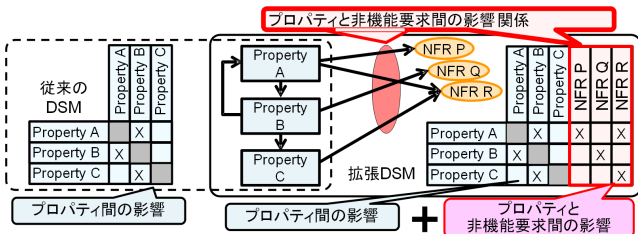


図5 拡張DSMによる影響伝播のモデル化

Fig. 5 Property dependency model by extended DSM.

4.3.3 振舞い設計表

振舞い設計表はサービスとプロパティ間の相互依存関係とサービスと非機能要求間の依存関係をサービスのリアルタイム性も含めて表形式でモデル化する。振舞い設計表によりサービスの同時起動とサービス競合の可能性が特定可能となる。

ハイブリッド車の仕様書[24]を例に作成した振舞い設計表を表2に示す。左側がサービス-プロパティ振舞い設計表で右側がプロパティ-サービス振舞い設計表である。

4.3.4 プロパティ振舞いモデル

プロパティ振舞いモデルは、DSMによる依存関係モデルと振舞い設計表を組み合わせて自動車全体の振舞いをモデル化し、協調制御の設計を可能とする。振舞いモデルはサービスのプロパティへの直接的影響だけでなく、間接的影響も表現できることに意義がある。

例として、ハイブリッド車の仕様書[24]に基づき作成したプロパティ振舞いモデルを図6に示す。ABS (Antilock Brake System) の実行結果が目標ホイールシリンダ圧力の変更を通して、ホイールシリンダ圧力へ影響する。さらに、ホイールシリンダ圧力が車輪速に影響し、車輪速がABSに影響する。ABSの目標ホイールシリンダ圧力への直接的影響だけでなく、車輪速などへのABSの間接的影響も表現可能である。ABSが制動力支援などの非機能要求に影響することも特定可能である。非機能要求を満たす協調制御の設計を支援する。

表1 拡張DSMによる影響伝播モデル

Table 1 A property dependency model by extended DSM for a hybrid car.

表2 振舞い設計表

Table 2 A property dependency table for a hybrid car.

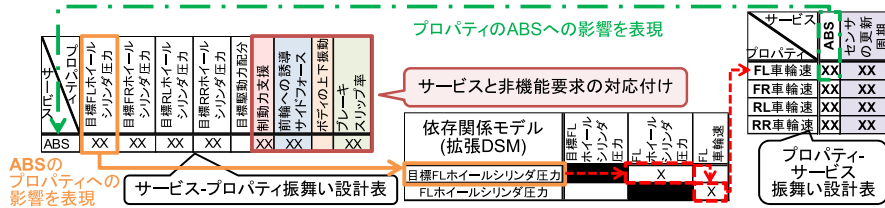


図 6 プロパティ振舞いモデル
Fig. 6 A property behavior model for a hybrid car.

表 3 プロパティとサービスの相互作用パターン
Table 3 Classification of interactions between properties and services.

	プロパティ	サービス
プロパティ	(1) プロパティ間の相互作用がある	(2) プロパティ-サービス間の相互作用がある
サービス	(3) サービス-プロパティ間の相互作用がある	(4) 考慮しない

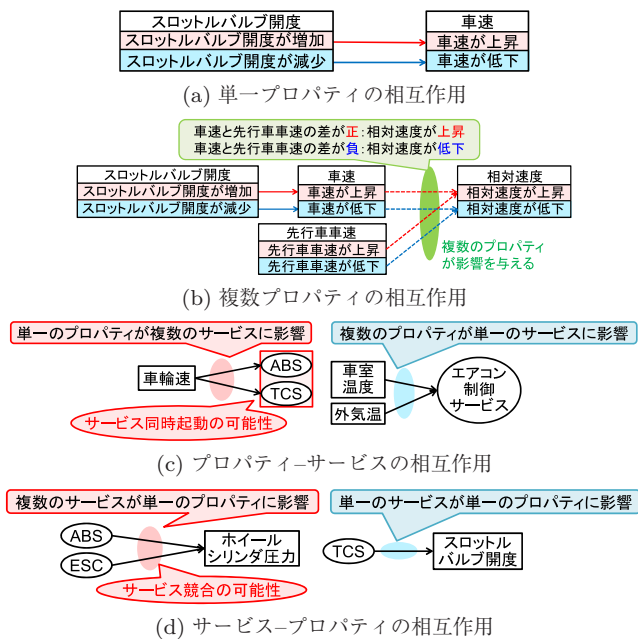


図 7 プロパティとサービスとの相互作用の 4 パターン

Fig. 7 Four interaction patterns of properties and services.

4.4 プロパティとサービスとの相互作用分析

サービスがプロパティを制御することによる相互作用を分析し、4パターンに分類した(表 3).

ここで、プロパティがさらに他のサービスの実行にも影響することに留意する必要がある。

(1) プロパティ-プロパティの相互作用

あるプロパティが異なるプロパティに影響する相互作用である。プロパティ間の影響伝搬の観点から、さらに4つのパターンに分類できる。ここでは特に、プロパティが単一のプロパティに影響するパターンについて述べる。

(1-a) 単一プロパティが単一のプロパティに影響するパターン

例として、スロットルバルブ開度と車速の2つのプロパティに着目する。スロットルバルブ開度の変化が車速に与える影響を図 7(a) に示す。スロットルバルブ開度が増加すると車速が上昇し、スロットルバルブ開度が減少すると

車速が低下することが分かる。この場合、スロットルバルブ開度のみが車速に影響する。

(1-b) 複数のプロパティが単一のプロパティに影響するパターン

例として、相対速度、車速、先行車車速、スロットルバルブ開度の4つのプロパティに着目する。これらの4つのプロパティの相互作用を図 7(b) に示す。スロットルバルブ開度の増加により車速は上昇する。しかし、先行車車速が車速よりも速い場合、車速が上昇しても相対速度は上昇しない。

(2) プロパティ-サービス相互作用

プロパティがサービスに影響する相互作用がある。特に、単一プロパティが複数サービスに影響するパターンでは、複数サービスが同時起動する可能性がある。

図 7(c) の例では、車輪速は ABS と TCS (Traction Control System) に影響を与え、車室温度と外気温度はエアコン制御サービスに影響を与える。この場合、ABS と TCS が車輪速の影響により同時起動する可能性があることが分かる。

(3) サービス-プロパティ相互作用

サービスがプロパティに影響する相互作用がある。特に、複数サービスが単一プロパティに影響するパターンでは、サービス競合が発生する可能性がある。

図 7(d) の例では ABS と ESC (Electronic Stability Control) がホイールシリンダ圧力に影響を与えることが分かる。ABS と ESC が同時にホイールシリンダ圧力に影響を与えるとサービス競合の可能性がある。

(4) サービス-サービス相互作用

サービス間の相互作用はプロパティを介して間接的に行われると考えられるので、サービス間の直接的な相互作用は考慮しない。

5. 協調制御アーキテクチャ設計プロセス

本稿で提案する協調制御アーキテクチャの設計プロセスを図 8 に示す。

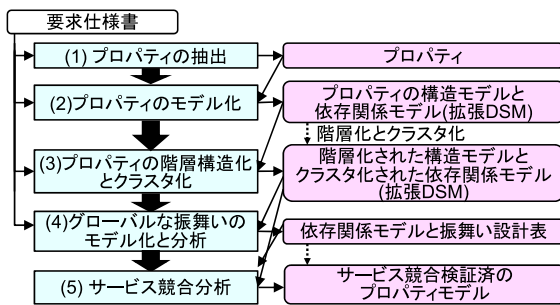


図 8 協調制御アーキテクチャ設計プロセス

Fig. 8 Architecture design process based on the behavioral property models.

(1) プロパティの抽出

プロパティを仕様書のセンサ検出値やサービスの制御則から抽出し、4.2 節で述べたプロパティの分類に基づき分類する。センサで検出できないプロパティは走行中の周辺環境を考慮して抽出する。

(2) プロパティのモデル化

分類したプロパティと仕様書などより抽出した非機能要求に基づき、プロパティ間とプロパティと非機能要求間の関係を分析してプロパティ構造モデルを作成する。さらに、プロパティを行に、プロパティと非機能要求を列にとり、その間の相互作用に着目して拡張 DSM による依存関係モデルを作成する。

(3) 構造モデルの階層化と依存関係モデルのクラスタ化

プロパティの構造モデルを階層化する。プロパティの階層構造モデルはプロパティを階層的にモデル化し、センサやアクチュエータなどの構成の違いを捨象してモデルを抽象化する。プロパティには同一のアクチュエータから抽出されたものがある。たとえば、図 4 において、FL 車輪速、FR 車輪速は車輪速センサから抽出される。そのため、同じ型のアクチュエータから抽出された複数のプロパティは同一型の上位のプロパティの具体化と考える。

一方、異なるアクチュエータやサービスの制御則から抽出されたプロパティも階層構造化される場合がある。この場合、上位のプロパティは下位のプロパティにより構成されると考えることができる。これによって、プロパティの型に基づく静的な階層構造化が行える。

一方、拡張 DSM で表される依存関係モデルに対して階層的クラスタリング [19] を行う。類似度の高いプロパティと非機能要求をまとめてクラスタとし、さらにクラスタ間の類似度を計算してクラスタリングを繰り返す。階層的クラスタリングによってクラスタ内に再帰的にクラスタが生成される場合、プロパティの依存関係を階層構造として抽出し、依存関係に基づく階層化が行える。階層的クラスタリングには DSM ツール [7] が適用できる。

(4) グローバルな振舞いのモデル化と分析

サービスとプロパティ間の影響とサービスの非機能要求

への影響を分析して振舞い設計表を作成する。振舞い設計表と依存関係モデルを組み合わせると振舞いモデルを作成する。振舞いモデルは自動車のグローバルな振舞いを表し、自動車組込みソフトウェアの協調制御が振舞いに与える影響を分析できるため、自動車全体のスコープから自動車組込みソフトウェアの協調制御設計が可能となる。

(5) サービス競合分析

プロパティ構造モデルとプロパティ振舞いモデルを重ね合わせてサービスとプロパティ間の依存関係をたどるとプロパティを介したサービスの影響の伝播をトレースできる。このサービスの伝播経路をユースケースマップ [6] の記法を用いて表現することにより、サービスのユースケースパスが重複するプロパティを抽出し、サービス競合の可能性を発見できる。

さらに、ユースケースマップをプロパティ振舞いモデルに重ね合わせて表現することによりプロパティ間の影響伝播を視覚的に表現し、プロパティと非機能要求に対するサービス競合の可能性を可視化する。これにより、次の 2 つのサービス競合の可能性を発見できる。

- 1) 複数のサービスのユースケースパスがプロパティ内で重複するとサービス競合の可能性がある。
- 2) プロパティ内で複数の非機能要求に関与するユースケースパスが重複すると非機能要求に関するサービス競合の可能性がある。

6. 提案設計方法の車両運動統合制御システム (VDM) への適用

提案設計方法の妥当性を確認するために、自動車ソフトウェアの中で最も高度で複雑な協調制御の 1 つである車両運動統合制御システム (VDM: Vehicle Dynamics Management system) [14], [22] に適用した。VDM はブレーキ、エンジン、ステアリング、サスペンションの協調制御により車両安定性の限界前から制御を行い、高度な走行性能と安定性を実現する。VDM の実現例である VDIM (Vehicle Dynamics Integrated Management system) の仕様書 [23] に提案方法を適用した。

6.1 プロパティの抽出

VDIM 仕様書 [23] より 68 個のプロパティを抽出し、4.2 節で述べた分類基準に基づき分類した。

- (1) ユーザプロパティ: アクセルペダル開度, ステアリング操舵角, ステアリング操舵方向, ブレーキペダル操作量, ブレーキペダル操作速度の 5 個を抽出した。
- (2) 自動車プロパティ: 61 個の走行プロパティのみを抽出した (表 4)。
- (3) 環境プロパティ: 現在位置, 路面摩擦係数の 2 個を抽出した。

表 4 VDIM の走行プロパティ
Table 4 Driving properties of VDIM.

プロパティと自動車の対応関係	目標値	現在値
FL (Front-Left)	目標 FL 車輪速, 目標 FL ホイールシリンダ圧力, 目標 FL 車高, 目標 FL サスペンション減衰力	FL 車輪速, FL 回転方向, FL ホイールシリンダ圧力, FL 車高, FL サスペンション減衰力, FL ばね上加速度
FR (Front-Right)	目標 FR 車輪速, 目標 FR ホイールシリンダ圧力, 目標 FR 車高, 目標 FR サスペンション減衰力	FR 車輪速, FR 回転方向, FR ホイールシリンダ圧力, FR 車高, FR サスペンション減衰力, FR ばね上加速度
RL (Rear-Left)	目標 RL 車輪速, 目標 RL ホイールシリンダ圧力, 目標 RL 車高, 目標 RL サスペンション減衰力	RL 車輪速, RL 回転方向, RL ホイールシリンダ圧力, RL 車高, RL サスペンション減衰力, RL ばね上加速度
RR (Rear-Right)	目標 RR 車輪速, 目標 RR ホイールシリンダ圧力, 目標 RR 車高, 目標 RR サスペンション減衰力	RR 車輪速, RR 回転方向, RR ホイールシリンダ圧力, RR 車高, RR サスペンション減衰力, RR ばね上加速度
自動車全体	目標スロットルバルブ開度, 目標ヨーレート, 目標懸動け直分, 目標ロール角, 目標ピッチ角, 目標ステアリングアシストトルク	前加速度, 後加速度, 右加速度, 左加速度, スロットルバルブ開度, スリップ角, スリップ角速度, ヨーレート, 懸動け直分, 上加速度, 下加速度, ロール角, ピッチ角, 予測スワット, ステアリングアシストトルク

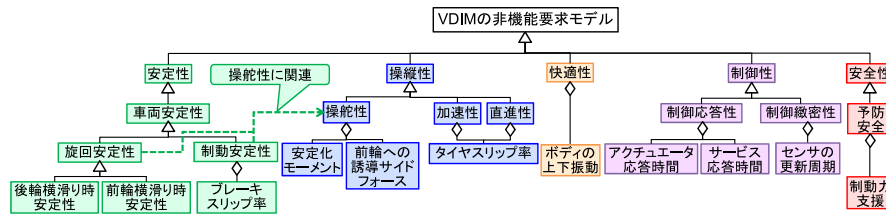


図 9 VDIM の非機能要求のモデル化
Fig. 9 Non-functional requirements for VDIM.

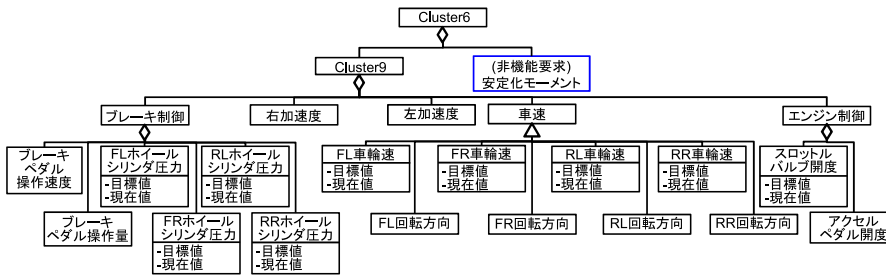


図 10 VDIM のプロパティ階層構造化モデル (一部抜粋)
Fig. 10 A property structure model for VDIM (excerpt).

6.2 プロパティのモデル化

プロパティ間の依存関係を分析した。VDIM 仕様書 [23] には VDIM を構成するサービスの入力値と出力値が記述されている。入力値はサービスの実行により出力値に影響を与えると考え、出力値に該当するプロパティは入力値に該当するプロパティと依存関係を持つと分析した。さらに、仕様書に基づき VDIM の非機能要求を抽出した (図 9)。この分析より VDIM の構造モデルを作成した。結果を付録 A.1 に示す。

6.3 プロパティの階層構造化とクラスタ化

ブレーキ制御、エンジン制御、サスペンション制御、ステアリング制御の振舞いに着目してプロパティを抽出した。ここで、車輪速センサに着目して、車輪速と回転方向のプロパティをまとめて車速として拡張 DSM によりモデル化した (図 10)。拡張 DSM を DSM ツール [7] により階層的クラスタリングした結果、新たなクラスタが 12 個生成され、クラスタの中にさらにクラスタが存在するものが 5 個発見できた。結果を付録 A.2 に示す。特に、ブレーキ制御、エンジン制御、車速を含む Cluster9 に着目すると、

階層構造モデルが構成できる。Cluster9 のプロパティを制御するサービスとして、エンジン制御とブレーキ制御により車速を制御する定速走行サービスを追加可能であることが分かった。このように、生成されたクラスタへの制御としてサービスが特定可能となる。

6.4 グローバルな振舞いのモデル化と分析

VDIM 仕様書 [23] より VDIM を構成するサービスを 11 個抽出してサービスとプロパティ間、サービスと非機能要求間の依存関係を分析し、2 つの振舞い設計表を作成した (表 5, 表 6)。さらに、依存関係モデルと振舞い設計表を組み合わせると振舞いモデルを作成した。

表 5 より複数サービスが単一プロパティに影響するパターンを発見した。ABS, ブレーキアシスト, ナビブレーキアシスト, TCS (Traction Control System), ESC (Electronic Stability Control), ステアリング協調, ヒルスタートアシストがともに目標ホイールシリンダ圧力に影響し、サービス競合の可能性はある。

さらに、表 6 より単一プロパティが複数サービスに影響するパターンを発見した。ブレーキペダル操作量はブレー

表 5 VDIM のサービスプロパティ振舞い設計表
Table 5 Services-to-properties behavioral model for VDIM.

複数サービスが単一プロパティに影響 → サービス競合の可能性を発見 XX: ハードリアルタイム

プロパティ	サービス	目標FLホイールシリンダ圧力	FLホイールシリンダ圧力	目標FRホイールシリンダ圧力	FRホイールシリンダ圧力	目標RLホイールシリンダ圧力	RLホイールシリンダ圧力	目標RRホイールシリンダ圧力	RRホイールシリンダ圧力	目標スロットルバルブ開度	スロットルバルブ開度	目標駆動力配分	駆動力配分	目標FFスリッパ	FFスリッパ	目標FRスリッパ	FRスリッパ	目標RRスリッパ	RRスリッパ	目標ステアリング	ステアリング	目標ヒルスタートアシスト	ヒルスタートアシスト	センサの更新周期	アクチュエータ応答時間
エンジン制御																									
ブレーキ制御			XX		XX		XX		XX																
ABS		XX		XX		XX		XX																	
ブレーキアシスト																									
ナビブレーキアシスト		XX		XX		XX		XX																	
TCS		XX		XX		XX		XX																	
ESC		XX		XX		XX		XX																	
i-Four													XX												
AVS													XX	XX	XX	XX	XX	XX	XX						
ステアリング協調		XX		XX		XX		XX																	
ヒルスタートアシスト		XX		XX		XX		XX																	
センサの更新周期			XX		XX		XX		XX																
アクチュエータ応答時間			XX		XX		XX		XX																

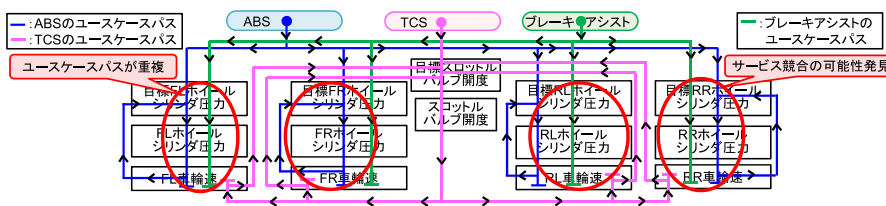


図 11 プロパティに対する ABS, TCS, ブレーキアシスト間のサービス競合の可能性の発見
Fig. 11 Search for service interactions to properties of the ABS, TCS and brake assist.

表 6 VDIM のプロパティサービス振舞い設計表

Table 6 Properties-to-services behavioral model for VDIM.

プロパティ	サービス	エンジン制御	ブレーキ制御	ABS	ブレーキアシスト	ナビブレーキアシスト	TCS	ESC	i-Four	AVS	ステアリング協調	ヒルスタートアシスト	センサの更新周期
FL車輪速				XX			XX		XX	XX	XX	XX	XX
FL回転方向				XX			XX		XX	XX	XX	XX	XX
FR車輪速				XX			XX		XX	XX	XX	XX	XX
FR回転方向				XX			XX		XX	XX	XX	XX	XX
RL車輪速				XX			XX		XX	XX	XX	XX	XX
RL回転方向				XX			XX		XX	XX	XX	XX	XX
RR車輪速				XX			XX		XX	XX	XX	XX	XX
RR回転方向				XX			XX		XX	XX	XX	XX	XX
前加速度										XX	XX	XX	XX
後加速度										XX	XX	XX	XX
右加速度										XX	XX	XX	XX
左加速度										XX	XX	XX	XX
ブレーキペダル操作量				XX	XX				XX	XX			XX
目標FLホイールシリンダ圧力			XX								XX		XX
FLホイールシリンダ圧力											XX		XX
目標FRホイールシリンダ圧力			XX										
FRホイールシリンダ圧力											XX		XX
目標RLホイールシリンダ圧力			XX										
RLホイールシリンダ圧力											XX		XX
目標RRホイールシリンダ圧力			XX										
RRホイールシリンダ圧力											XX		XX
ブレーキペダル操作速度				XX	XX								XX
ステアリング操舵角								XX			XX		XX
現在位置					XX				XX				XX
目標スロットルバルブ開度	XX												
スロットルバルブ開度								XX					XX
スリッパ角									XX				XX
スリッパ角速度									XX				XX
ヨーレート									XX				XX
目標ヨーレート									XX				XX
路面摩擦係数			XX				XX	XX					
ロール角									XX				XX
ピッチ角									XX				XX
FLばね上加速度									XX				XX
FRばね上加速度									XX				XX
RLばね上加速度									XX				XX
RRばね上加速度									XX				XX
予測スタック									XX				XX

キアシスト, ナビブレーキアシスト, i-Four (4輪駆動制御), AVS (Adaptive Variable Suspension System) に影響し, サービス同時起動の可能性がある。

6.5 サービス競合分析

図 11 に示すように, プロパティ構造モデルにプロパティ振舞いモデルから求められるユースケースマップを重ね合わせて ABS, TCS, ブレーキアシストのユースケースマップを作成すると, FL 車輪速などの 12 個のプロパティでユースケースパスが部分的に重複する. これらのプロパティに対する 3 つのサービス, ABS, TCS, ブレーキアシストがサービス競合を起こす可能性があることが分かる。

さらに, 非機能要求に影響するプロパティの影響伝搬の順序をユースケースマップで表現し, ユースケースパスが重複するプロパティを抽出して非機能要求に対するサービス競合の可能性がある. 前輪への誘導サイドフォースと制動力支援の 2 つの非機能要求に着目すると, FL ホイールシリンダ圧力などの 8 個のプロパティでユースケースパスが重複する (図 12). ここから, 前輪への誘導サイドフォースと制動力支援の 2 つの非機能要求間でサービス競合の可能性を発見できる。

7. 提案設計方法の評価

(1) プロパティに基づく振舞いのモデル化

本稿で提案したプロパティによる振舞いモデルは, 従来のソフトウェアの機能モデル, 構造モデルに対し, 異なる視点から補完するモデルとなる. 自動車組込みソフトウェアなどのリアルタイム分散処理ソフトウェアでは振舞いが本質的な特性であるので, プロパティによるグローバルな振舞いモデルは本質的モデルと考えられる. 特に, 自動車

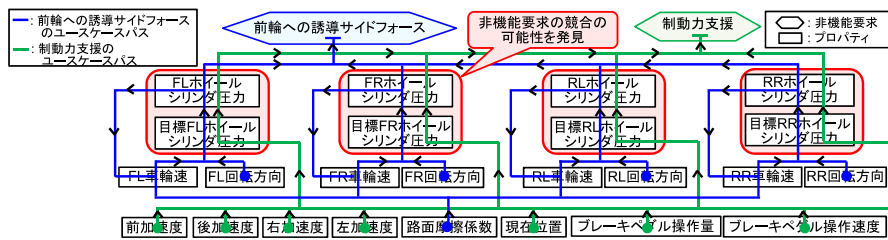


図 12 非機能要求間のサービス競合の可能性の発見

Fig. 12 Search for service interactions to non-functional requirements for VDIM.

表 7 5つの観点からの提案設計方法の考察

Table 7 Modelig capabilities of proposed method from the five viewpoints of DAF.

dAF が着目すべき観点	提案設計方法
a) 機能的観点	プロパティを介して、サービス間の相互作用を表現
b) 物理的観点	自動車の物理コンポーネントを抽象化したプロパティの構成や関連は表現したが、物理コンポーネント自体の構成や関連は表現していない
c) データの観点	自動車の振舞いを表現する情報を表現したが、ECU 間の通信プロトコルは表現していない
d) ドライバと環境の観点	自動車とユーザ、自動車と周辺環境との相互作用を表現
e) ステークホルダの観点	提案設計方法では表現していない

に加え周辺環境，ユーザなどからなる物理的実体を通じた相互作用を含む振舞いのモデル化が可能となっている．従来は自動車の複数のセンサやアクチュエータなどは ECU ごとにモデル化されたり，モデル化の対象とされていなかったりした．これらの要素を統一的にモデル化できることにより，自動車全体のグローバルな振舞いのモデル化が可能となった．

さらに，プロパティを構造と振舞いの視点から4つのモデルに分けてモデル化することにより，コンサーンを分離してモデル化できるとともに各モデルの複雑度を軽減できる．特に，プロパティ間の相互作用がクラス図による構造モデルと拡張 DSM による依存関係モデルを用いて，プロパティ間の構造と影響伝播の2つの視点に分けて表現するため，従来はモデル化できていなかった振舞いの影響の表現が可能となる．

(2) プロパティによる振舞いの視点からのソフトウェアアーキテクチャ設計

コンポーネント・コネクタモデルなど，ソフトウェアアーキテクチャの設計に関する主要な研究は構造の視点に基づく．一方，アーキテクチャ設計の主目的は性能や信頼性などの非機能要求を満たすことにある．構造中心の設計では非機能要求への影響を特定することが困難である．非機能要求と，その決定を支配する振舞いをアーキテクチャと関係づける必要がある．このようなアプローチに基づくアーキテクチャの新たなモデルが提案されている．Ahlgren らはアーキテクチャが満たすべきシステムの特性をインバリエントとして定義する [1]．本稿では，このアプローチを拡張し，プロパティによりソフトウェアアーキテクチャに直接対応付け可能な方法を提案している．また，Sant'Anna らは [18]，アーキテクチャ要素にコンサーンを割り当て，コンサーンに関するモジュール性を評価する方法を提案し

ている．しかし，アーキテクチャ全体とコンサーンとの関係づけは明確になっていない．

提案設計方法では，プロパティ間の影響伝播やサービスとプロパティ間の影響伝播が追跡可能となることから，グローバルな特性を分析できる．

一方，自動車ドメインではアーキテクチャモデル (DAF) として共通に着目すべき5つの視点が提案されている [5]．この5つの視点と提案設計方法を対比した結果を表 7 に示す．提案設計方法は a)~e) までの視点を表現できる．e) は関連するステークホルダに対する視点のため，協調制御設計では直接扱う必要がない．したがって，提案設計方法は自動車協調制御アーキテクチャの多様な視点に対応できるといえる．

(3) 拡張 DSM を用いた非機能要求の設計

構造中心設計では制御による非機能要求への影響の特定が困難である．提案設計方法ではプロパティと非機能要求間の影響を拡張 DSM で表現し，サービスの制御によるプロパティの変化が影響する非機能要求を特定できる．これは，DSM による依存関係モデル [4], [12] の拡張となっている．非機能要求を満たすためのサービスを特定可能となるため，非機能要求と協調制御設計との関連が分析可能となる．従来の非機能要求モデル，たとえば NFR フレームワーク [8] は，非機能要求とゴールとの関係を表現できるが，振舞いと関係は表現できない．自動車ソフトウェアでは振舞いが本質であることから，提案方法は有用であると考えられる．

(4) 競合の特定

従来の設計方法では分散する機能間競合や非機能要求との競合の特定が困難である．提案設計方法は振舞いに着目し，プロパティモデルにユースケースマップを応用し，サービスのプロパティへの影響をユースケースパスで表現

する。ユースケースパスの重複により、そのユースケースパスを持つサービス間の競合の可能性、非機能要求に対する制御の競合の可能性を特定できる点で改善となっている。しかし、ユースケースパスの存在は必ずしも競合の存在を意味しない。このような点から、競合を正確に検出できる方法の検討が必要である。

8. 今後の課題

今後、次のような研究が必要である。

- (1) モデルと分析方法の強化：拡張 DSM による依存関係モデルの表現能力の向上とサービス競合や非機能要求の競合の分析方法の向上が必要である。協調制御の設計では、競合回避はきわめて重要である。
- (2) 設計方法の向上：プロパティを中心とするアーキテクチャ設計としての方法の深化と支援環境が必要である。
- (3) 提案方法の評価：他の例題などによる提案設計方法の有効性と妥当性を評価する。
- (4) 関連研究領域への展開：物理コンピューティングシステム [21] や CPS [13] などの物理的実体との結合が高い領域では提案方法の活用が期待されるので、適用を検討する。

9. まとめ

自動車ソフトウェアの協調制御アーキテクチャ設計を支援するために、振舞いをプロパティでモデル化する方法とプロパティに基づくアーキテクチャの設計方法を提案した。信頼性などの非機能要求を満たすために、DSM を拡張した拡張 DSM を提案し、振舞いと非機能要求とを関係づける設計方法を提案した。提案設計方法を自動車ソフトウェアの中で最も高度で複雑な協調制御の1つである車両運動統合制御システムに適用し、プロパティによる自動車の振舞いのモデル化と、サービス競合と非機能要求の競合可能性の特定が可能であることを示した。さらに、自動車全体の協調制御の設計に有効であることを確認した。提案方法は、従来の構造に基づくアーキテクチャ設計を補完する新たな設計方法の基礎となる。

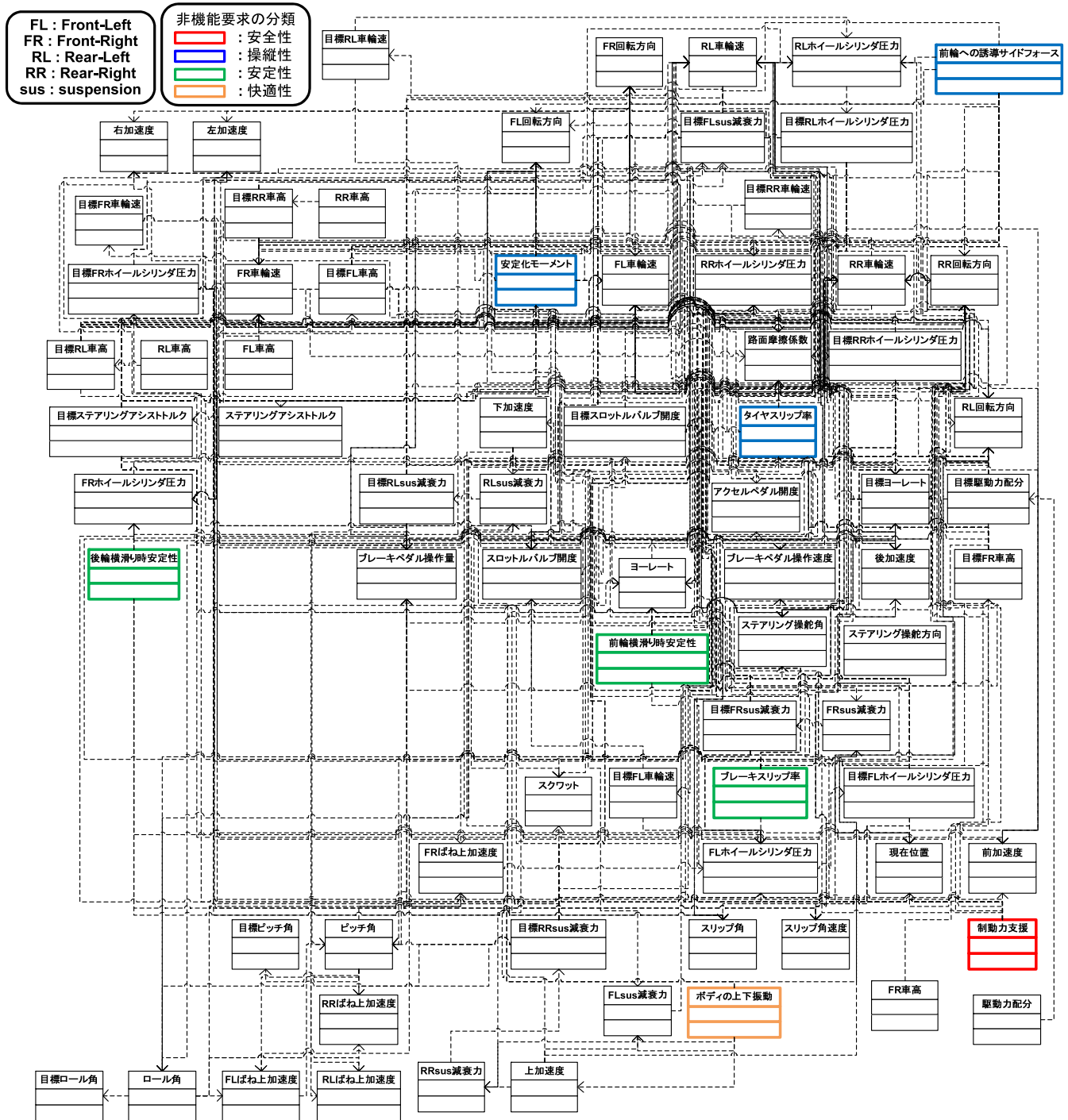
参考文献

- [1] Ahlgren, B. et al.: Invariants: A New Design Methodology for Network Architectures, *Proc. ACM SIGCOMM Workshop on Future Directions in Network Architecture*, pp.65-70 (Aug. 2004).
- [2] Albin, S.T.: *The Art of Software Architecture*, Wiley (2003).
- [3] 青山幹雄ほか：車載ソフトウェアのサービスプラットフォームのモデルとアーキテクチャ, 自動車技術会論文集, Vol.40, pp.1599-1604 (Dec. 2008).
- [4] Baldwin, C.Y. and Clark, K.B.: *Design Rules: The Power of Modularity*, MIT Press (2000).
- [5] Broy, M. et al.: Toward a Holistic and Standardized Automotive Architecture Description, *IEEE Computer*,

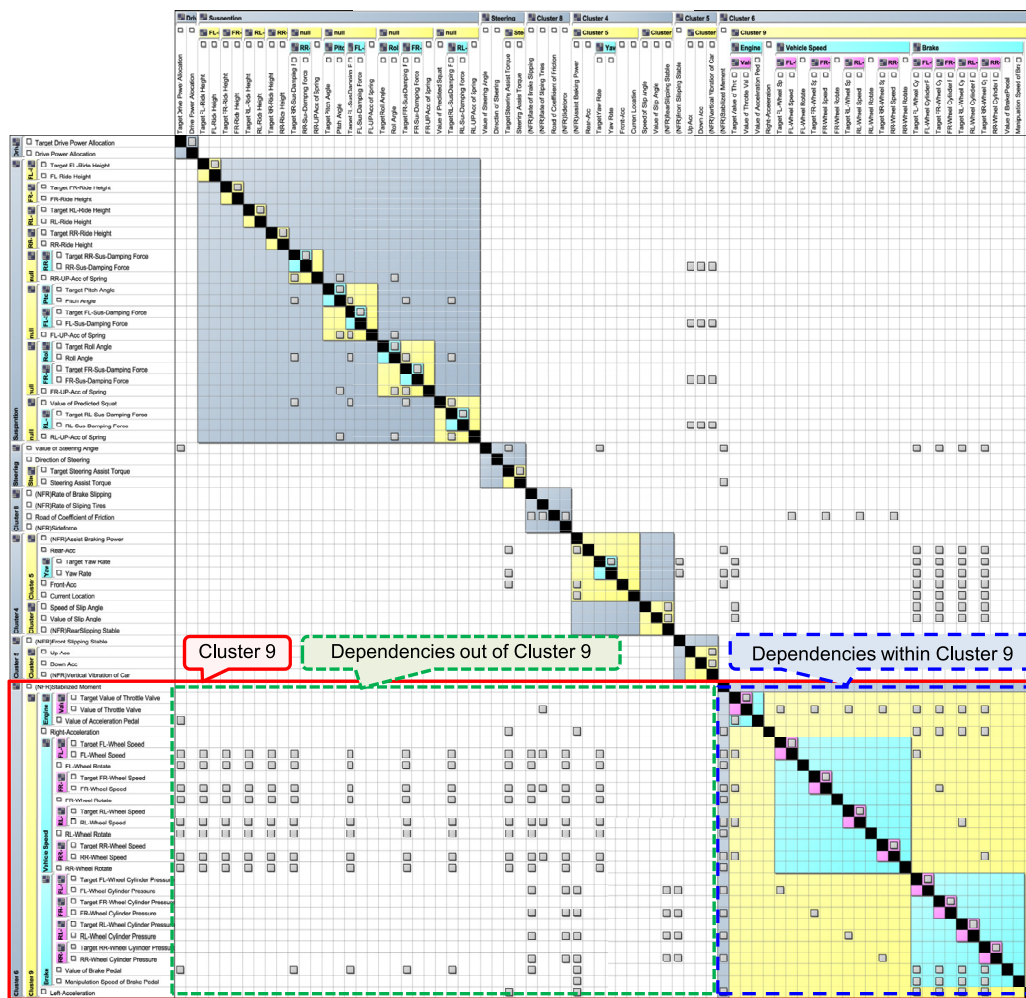
- Vol.42, No.12, pp.98-101 (Dec. 2009).
- [6] Buhr, R.J.A. et al.: *Use Case Maps for Object-Oriented Systems*, Prentice Hall (1996).
- [7] Cambridge Advanced Modeller (CAM), available from (<http://www-edc.eng.cam.ac.uk/cam/>).
- [8] Chung, L. et al.: *Non-Functional Requirements in Software Engineering*, Kluwer Academic (1999).
- [9] Farcas, C. et al.: Addressing the Integration Challenge for Avionics and Automotive Systems-From Components to Rich Services, *Proc. IEEE*, Vol.98, No.4, pp.562-583 (Apr. 2010).
- [10] Jamshidi, M. (Ed.): *Systems of Systems Engineering*, CRC Press (2009).
- [11] Juarez-Dominguez, A.L. et al.: Modelling Feature Interactions in the Automotive Domain, *Proc. MSE '08*, pp.45-50, ACM (May 2008).
- [12] Larses, O.: Applying Quantitative Methods for Architecture Design of Embedded Automotive Systems, *Proc. INCOSE 2005* (July 2005).
- [13] Lee, E.A.: Cyber Physical Systems: Design Challenges, *Proc. IEEE ICSORC '08*, pp.363-369, IEEE Computer Society (May 2008).
- [14] 見市善紀：統合安全コンセプト, 自動車技術, Vol.63, No.2, pp.82-88 (Feb. 2009).
- [15] Mitchell, B.S. and Mancoridis, S.: On the Automatic Modularization of Software Systems Using the Bunch Tool, *IEEE Trans. Softw. Eng.*, Vol.32, No.3, pp.193-208 (Mar. 2006).
- [16] Neeraj, S. et al.: Using Dependency Models to Manage Complex Software Architecture, *Proc. OOPSLA 2005*, pp.167-176, ACM (Oct. 2005).
- [17] Pretschner, A. et al.: Software Engineering for Automotive Systems: A Roadmap, *Proc. ICSE '07, Future of Software Engineering*, pp.55-71, IEEE Computer Society (May 2007).
- [18] Sant'Anna, C. et al.: On the Modularity of Software Architectures: A Concern-Driven Measurement Framework, *Proc. ECSA 2007*, LNCS 4758, pp.207-224 (Sep. 2007).
- [19] 佐藤善治：多変量データの分類—判別分析・クラスター分析, 朝倉書店 (2009).
- [20] 曾根 学ほか：EV 用 IT システム開発, 自動車技術, Vol.65, No.2, pp.40-45 (Feb. 2011).
- [21] Stankovic, J.A. et al.: Opportunities and Obligations for Physical Computing Systems, *IEEE Computer*, Vol.38, No.11, pp.23-31 (Nov. 2005).
- [22] 鈴木将人ほか：車両運動統合制御 (VDIM) の現状と展望, *TOYOTA Technical Review*, Vol.55, No.1, pp.48-53 (Nov. 2006).
- [23] トヨタ自動車：CROWN MAJESTA 新型車解説書 URS206/UZS207 (2009).
- [24] トヨタ自動車：PRIUS 新型車解説書 NHW20 系 (2003).

付 録

A.1 VIDMのプロパティ構造モデル



A.2 VDIMの依存関係モデル (拡張DSM)



青山 幹雄 (正会員)

1980年岡山大学大学院工学研究科修士課程修了。同年富士通株式会社入社。大規模ソフトウェア開発とプロジェクト管理、ソフトウェア工学の実践に従事。1986~1988年米国イリノイ大学客員研究員。1995年4月~

2001年3月新潟工科大学情報電子工学科教授。2001年4月~2009年3月南山大学数理情報学部情報通信学科教授。2009年4月より同大学情報理工学部ソフトウェア工学科教授。クラウドコンピューティング、サービス指向アーキテクチャ、組み込みソフトウェアを対象として、要求工学、ソフトウェアアーキテクチャ技術、ソフトウェア進化の研究・開発と教育・人材育成に取り組む。著書『要求工学知識体系』(共著, 2011年)ほか多数。IEEE Transactions on Services Computing等の編集委員, 本会理事を歴任。1993年情報処理学会研究賞受賞。ソフトウェア科学会, 自動車技術会, IEEE, ACM, SAE各会員。



田邊 隼希

2011年3月南山大学大学院数理情報研究科(ソフトウェア工学専修)修士課程修了。現在, 株式会社デンソーにて開発, 設計業務に従事。在学中, 自動車ソフトウェア工学の研究に従事。