社団法人 情報処理学会 研究報告
IPSJ SIG Technical Report

2006－SE－151 (4)
2006／3／23

車載組込みシステム向けアスペクト指向モデリング・分析方法論

東 篤子† 青山 幹雄‡

†南山大学 大学院 数理情報研究科 ‡南山大学 数理情報学部 情報通信学科

車載組込みソフトウェアが自動車の運転や衝突予防などの中枢を担うようになっていることから，その安全性が極めて重要となっている．さらに，車載組込みソフトウェアがネットワークを介して相互に連携するようになり，相互連携する組込みソフトウェアシステム全体の一貫した安全性の設計が求められている．本論文では，車載組込みソフトウェアの安全性の特性と課題をアスペクト指向モデルとしてモデル化し，ISO61508 の SIL(Safety Integrity Level)の基準を考慮した安全性と性能，コストなどの多元的な非機能要求を満たすシステム設計を支援するアスペクト指向組込みソフトウェア開発方法論を提案する．提案方法を車載組込みソフトウェアの例に適用し，非機能要求に応じた 3 段階のプロダクトラインを導出し，その有効性を示す．

Aspect-Oriented Modeling and Analysis for Safety-Critical Distributed Real-Time Automotive Software Systems

Atsuko Higashi†, Mikio Aoyama‡

†Graduate School of Mathematical Sciences and Information Engineering, ‡Dept. of Information and Telecommunication Engineering, Nanzan University

Automotive software is becoming a source of innovations in controlling and safety enhancement of automobiles. Thus, the safety of automotive software is critical. Moreover, the automotive software is interoperating with other automotive software and ground services over the networks. Rigorous safety assurance across the systems is required while satisfying other requirements of performance and cost. This paper proposes a modeling methodology for modeling both functional and non-functional requirements based on aspect-oriented modeling, and analyzing multiple non-orthogonal aspects across the systems. Applying the proposed methodology to a distributed automotive software system enabled to generate three product lines of different patterns of non-functional requirements while assuring certain levels of safety integrity levels of ISO61508.

1. はじめに

自動車では，X-By-Wire と呼ばれる，機械的なバックアップを持たないコンピュータ制御が走行制御の中核を担うようになり，その安全性が極めて重要になっている[1, 5, 12]．一方，自動車は大衆製品であるので，依然としてコスト競争力が重視されている．しかし，安全性とコスト要求とは，相反する側面がある．車載組込みソフトウェアの設計では，このような，相互に関連しかつ，相反する多様な要求を満たす必要がある．

従来，このような課題に対し，様々なアプローチが取られてきた．しかし，それらは主として個別の側面を扱ってきた．例えば，安全性はハードウェアの安全性の研究から発展してきた[20]．しかし，X-By-Wire の中核はソフトウェア制御にある．ソフトウェア工学の視点から，ソフトウェア開発のライフサイクルを通して安全性を指向する開発フレームワークの確立が必要である．

また，車載組込みソフトウェアがネットワークで相互連携する分散処理システムとなり，かつ，車外に対してもネットワークで接続されるようになっている[17]．このような分散処理システム全体の安全性を扱うモデルが必要となっている．

本稿では，セフティクリティカルで，かつ，リアルタイム分散処理システムである車載組込みソフトウェアの安全性を中心とする多様な非機能要求をアスペクト指向に基づき統一的にモデル化し，分析する方法を提案する．特に，ネットワークを介して連携する複数のサブシステムやコンポーネントである ECU(Electronic Control Unit)にまたがり，相互に関係し，かつ，相反しうる多次元の非機能要求をモデル化し，システム全体が非機能要求を満たすための分析方法を提案する．さらに，車載組込みシステムを例として，本稿で提案する方法を適用し，ISO 61508 の SIL (Safety Integrity Level)を参照モデルとする安全性，コスト，性能を定量的に分析する．

2. 車載組込みソフトウェアの安全性とその設計アプローチ

2.1. 車載組込みシステムのアーキテクチャ

図-1 に高度な機能を備えた車載組込みシステムの一般的アーキテクチャの例を示す[7]．各ボックスは ECU であり，組込みコンピュータシステムである．現在，車種によっては，一台に 50 個以上もの ECU が搭載されている．これらの ECU が図に示すように，異なるネットワークプロトコルを介して連携している．

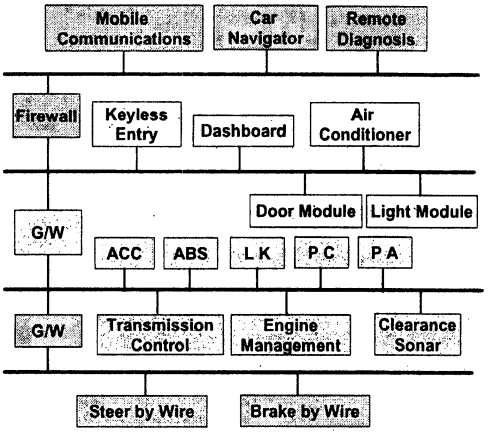

図-1 車載組込みシステムのアーキテクチャの例

2.2. 車載組込みソフトウェアシステムの特性

車載組込みソフトウェアシステムの開発には次の特性がある.

(1) 開発特性:高度な安全性と厳しいコスト制約などの相反する多元的非機能要求
(2) システムアーキテクチャ特性:リアルタイム分散処理システム
(3) ソフトウェアアーキテクチャ特性:イベントトリガ制御とタイムトリガ制御が混在するハイブリッド制御アーキテクチャをとる組込みソフトウェア
(4) プロダクト特性:マルチプロダクトライン

(1) 開発特性:高度な安全性と厳しいコスト制約などの相反する多元的非機能要求の充足
安全性は車載組込みソフトウェアの第一の要求である. 特に, 機械系のバックアップなしで走行の中核的機能を担うようになっているため, 安全性の問題は人命に直接関わる可能性がある. 一方, 大衆製品であることから, コスト低減要求も厳しい. 安全性の確保とコストダウンは相反する要求となる可能性があることから, 設計ではこのような多元的要求を充足する必要がある.

(2) システムアーキテクチャ特性:リアルタイム分散処理システム
車載システムは, エンジン制御などの走行に直接関係するシステムから, エアコン, カーナビゲーションシステムなど, 多様なシステムがネットワークで連携するリアルタイム分散処理システムであり, 厳しい時間制約とリソースの制約を満たさなければならない. ECU ごとに時間制約やリソースの制約が異なる場合もあり, システム全体としてこの多様な時間制約とリソースの制約を満たし協調しなければならない. さらにこれらのシステムが ITS (Intelligent Transport System)や地上の各種ネットワークサービスと連携して, 広域ネットワークシステムを形成することも考慮しなければならない.

(3) ソフトウェアアーキテクチャ特性:ハイブリッド型の組込みソフトウェアアーキテクチャ特性をもつ.
ECU には, エンジン制御などの周期的に起動されるタイムトリガ制御のコンポーネントと, エアバッグ制御などのように事象の発生に応じて起動されるイベントトリガ制御のコンポーネントがある. これらのコンポーネントが同じネットワーク上に混在するハイブリッド制御ソフトウェアアーキテクチャを実現しなければならない.

(4) プロダクト特性:マルチプロダクトライン
自動車は多くの派生モデルがあり, かつ, 国や地域毎に異なる法律に準拠する必要があることから, 仕様に多様性がある. そのため, プロダクトライン型開発をとる[6]. さらに, 大衆車から高級車まで, 複数のプロダクトラインがある. 車載コンピュータもマルチプロダクトライン型開発[2]が求められる.

2.3. 車載ソフトウェアの安全性へのアプローチ

従来の車載ソフトウェアの安全性へのアプローチは次のように分類できる.

(1) 要求工学のアプローチ
(2) アスペクト指向のアプローチ
(3) ディペンダビリティ・フォルトトレランスのアプローチ

(1) 要求工学のアプローチ[4, 13, 21]
要求工学では, 安全性などの非機能要求の重要性, 要求分析時の機能要求と非機能要求の分離の必要性が認識されている. 組込みソフトウェアでも, システム不具合の主要な原因として, 要求分析工程における機能要求と非機能要求の分離があいまいである点が指摘されている[15]. しかし, 非機能要求はシステムの横断的特性であり, 機能要求に比べ, そのモデル化や分析は依然として困難な課題である.

(2) アスペクト指向のアプローチ[8, 9, 11, 18, 19]
アスペクト指向は非機能特性などのシステムの横断的特性を機能要求から分離させて多元的モジュール化を推進する枠組みである. 要求分析などの上流レベルでのアスペクト分析は Early Aspect と呼ばれ, 近年, 注目されている. しかし, このレベルでは, 要求工学における非機能要求の研究と共通点が多く, 依然として, 課題が多い.

(3) ディペンダビリティ・フェイルトレランスのアプローチ[2, 3]
システムの信頼性や安全性などの包括的な概念が幾つか提案されている. これによって広範な問題を視野に入れて扱えるが, システムの包括的特性に主眼が置かれ, ソフトウェアの要求分析や設計への展開は課題となっている.
また, フォールトトレラントなシステムの実現のため

に，フェイルセイフや冗長アーキテクチャなどの技術が航空宇宙システムなどで開発されてきた．しかし，車載組込みソフトウェアでは，安全性とともに経済性も主要な課題であることから，両者の関連を分析し，要求に応じた適度な安全性を実現する必要がある．

2.4. 従来のアプローチの問題点

従来，車載組込みソフトウェアの非機能要求は機能要求と同様に重要な設計要素と考えられ，安全性などの非機能要求を扱うために 2.3 項で挙げたようなアプローチが取られてきた．

要求工学やアスペクト指向では，安全性を含む非機能要求について研究されているが，安全性についての定性的，あるいは，定量的扱いが実務への適用の点でいまだ不十分である

また，安全性やフェイルセイフのアプローチから，安全性の解析に関する多くの成果がある．しかし，これらは，安全性のみを扱い，他の要求との関係が議論されていない．

本稿で扱う車載組込みシステムでは，前述のように，安全性，性能，コストという相互に関連し，かつ，相反する非機能要求を一定の水準で満たす設計方法論が求められている．これは，従来の個別の特性を対象とする方法論では解決が難しい．

2.5. 安全性指向設計へのアプローチ

安全指向開発フレームワークの検討に当たり，車載組込みソフトウェアの特性と従来のアプローチに基づき，図-2 に示すプロセスとプロダクトの視点から，次の戦略を設定した．

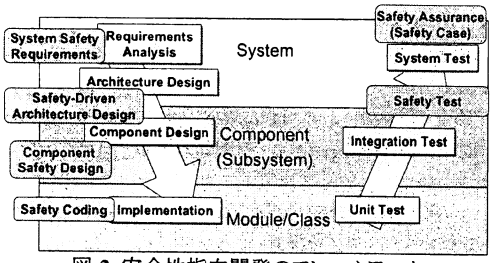

図-2 安全性指向開発のフレームワーク

(1) プロセス：ソフトウェア開発プログラムへの安全設計の統合
開発プロセスに沿って適切な安全技術を統合することにより，横断的特性である安全性をプロセスによって織込む(Weaving)戦略をとる．

(2) プロダクト：SIL に基づく安全性のモジュール化
複合システムのサブシステム，あるいは，システム群の振舞いに応じて異なる安全性を SIL により統一的に評価するとともに，サブシステム，あるいは，システム群の振舞いに応じて異なる基準値をとる安全性のモジュール化を支援する．

3. 多元的アスペクト指向モデリング・分析方法論

3.1. 多元的アスペクト指向プロダクトライン開発プロセス

本論文では，図-3 に示す多元的アスペクト指向プロダクトライン開発プロセスに従い，次のステップで多元的アスペクト分析プロダクトラインアーキテクチャを導出する．

(1) 図-1 のコンポーネント/コネクタモデルを前提として，次のステップでプロダクトラインアーキテクチャをメタモデル化する．

(2) プロダクトラインアーキテクチャのメタモデルと機能要求，非機能要求からアスペクト指向モデルを導出する．アスペクト指向モデルでは，コンポーネント/コネクタとアスペクトの関係を示す．レベル分けされた各アスペクトから，コンポーネント，コネクタごとに適切なレベルのアスペクトを割り当てる．

(3) アスペクト指向モデルを入力として，アスペクトトレース，アスペクトドメイン，アスペクトマッチングのコンセプトに基づき，多元的アスペクト指向分析を行う．

(4) 上記ステップまでに生成されたアスペクトを割り当てた各コンポーネント，コネクタと，多元的アスペクト指向分析の結果を再利用可能な資産として組合せることによりプロダクトラインの条件を満たすプロダクトラインアーキテクチャを導出する．

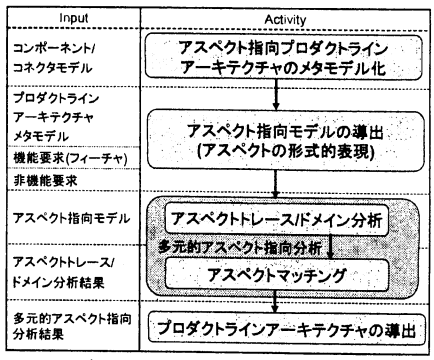

図-3 多元的アスペクト指向プロダクトライン開発プロセス

3.2. アスペクト指向モデリング・分析のプロセス

図-4 に本稿におけるアスペクト指向モデリング・分析のプロセスの一例を示す．一般に，モデリングのアプローチには，トップダウンとボトムアップがある．本稿では，ECU 毎に安全性などの様々な非機能要求があることを出発点とし，それらを組み合わせたネットワーク分散処理システムが所与の条件を満たすようにすることを主眼としている．従って，ボトムアップのアプローチを基礎としている．しかし，システムの新規開発など，従来型の開発に対しては，トップダウンアプローチなどの他のプロセスを想定する．

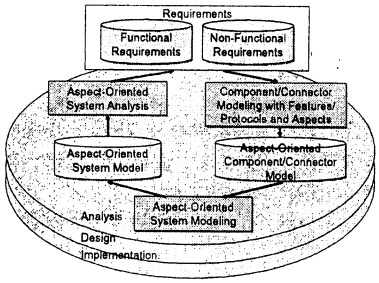

図-4　アスペクト指向モデリングのプロセス

3.3. 対象システムのメタモデル

(1) 対象システムのメタモデル

図-5 に，本稿で提案するモデリングの対象要素を定義するメタモデルを示す．アーキテクチャモデルとして広く知られているコンポーネント/コネクタモデルを想定する．本稿の対象システムは，それ自体がコンピュータシステムである ECU をコンポーネントとし，複数の ECU が異なる特性の通信プロトコルを提供するコネクタを用いてネットワーク上で連携する分散システムである[10]．同システムを次の 2 つの視点でモデル化する．

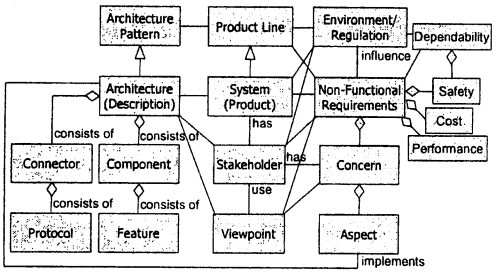

図-5 アスペクト指向モデルのメタモデル

(1)アーキテクチャを軸にフィーチャを基礎とする視点：
アーキテクチャの構成要素はコンポーネントとコネクタである．コンポーネントは要求機能を表すフィーチャで構成される．コネクタはコンポーネント同士のインタフェースとなる車載ネットワークのプロトコルで構成される．

(2)非機能的要求を軸にアスペクトを基礎とする視点：
非機能要求は横断的特性であるコンサーンで構成される．コンサーンにはシステムの視点で必要なものと，ステークホルダ(開発関係者)の視点で必要なものがあり，設計時に考慮される点である．アスペクトはコンサーンを実装したもの*である[8]．

本稿では，後述する事例で，非機能要求，あるいはコンサーンとして，安全性，性能，コストを取り上げる．さらに，車載組込みシステムでは，プロダクトライン開発が重視されることから，プロダクトラインをモデルに導入する．

3.4. アスペクト指向モデリングのためのアスペクトの表現

コンポーネントとコネクタを定義する上で，従来，アスペクトは明示的に定義されていなかった．本稿では，アスペクトをフィーチャやプロトコルと同等の重要な設計情報と位置づけ，コンポーネントとコネクタを式(1)(2)に示すように，フィーチャ/プロトコルとアスペクトの対として定義する．

Component = F(Feature, Aspects)　(1)

Connector = P(Protocols, Aspects)　(2)

図的表現としては，図-6 に示すように，UML のクラス図の拡張として，アスペクトを表現する．

図示するように，各コンポーネント，コネクタは複数のアスペクトを持つ．

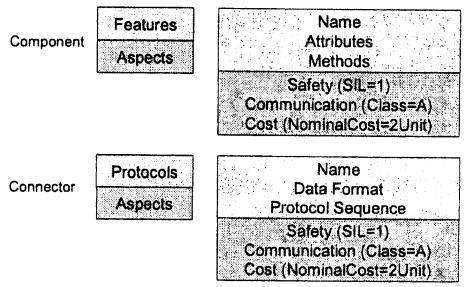

図-6 コンポーネント/コネクタのアスペクトの図的表現

3.5. アスペクト指向モデルの多次元アスペクト分析

リアルタイム分散処理組込みシステムでは，コネクタを介してコンポーネントを組み合わせてシステムを構成する上で，システムの横断的特性であるアスペクトのマッチングを取ることが設計の困難さの原因となっている．

本稿では，分散システム上で，アスペクトやアスペクト群毎のマッチングを設計・検証するために，図-7 に示す，次の諸概念を提案する．

(1)アスペクトトレースとアスペクトドメイン

アスペクトトレースとはコネクタを介して連携するコンポーネン群で，制御の流れに沿って順序付けられたアスペクトの集合である．

コンポーネントあるいはコネクタ i の n 個のアスペクトは式(a)のベクトルとして定義できる．

$\boldsymbol{Ai}$=(Ai(1), Ai(2), ..., Ai(n))　(a)

j 個のコンポーネントとコネクタにわたるアスペクトトレースは，式(b)で表される．

$\boldsymbol{Tj}=\cup j\boldsymbol{Aj}=(\cup j\ \mathrm{Aj}(1), \cup j\ \mathrm{Aj}(2), ..., \cup j\ \mathrm{Aj}(n))$　(b)

ここで，$\cup j$ はj個の集合の和を表す．図-7 では，

---

* アスペクトは，コストなどの実装とは直接対応しない要素も含む．従って，図-5 のメタモデルではコンサーンと呼ぶべきものである．しかし，オブジェクト指向モデリングと同様，要求分析から実装まで一貫してモデル化と分析ができるように，本稿では，以後，特に断らない限り，コンサーンとアスペクトをまとめてアスペクトを呼ぶ．

Componnet1 から Component2, 3, 4 へと制御の流れがある. さらに, Coponent4 は Component5 を利用している. 従って, 3 つの制御の流れに沿って, アスペクトトレースを定義できる.

アスペクトトレースの共通集合はアスペクトが同一の範囲を定義する. これは, アスペクトがマッチングしているコンポーネントとコネクタの範囲を示す. アスペクト毎に同一のアスペクトとなるコンポーネントとコネクタの範囲をアスペクトドメインと呼ぶことにする. アスペクトドメインは式(c)で定義される.

$$Dk=\cup k(Tk \mid Ck) \qquad (c)$$

すなわち, アスペクトが Ck=(Ck(1), Ck(2), ..., Ck(n)) という条件を満たすトレースの和集合となる.

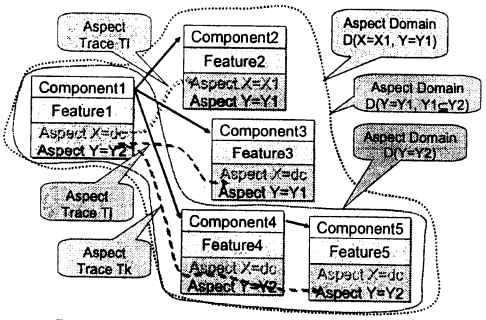

図-7 アスペクトトレースとアスペクトドメイン

(2)アスペクトマッチング

アスペクトマッチングとはシステムのアスペクトの整合性をとることである. このマッチングには強いマッチングと弱いマッチングを定義できる.

(i) 強いアスペクトマッチング: システム全体に渡りすべてのアスペクトトレースが所与の条件を満たす.

(ii) 弱いアスペクトマッチング: システムの所定のアスペクトのトレースが所定の範囲で所与の条件を満たす.

(3)アスペクトマッチングのための優先順位付け: 必須アスペクトとオプションアスペクト

アスペクトには, 必達すべきアスペクトと, 必達は要求されないが達成が望ましいアスペクトに分類できる. 本稿では, 前者を必須アスペクト, あるいは主アスペクトと呼び, 後者は, オプションアスペクトと呼ぶ.

アスペクト分析では, 決定可能性の点から, 必須アスペクトは単一とすべきである. もし必須アスペクトに相当するアスペクトが複数ある場合は, まず, 最重要のアスペクトを必須アスペクトとし, それが満たされたなら, その満足条件の範囲で, 次に重要なアスペクトを必須アスペクトとして, 順次, アスペクト分析を行なう.

車載組込みシステムでは, 安全性が必達要求であるので, 安全性を必須アスペクトとする. また, 性能アスペクトも安全性に次いで重要であるが, 後述するように, 安全性と関連するアスペクトである.

4. 車載組込みシステム例題への適用

4.1. 車載組込みシステムの例題とアスペクト

図 1 に示す, 車載組込みシステムを例としてアスペクト指向モデリング・分析の概要とその効果を示す.

ここで, 要求仕様として満たすべきアスペクトは, 安全性, 性能, コストである. 一般に, 安全性を高めるためには, 開発コストが増大することから, 安全性とコストは相反するアスペクトである. また, 安全性を保証するため, コネクタの通信プロトコルや性能に一定の条件が課されることから, 安全性と性能は相反しないが, 関連するアスペクトとなる. 本稿では, 上記の 3 種類のアスペクを表-1 に示す尺度と基準値で定義する.

表-1 アスペクトの定義

アスペクト	尺度	基準値
安全性	IEC 61508 の SIL	1(Low)~4(High)
通信性能	SAE Class[12]	A(Low)~D(High)
コスト	Unit Cost	1, 5, 10, >10

4.2. ISO/IEC61508 の安全性アスペクトの基準

組込みシステムを対象とした安全性基準として表-2 に示す ISO/IEC61508 の SIL(Safety Integrity Level)が提案されている[14].

表-2 ISO/IEC61508 の 2 つの実行モデルの SIL

SIL	タイムトリガ[発生率＝発生数/時]	イベントトリガ [発生率=発生数/使用回数]
4	$10^{-9}<X<10^{-8}$	10-5<X<10-4
3	10-8<X<10-7	10-4<X<10-3
2	10-7<X<10-6	10-3<X<10-2
1	10-6<X<10-5	10-2<X<10-1

車載組込みシステムでも, その導入が検討されている[16]. 本稿では, (1)安全性のモジュラリティが扱える, (2)タイムトリガ制御とイベントトリガ制御の両方の安全性モデルが扱えることから SIL をアスペクトとして採用する. しかし, SIL ではタイムトリガ制御とイベントトリガ制御とを分けているので, アスペクトマッチングでは, 安全性アスペクトのマッチングに加え, 性能アスペクトの制御型(タイムトリガ, イベントトリガ)ともマッチングを取る必要がある.

4.3. 車載組込みシステムのアスペクト指向モデル

図 1 のコンポーネントに対し, 安全性(Safety), 性能(Com), コスト(Cost)のアスペクトを割り当て, さらに, コンポーネント間の関連を加えて図-8 に示す. 図-8 に示すように, Brake-by-Wire と ABS では同様の機能を提供するが, 接続されるネットワークプロトコルによっていずれかのコンポーネントが選択されるため, Alternative(代替)の関係をもつ部品と考える. このようなコンポーネント群では, アスペクトやアスペクトの満たすべきレベルによっていずれかを選択することにより, 別のプロダクトラインアーキテクチャを構成できる.

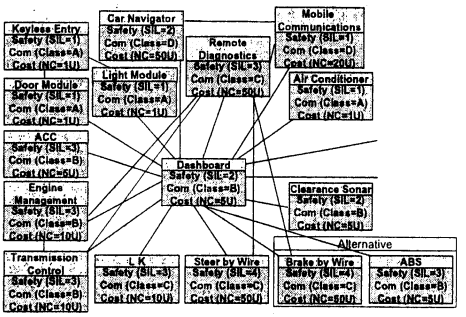

図-8 車載組込みシステムのアスペクトネットワーク

4.4. アスペクト指向モデルの分析

図-8 だけでは，アスペクトトレースが分りにくいので，必達アスペクトである Safety を必須アスペクトとし，Safety と性能，Safety と Cost との 2 つの 2 次元の表に分けた結果をそれぞれ図-9a，図-9b に示す．

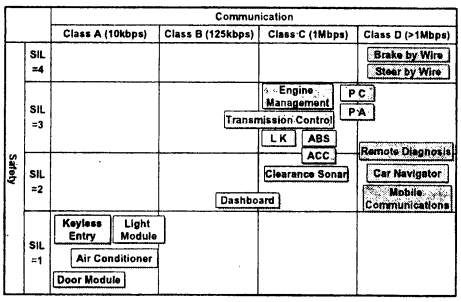

図-9a アスペクトマトリクス (Safety vs Performance)

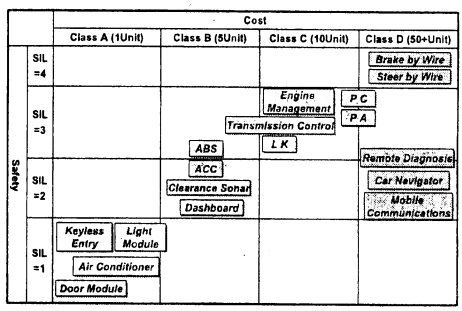

図-9b アスペクトマトリクス (Safety vs Cost)

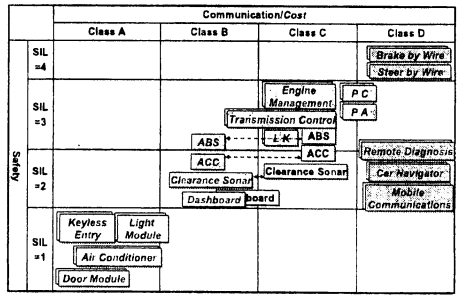

図-9c アスペクトマトリクスのオーバーレイ (Safety vs Performance/Cost)

これら 2 つのマトリクスを重ねて 3 つのアスペクトと各コンポーネントとの関係を図-9c に示す．図-9c の赤の破線で示すように，個々に分析したマトリクスを重ねると，ABS ECU，ACC ECU，Clearance Sonar ECU，Dashboard ECU のように，位置が重ならないコンポーネントが存在する．これらのコンポーネントについては，Safety 以外のいずれかのアスペクトを 2 番目の必須アスペクトとして扱わなければならない可能性があるため，設計時には注意すべきコンポーネントとアスペクトであることが分かる．

なお，図 8 をもとにコンポーネント/コネクタと安全性，通信性能，コストとの関連を表したものが表-3 である．この表を，コンポーネント/コネクタと各アスペクトの関連を表す部品表として利用できる．

表-3 コンポーネント/コネクタとアスペクトのマトリクス

Safety-Criticality: Throughput: NFR			Component																										
			Information Systems				Body Electronics							Vehicle Control Systems															
														A													B		
			Mobile Communications	Car Navigator	Remote Diagnosis	MOST	Air Conditioner	Keyless Entry	Dashboard	Door Module	Light Module	Low-Speed CAN	ACC (Adaptive Cruise Control)	ABS	LK (Lane Keep)	PC (Pre-Crash)	PA (Parking Assist)	Transmission Control	Engine Management	Clearance Sonar	High-Speed CAN	Steer by Wire	Brake by Wire	FlexRay
Dependability	Time-Driven		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	3	3	-	-	4	4	4
	Event-Driven		2	2	3	2	1	1	2	1	1	1	3	3	3	4	1	-	-	2	3	-	-	-
Perform-ance	Time Behavior	Response Time (ms)	D C	D C	D C	D C	D C	D C	D C	D C	D C	D C	D C	D C	D C	D C	D C	D C	D C	D C	D C	40 ms	20 ms	5 ms
		Throughput	4	4	3	4	1	1	2	1	1	1	2	2	3	4	3	2	2	2	2	3	3	3
Trigger Condition	Time-Driven		-	-	-	-	-	-	-	-	-	-	-	+	-	-	-	+	+	-	+	+	+	+
	Event-Driven		+	+	+	+	+	+	+	+	+	+	+	-	+	+	+	-	-	+	-	-	-	-
Cost			4	4	4	4	1	1	2	1	1	1	2	2	3	3	3	3	3	2	2	4	4	4

4.5. プロダクトラインアーキテクチャの導出

前項までに，車載組込みシステムアーキテクチャを構成するコネクタ，コンポーネント，およびそれぞれに関連するアスペクト(安全性，通信性能，コスト)とアスペクト間の関係について分析した．本項では，プロダクトライン開発の視点から，これらの部品を再利用可能な資産として考え，プロダクトラインアーキテクチャの導出を試みる．

図-8 と表-3 のアスペクトに基づき，安全性アスペクトのマッチングを取ると，図-10a に示す，4 つのサブシステムにグループ化できる．これを組み合せると，すべてのコンポーネントを組み合せた分散処理システムとして実現できる．

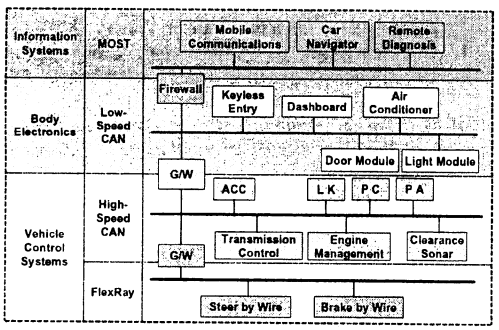

図-10a　Grade A (Luxury)プロダクトライン

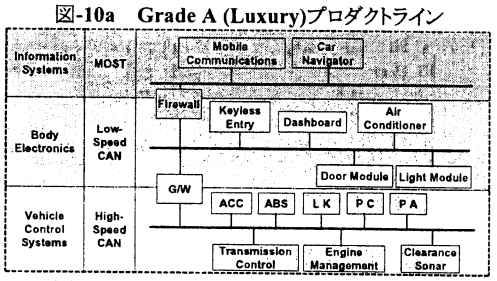

図-10b Grade B (Midsize) プロダクトライン

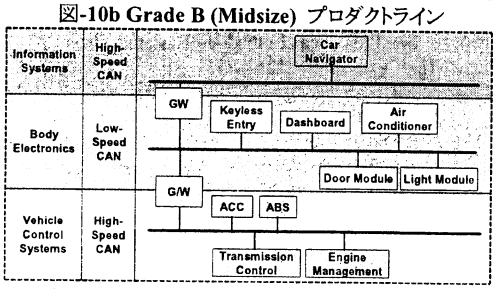

図-10c Grade C (Compact) プロダクトライン

また，アスペクトの分析結果に基づき，各アスペクトのレベルを計算すると図 11 のように表せる．これは，プロダクトラインアーキテクチャを構成する場合に，アーキテクチャが満たすべき安全性，性能，コストの各アスペクトの範囲を示す．このアスペクトレンジを満たしながら，プロダクトライン開発アプローチに基づいてコンポーネントとコネクタ，およびアスペクトを再利用資産として組合せるとプロダクトラインアーキテクチャを構築できる．

ここで，図-10a を Grade A (Luxury class)のシステムアーキテクチャと考えると，アスペクトマッチングを維持し，幾つかのコンポーネントを取り外すことにより，図-10b と図-10c に示す，より廉価な 2 つのプロダクトラインを構成できる．

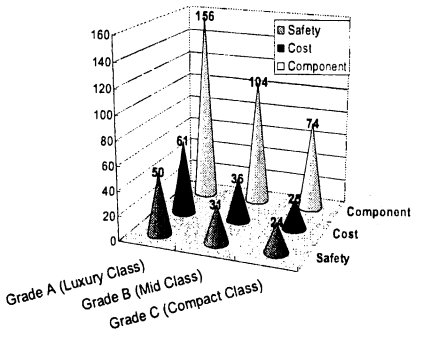

図-11　3 つのプロダクトラインのアスペクトレンジ

5. 本研究の評価

5.1. アスペクト指向モデリング・分析方法論の評価

本稿で提案したアスペクト指向に基づくモデリング・分析方法論を実システムの特性を反映した車載組込みシステムの例に適用した結果，システムの横断的特性である安全性などの相互に干渉する非機能的特性の多元的モジュール化を支援できた．

また，分析の結果であるコンポーネント/コネクタとアスペクトの関係，アスペクトの定量的評価，アスペクト間の関連を再利用資産として利用することにより，実際のシステムと同等のアーキテクチャの導出を支援できた．

これらのことから，本稿で提案する方法論は実践の点で有効であるといえる．

5.2. 従来のアプローチとの比較

本稿では，車載組込みシステムを例として，アスペクト指向に基づき，システムや複数のコンポーネント(ECU)/コネクタ(ネットワークプロトコル)に横断的に関わる複数のアスペクトについて，アスペクトを定量的に評価し，システムとアスペクトおよびアスペクト間の関係を統一的に分析する方法論を提案した．

従来のアプローチでは，以下の問題があった．

(1) 機能要求と非機能要求を分離する方法論が確立されていない[4, 13, 21].

(2) アスペクトを分離しても，複数のアスペクト同士の関係を分析する方法が確立されていない[8, 9, 11, 18, 19].

(3) 特定のアスペクトにのみ注目しており，システム全体との関係，他のアスペクトとの関係を分析する方法については議論されていない[2, 3].

本稿で扱った車載組込みシステムの設計では，システム全体にわたり一定の水準で多元的なアスペクトを

満たさなければならず，従来のアプローチではこの問題の解決が難しかった．

そこで，本研究では，従来のアプローチを統合し，さらに，アスペクトを定量的に評価し，コンポーネント/コネクタとの関連をモデル化，アスペクト間の関連を分析することにより，多元的アスペクトを扱えるモデリング・分析方法論を提案した．

6. 今後の課題

実際には，ネットワーク上で連携する ECU の数は，車種によっては 50 個を超えることがある．また，非機能的特性についても，本稿で扱った例よりも種類が多くなるため，機能要求と非機能要求，非機能要求同士の関係では，多数の組合せが存在する．したがって，実際の開発ではモデル化，分析すべき要素が増大するため，効率化を図るためには，コンピュータによるモデリング，分析の支援あるいは，シミュレーションによる設計支援が不可欠である．

今後，モデリング・分析，設計を一貫して支援できる環境，シミュレータの開発を検討する．

7. まとめ

リアルタイム分散処理システムである車載組込みシステムを対象として，相互干渉や関係するアスペクトをシステム全体でモデル化し，アスペクトマッチングを分析・設計する方法論を提案した．車載組込みシステムを例として試行した結果，安全性，性能，コストをバランスする3種類のプロダクトラインアーキテクチャを導出できた．

本稿で提案した，分散システム全体にわたるアスペクト指向モデルの概念とそのモデリング方法論は，実践的組込みシステムを例とする試行評価により，有効性を確認している．

今後，本稿で提案する方法論，設計のコンピュータ支援などを研究し，実用的な方法論への発展を図る．

謝辞：本研究を進めるにあたり，ご教示とご支援を頂いた，(株)デンソーの村山浩之氏，岩井明史氏，佐藤洋介氏，ならびに，(株)デンソー技研センターの関係各位に感謝する．

参考文献

[1] S. Amberker, et al., A System Safety Process for By-Wire Automotive Systems, *Design and Technologies for Automotive Safety-Critical Systems*, SP-1507, SAE, 2000, pp. 69-74.

[2] M. Aoyama, et al., Embracing Requirements Variety for e-Governments Based on Multiple Product-Lines Frameworks, *Proc. IEEE RE '03*, Sep. 2003, p. 285.

[3] A. Avizienis, et al., Basic Concepts and Taxonomy of Dependable and Secure Computing, *IEEE Trans. Dependable and Secure Computing*, Vol. 1, No. 1, Jan.-Mar. 2004, pp. 11-33.

[4] A. Bondavalli, et al., Dependability Analysis in the Early Phases of UML-Based System Design, *Int'l. J. of Computer Systems Science and Eng.*, Vol. 16, No. 5, Sep. 2001, pp. 265-275.

[5] V. Claesson, et al., The XBW Model for Dependable Real-Time Systems, *Proc. IEEE ICPDS (Int'l. Conf. on Parallel and Distributed Systems)*, Dec. 1998, pp. 130-138.

[6] P. Clements, et al., *Software Product Lines: Practices and Patterns*, Addison Wesley, 2001.

[7] A. Ferrari, et al., System Level Design For Real Time Applications, *MDA for Embedded System Development*, Sep. 2002, http://www.ensieta.fr/mda/ecoleMDA2002/.

[8] R. E. Filman, et al. (eds.), *Aspect-Oriented Software Development*, Addison Wesley, 2005.

[9] G. Georg, et al., Specifying Cross-Cutting Requirements Concerns, *Proc. UML 2004, LNCS, Vol. 3273*, Springer, Oct. 2004, pp. 113-127.

[10] H. Gomaa, *Designing Concurrent, Distributed, and Real-Time Applications With UML*, Addison Wesley, 2000.

[11] J. Grundy, Aspect-Oriented Requirements Engineering for Component-Based Software Systems, Proc. *IEEE RE '99*, Jun. 1999, pp. 84-91.

[12] B. Hedenetz, et al., *Brake-By-Wire without Mechanical Backup by Using a TTP-Communication Network*, SAE Tech. Paper, No. 981109, SAE, 1998.

[13] A. Higashi, et al., Design of Education Program to Practice Requirements Analysis and Requirements Specification for Automotive Software Engineers in DENSO, *Proc. AuRE '04, RE '04 (Int'l Workshop on Automotive Req. Eng.)*, Sep. 2004, pp. 51-56.

[14] IEEE, *Functional Safety of Electrical/ Electronic/ Programmable Electronic Safety-Related Systems – Part 3: Software Requirements*, IEC 61508-3, 1998.

[15] R. Lutz, Analyzing Software Requirements Error in Safety-Critical Embedded Systems, *Proc. IEEE RE '93*, 1993, pp. 126-133.

[16] MISRA, *Development Guideline for Vehicle Based Software*, V. 1.1, 2001.

[17] N. Navet, et al., Trends in Automotive Communication Systems, *Proc. of the IEEE*, Vol. 93, No. 6, Jun. 2005, pp. 1204-1223.

[18] R. Reddy, et al., An Aspect-Oriented Approach to Analyzing Dependability Features, *Proc. AOM '05*, Mar. 2005, http://dawis.informatik.uni-essen.de/events/AOM_AOSD2005/.

[19] P. Tarr, et al. N Degree of Separation: Multi-Dimensional Separation of Concerns, *Proc. ICSE '99*, May 1999, pp. 107-119.

[20] US DoD, *System Safety Program Requirements*, MIL-STD-882D, 2000.

[21] M. Weber, et al., Requirements Engineering in Automotive Development: Experience and Challenges, *IEEE Software*, Vol. 20, No. 1, Jan./Feb. 2003, pp. 16-24.