

自動車ソフトウェアにおけるサービス指向アーキテクチャの提案

2005MT029 生駒 光平 2005MT055 近藤 貴大 2005MT115 田邊 隼希

指導教員 青山 幹雄

1. はじめに

複数のベンダで開発した自動車ソフトウェアのネットワークを介した協調制御が必要となっている[2]。本稿では、プロパティに基づき従来のサービス指向アーキテクチャ(SOA: Service-Oriented Architecture)の直接的なサービス提供に加え、イベント起動や間接的なサービス提供を含むモデルとそれを実現するアーキテクチャを提案する。

2. SOA 適用の現状と問題点

自動車ソフトウェアにおいてセントラルロックシステム[4]を例として SOA を適用する研究がある。SOA の適用によって、メッセージベースの処理が可能となる。しかし、従来の SOA には以下の問題がある。

(1) イベントの処理が未対応

自動車などでは、イベント処理が重要であるが、従来の SOA ではイベントの処理を扱えない。

(2) 空間を介したサービス提供の必要性

自動車などでは、ユーザが直接結果を受けるサービスと、空間を介してユーザが効果を得るサービスも存在する。空間を介したサービス提供の例を図 1 に示す。



図 1 空間を介したサービス提供

3. プロパティ中心サービス提供モデル

3.1. プロパティ中心サービス提供モデル

SOA を適用する上でプロパティ中心にモデル化を行う。プロパティを次の三つに分類する。

- (1) サービスプロパティ: 動的に変化するサービスの特性や状態。
- (2) 環境プロパティ: ユーザを取り巻く環境の状態。
- (3) ユーザプロパティ: ユーザの特性、システムへの要求。

3.2. 自律的サービス提供と協調的サービス提供

図 2 に示すように、自動車ソフトウェアのサービス提供における相互作用をその性質に着目し分類する。

- (1) 自律的サービス提供: 各機器は、それを取り巻く環境プ

ロパティが目標値に収束するように自身を制御する。目標値とはユーザの要求を示す。

- (2) 協調的サービス提供: 複数のサービスを協調動作させることで全体としてユーザの要求を満たす。

これらの異なる制御モデルのアーキテクチャは異なり、統一性を欠くことから、振舞いの一貫性の欠如や制御が競合する可能性があるため、これらを統合できる SOA の拡張とその設計方法が必要となる。

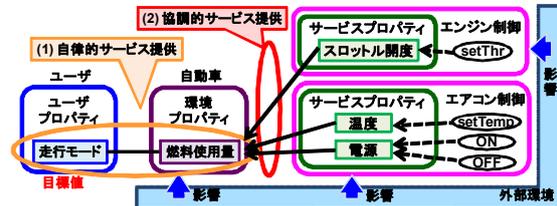


図 2 自律的サービス提供と協調的サービス提供

4. モデルに基づく SOA 設計方法

サービス提供を様々な観点から分類し、そのサービス提供を実現するアーキテクチャパターンを決定する。

4.1. 対話型サービス提供と自律型サービス提供

サービスが制御するプロパティの性質とユーザとの相互作用の違いに着目し、分類する。

- (1) 対話型サービス提供: ユーザの要求に対しプロパティを動的に変更する。
- (2) 自律型サービス提供: 操作対象のプロパティ値を一定に保つためシステムが自律的にサービス提供を行う。

4.2. 直接的サービス提供と間接的サービス提供

ユーザとサービス間の相互作用に着目し、分類する。

- (1) 直接的サービス提供: サービスとユーザが直接相互作用する。
- (2) 間接的サービス提供: 環境プロパティを制御する場合、サービスは環境を介してユーザに影響を与える。

4.3. アーキテクチャパターンの決定と統合

対話型サービス提供では、ユーザはスイッチ操作などでサービスプロバイダを直接指定、起動し、その結果を待つ。従ってメッセージベースの要求応答を用いる。

自律型サービス提供では、環境の情報をセンサなどでイベントとして検出し、検出内容と関連するサービスを起動する。環境情報の発生とそれに応じた処理を独立に行うためイベントベースのパブリッシュ/サブスクライブを用いる。

この二つのアーキテクチャパターンは振舞いが異なるが、システム全体としては統一的に制御する必要がある。この

ため、ブローカを用いてこれらのアーキテクチャパターンの振舞いを統合する。

5. ブローカアーキテクチャ

メッセージベース、イベントベースの異なるモデルを統一的に制御するブローカアーキテクチャを提案する。

5.1. 導入するブローカ

ブローカをイベント処理とサービス起動に分離する。イベント処理をイベントマネージャ、サービス起動をサービスコーディネータが行う(図3)。

- (1) イベントマネージャ: センサからのイベントを受信し、登録されたサブスクリプションに基づきフィルタリングを行い、サービスコーディネータにイベントを配信。
- (2) サービスコーディネータ: ユーザ、サービスからの要求やイベントマネージャとの連携でサービスを起動。

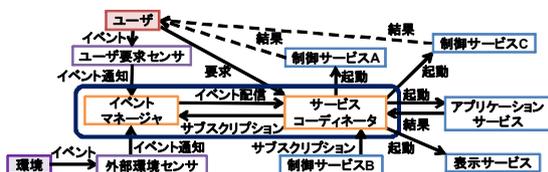


図3 ブローカアーキテクチャ

5.2. ブローカアーキテクチャの振舞い

ブローカアーキテクチャの振舞いを対話型、自律型複合ユースケースにより検証する(図4)。

対話型、自律型サービス提供パターンという制御モデルの異なるサービス提供パターンを実現し、サービス提供を統一的に制御できた。またパブリッシュ/サブスクライブを用いて、イベントを複数のサービスに通知できる。

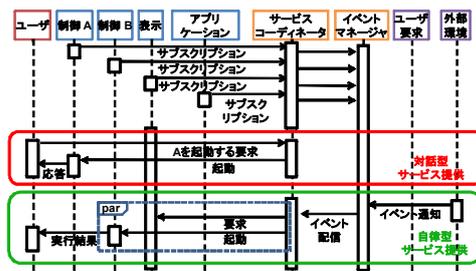


図4 ブローカアーキテクチャの振舞い

6. ブローカアーキテクチャの適用例

ブローカアーキテクチャを ACC(Adaptive Cruise Control System)に適用し、分析する(図5)。



図5 ACCの分析プロセス

(1) 機能分析

- a) ACC の機能をユースケース図を用いて分析し、構成

するセンサ、サービスを決定する。

- b) ユースケース図に基づき、ユースケース記述を作成し、シナリオによりサービス提供パターンを発見する。

(2) 相互作用分析

- a) シナリオからシーケンス図を作成し、実行順序に着目した相互作用分析をする。
- b) シーケンス図からコミュニケーション図を作成し、要素間の相互作用分析をする。

6.1. 機能分析

(1) ユースケース分析

ACC の提供機能とアクタの関係を示す(図6)。アクタは、ユーザ、レーダ、車速センサ、エンジン制御サービス、ブレーキ制御サービスである。メインスイッチの操作はユーザの操作なのでユースケース図には示さない。ACC は距離計算、追従制御、速度制御、ブレーキ制御に分割できる。

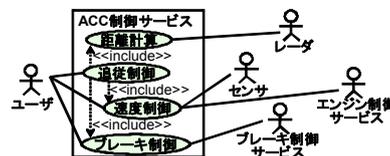


図6 ACCのユースケース図

(2) シナリオ分析

図6のユースケース図に基づき、ACCの開始から追従走行制御までのシナリオを作成した。ユーザがサービスを指定して要求を行いサービスが起動する対話型サービス提供パターンと、環境プロパティの変化からサービスが起動する自律型サービス提供パターンがある。

6.2. 相互作用分析

(1) 実行順序に着目した相互作用分析

ACCの相互作用を分析するために、シナリオに基づきシーケンス図を作成した(図7)。シーケンス図から、次の二つのサービス提供パターンを発見できた。

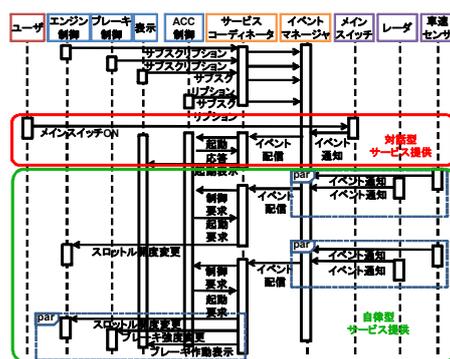


図7 ACCのシーケンス図

- 1) 対話型サービス提供パターン: ユーザ要求により ACC 制御サービスが起動。
- 2) 自律型サービス提供パターン: 環境プロパティの変化により ACC 制御サービスが起動され、プロパティの値を保持するために制御。

ACC 制御サービスはコンテキストに応じてエンジン制御サービスとブレーキ制御サービスの協調制御を行う。

(2) 要素間の関係に着目した相互作用分析

図7のシーケンス図より、コミュニケーション図を作成し要素間の相互作用を分析した(図8)。

イベントはイベントマネージャに通知され、サービスコーディネータに配信される。サービスコーディネータはイベントやサービスの要求に基づきサービスを起動する。サービスはサービスコーディネータを介して起動され、サービス間で直接メッセージのやり取りはしない。ブローカーアーキテクチャの適用でイベント処理とサービス起動を分離できた。

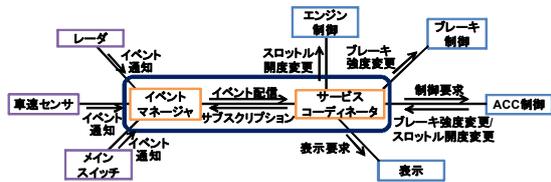


図8 ACCのコミュニケーション図

7. ACCによるアーキテクチャの評価

提案するアーキテクチャの評価のために、6章の分析プロセスにより従来のシステム、従来の SOA から構成されるシステムを分析、比較を行う。対象システムはACCとする。

7.1. 従来ACCシステムの分析

7.1.1. 機能分析

(1) ユースケース分析

ACCは追従制御、速度制御、ブレーキ制御という三つの機能から構成される。アクタは仕様[5]に基づきユーザ、レーダ、ディスタンスコントロールコンピュータ、車速センサ、エンジンコントロールコンピュータ、スキッドコントロールコンピュータとした。追従制御にはメインスイッチも関係するがユーザの操作なのでユースケース図には示さない。

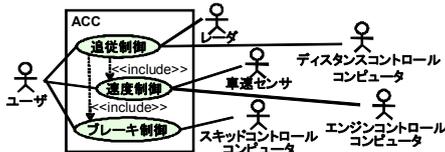


図9 従来ACCシステムのユースケース図

(2) シナリオ分析

図9のユースケース図に基づきACCの開始から追従走行制御までのシナリオを作成した。

7.1.2. 相互作用分析

(1) 実行順序に着目した相互作用分析

従来ACCシステムの相互作用分析のためにシナリオに基づき、シーケンス図を作成し、分析した(図10)。

(2) 要素間の関係に着目した相互作用分析

図10のシーケンス図より、コミュニケーション図を作成し要素間の相互作用を分析した(図11)。

コンピュータとコンピュータが直接制御を行っており、コ

ンピュータ間の結合が強い。またセンサとコンピュータも直接イベント通知を行っているため、この二つの要素間でも結合が強い。

従って新しい機能、センサの追加や変更を行う際には、システムの大規模な変更が必要となる。センサのイベントはあらかじめ指定されたコンピュータにしか配信できず、イベント配信は固定的である。よってセンサのイベントが追加されたコンピュータで必要な場合に対応することが難しい。

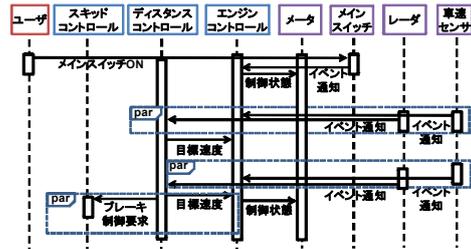


図10 従来ACCシステムのシーケンス図

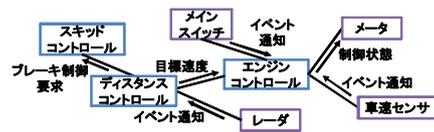


図11 従来ACCシステムのコミュニケーション図

7.2. 従来のSOAによるACCの分析

機能分析に関しては、6.1節の分析結果を用いる。

7.2.1. 相互作用分析

従来のSOAによるACCの相互作用分析のために6.1節のシナリオよりシーケンス図を作成し、分析した(図12)。

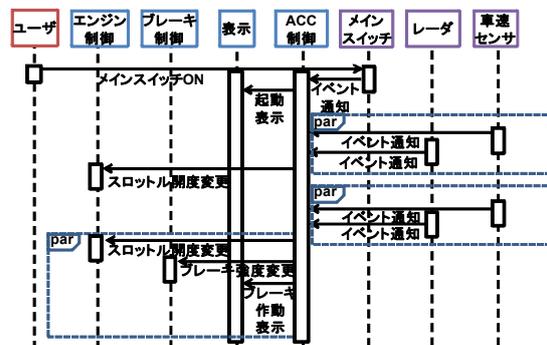


図12 従来のSOAによるACCのシーケンス図

車速センサ、レーダのイベントがACC制御サービスに通知されているが、従来のSOAではイベントを扱えない。またエンジン制御サービスやブレーキ制御サービスの実行結果は間接的にユーザに作用し、従来のSOAの直接的なサービス提供形態とは異なる。

7.3. 提案アーキテクチャの評価

従来のシステム、従来のSOAと提案アーキテクチャをSOA、イベント、プロパティという三つの観点から比較を行い、評価した(表1)。

表 1 ACC によるアーキテクチャの評価

評価の観点	評価項目	従来のシステム	従来の SOA	提案アーキテクチャ
SOA	インタフェース形式	ベンダごとに異なる	標準化されている	標準化されている
	プラットフォームへの依存性	依存	非依存	非依存
イベント	イベントの処理	可能	不可能	可能
	イベントの配信制御	固定的	イベントの処理までできない	追加・変更が可能
プロパティ	サービス提供形態	サービスは存在しない	直接的サービス提供	直接的・間接的サービス提供

(1) インタフェース形式

従来のシステムの場合、インタフェースがベンダごとに異なりソフトウェアの連携が困難であった。提案アーキテクチャや従来の SOA ではインタフェースの標準化により、サービスの再利用性が高まり、車内のサービスとテレマティクスサービスの連携も容易となる。

(2) プラットフォームへの依存性

従来のシステムはプラットフォームに依存しソフトウェアの連携が困難である。一方、従来の SOA や提案アーキテクチャはプラットフォームに非依存で、プラットフォームが異なるベンダ間で開発されたサービスの連携が容易である。

(3) イベントの処理

自動車ソフトウェアはイベントにより制御を行うが、従来の SOA ではイベントを扱えない。従来のシステムはイベント処理が可能である。提案アーキテクチャでもイベントマネージャとサービスコーディネータの連携により、イベント処理が可能である。

(4) イベントの配信制御

従来のシステムではセンサとコンピュータ間で直接イベント通知を行い、センサのイベント配信先は固定的である。提案アーキテクチャはパブリッシュ/サブスクライブを用いてイベント配信先の追加と変更が可能となり、またイベントを複数サービスに配信できる。

(5) サービス提供形態

ACC ではサービスの実行結果が間接的にユーザに作用し、従来の SOA では対応できない。提案アーキテクチャではプロパティに着目し制御を行うため、間接的なサービス提供が可能になる。

以上の議論より、提案アーキテクチャは SOA に基づく自動車ソフトウェアの実現に有効なアーキテクチャといえる。

8. 考察

8.1. ブローカーアーキテクチャの妥当性

サービス提供を対話型、自律型に着目し、この分類からアーキテクチャパターンを決定した。自律型サービス提供にパブリッシュ/サブスクライブを用いることで同時多発的イベントに対して適切なサービスの提供が可能となる。

8.2. サービスの再利用性

図 13 に ACC と CC(Cruise Control System)のユースケ

ース図を示す。この図から ACC は CC に距離計算、追従制御とレーダの追加で実現可能であることがわかる。既存の機能に新機能を追加すれば一から開発を行う必要が無く、再利用により開発期間の短縮やコスト削減が期待できる。

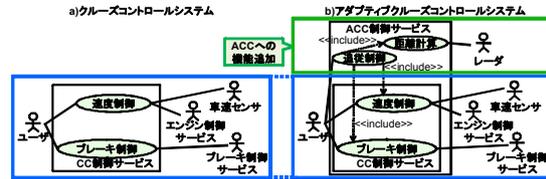


図 13 サービスの再利用

9. 関連研究

9.1. 組込みネットワークへの SOA 適用

組込みネットワークシステムにおいて SOA を拡張する研究が行われ、ホームネットワークシステム (HNS: Home Network System) に適用されている[1].

9.2. AUTOSAR

AUTOSAR では、ネットワーク上でソフトウェアを連携することを主眼とし、ECU ソフトウェアに関する技術の研究が行われている[3]. しかし、ネットワークを介したアプリケーション間でのインタフェースの標準化は行われていない。

10. 今後の課題

本稿で検証した ACC は間接的サービス提供に限られ、直接的サービス提供を含むシナリオを検証する必要がある。また、自動車ソフトウェアの SOA ではタイミング協調やリアルタイム性を考慮する必要がある。

11. まとめ

従来の SOA の直接サービス提供に加え、イベント起動と空間を介したサービス提供を含むプロパティ中心サービス提供モデルを提案した。ブローカを用いたアーキテクチャを提案し、シナリオを用いて有効性を評価した。

参考文献

[1] 青山 幹雄, 藤山 麻衣, 組込みネットワークシステムのユニバーサルサービス指向アーキテクチャ, SES 2008 論文集, Sep. 2008, pp. 147-154.
 [2] 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, 自動車技術学会秋季論文集, 2008.
 [3] AUTOSAR, <http://www.autosar.org>.
 [4] I. H. Kruger, et al, Service-Based Software Development for Automotive Applications, Proc. Convergence 2004, No. 2004-21-0040 CTE, 2004.
 [5] トヨタ自動車, CROWN MAJESTA 新型車解説書, 2004.