

# 自動車ソフトウェアの 外部連携サービス指向アーキテクチャの提案

M2011MM030 伊藤 智基

指導教員 青山 幹雄

## 1. はじめに

自動車テレマティクスサービスではサードパーティサービスとの連携が進んでいる。今後、オープンかつシームレスな連携が必要となる[1]。しかし、サービス連携においてプロバイダ毎に開発環境が異なり、サービス間のインタフェースが固定的となっている。また、車載ソフトウェアにはリソース制約や車内環境といった様々な制約があることから、指定時間内のサービス利用が困難となっている。

## 2. 研究の課題

本稿の課題として以下の二つを挙げる。

### (1)プラットフォーム依存

車外サービスではサービスを提供するプロバイダ毎に開発環境が異なり、特定の車種やベンダでサービス間のインタフェースが固定的となっている。よって、異なるプラットフォーム上での車載ソフトウェア機能の再利用や機能追加、テレマティクスサービスの利用といったサービス連携が困難となっている。

### (2)イベント配信のタイミング

車載ソフトウェアには様々な制約がある。そのため、車内外で発生するイベントやメッセージを処理し、制御対象が指定した要求時間内にイベント配信を行い、利用することが困難となっている。また、発生イベントやメッセージは指定時間以前のタイミングに通知されても意味をなさない。

## 3. 関連研究

本稿と関連する研究として以下の三つを挙げる。

### 3.1. Publish/Subscribe Architecture

相互作用の疎結合の形態を提供する対話方式である[2]。パブリッシャは特定のサブスクライバを想定せずにメッセージを送信する。サブスクライバは、ブローカにサブスクリプションを送信し、関心のあるメッセージのみ受信する。また、パブリッシャとサブスクライバは相互のエンドポイントを知る必要がなく、空間的に分離される。

### 3.2. WS-Notification

Web サービス間の非同期メッセージングを Publish/Subscribe 型で実現する技術である。WS-Notification はいくつかの仕様から構成されており、基本的な部分は WS-BasedNotification[3]で規定され、メッセージを分類する

Topic の構造を規定した WS-Topics, NotificationProducer と NotificationConsumer の間に仲介する NotificationBroker の役割を規定した WS-Brokered Notification の三つの仕様からなる。

### 3.3. 車載イベント駆動アーキテクチャの提案[4]

車内外で発生するイベントをモデル化し、車載ブローカを導入することでイベントを共有するアーキテクチャが提案されている。SOA(Service-Oriented Architecture)の適用と、イベントモデルに基づくイベント配信の統一化によって車載イベントブローカのサービス化を行っている。

## 4. アプローチ

本稿では以下の2点に着目し、オープンかつシームレスな連携を実現する(図1)。

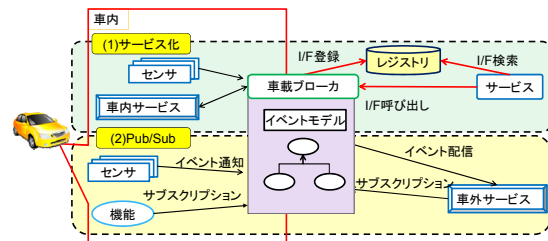


図1 アーキテクチャの全体像

### 4.1. 車内外システムのサービス化

車内外のシステムをサービスとみなすことでオープンな連携を実現する。車内システムと外部システムの機能を標準インタフェースを介して提供するサービスととらえる。これにより、特定の車種やプラットフォームに依存しない外部連携が実現可能となる。

### 4.2. Publish/Subscribe Architecture の適用

Publish/Subscribe モデルに基づく車載ブローカを導入することでシームレスな連携を実現する。コンシューマは車載ブローカにサブスクリプションを登録する。サブスクリプションの条件として通知タイミングを指定する。これにより、車内外サービスは指定したタイミングにイベント通知を受け取ることが可能となる。また、車内外では多くのイベント配信が行われている。Publish/Subscribe Architecture の空間的分離により相互のエンドポイントを知る必要はなく、多対多のイベント配信が可能となる。



### (1) イベントマネージャ

イベントマネージャはイベント情報の管理を行うコンポーネントである。プロデューサからのイベント通知をイベントモデルに基づきイベントデータベースに登録する。また、Subscription マネージャからサブスクリプションの照合をする際には、イベント情報をイベントデータベースに対して問い合わせを実行し、参照する。

### (2) タイミングマネージャ

タイミングマネージャはイベント配信のタイミングなど、イベントにタイミング条件を付加し制御するコンポーネントである。サブスクリプションデータベースからコンシューマがサブスクリプション条件として登録した要求時間を参照する。通知型イベントに対しては、イベント情報に通知タイミングを付加することでタイミングを制御し、キューに挿入される。

### (3) Subscription マネージャ

Subscription マネージャはコンシューマからのサブスクリプションを管理するコンポーネントである。コンシューマからのサブスクリプションをサブスクリプションデータベースへ登録する。また、サブスクリプションの更新や削除が行われた場合にもデータベースへアクセスし管理する。その際、サブスクリプションの更新やキャンセルなどを行う。また、フィルタリングを行いイベントを配信する。

#### 5.2.3. 車載ブローカの振舞い

イベント配信時のブローカの振舞いを図 5 に示す。

Subscription マネージャ、タイミングマネージャを用いて、(1) イベント登録、(2) サブスクリプション登録、(3) イベント配信の三つに分けて示す。

#### (1) イベント登録の振舞い

プロデューサはイベント情報として、イベント ID とタイミングレベルの登録を行う。タイミングレベルとは時間制約のことであり、制約が厳しい処理を優先的に実行することで時間制約を達成する。例えば、カーナビゲーションシステムにおいて、「GPS からの位置情報処理」は「ルート表示」よりも時間制約が厳しく、優先的に処理される必要がある。

#### (2) サブスクリプション登録の振舞い

コンシューマから Subscription マネージャにサブスクリプションが登録されると、その条件を基にイベント照合を行う。

Subscription マネージャからイベントマネージャに照合依頼が通知され、イベントとサブスクリプション条件が一致した場合にイベント配信を行う。また、サブスクリプションの有効期限切れなどサブスクリプションに変更があった場合などは、コンシューマから再び Subscription マネージャに対して変更内容を通し、サブスクリプションの更新を行う。

#### (3) イベント配信の振舞い

リクエスト型イベントの場合、要求時間をあらかじめ保持しているため要求時間に応じてイベントディスパッチャにイベントが通知される。通知型イベントの場合、イベント発生時刻は保持されているが要求時間は保持していない。その

ためタイミングマネージャがサブスクリプション条件に基づき、要求時間の情報を付加してイベントディスパッチャにイベントが通知される。

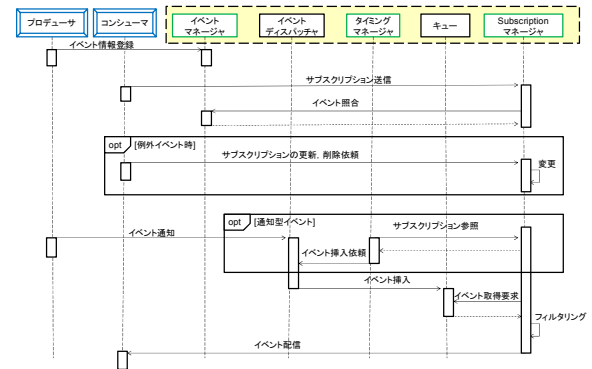


図 5 車載ブローカの振舞い

## 6. テレマティクスサービスへの適用

本稿で提案した外部連携アーキテクチャ内において、イベント配信のタイミング管理がテレマティクスサービスの例題で対応可能か確認する。テレマティクスサービスの例題としてオートアラームシステム[5]とウォーニング通知[5]に適用し、評価を行う。

### 6.1. オートアラームシステムへの適用

オートアラームシステムでのユースケース分析の結果から抽出した機能は「センサ状態送信サービス」、「オートアラームセットサービス」、「アラーム通知サービス」、「車内検知サービス」の四つである。各サービスとアクタの関係を以下に表 1 にまとめ、提案を適用した際の振舞いを図 6 に示す。

表 1 オートアラームの構成サービスとアクタの関係

サービス名	対応アクタ
センサ状態送信サービス	プロデューサ
オートアラームセットサービス	プロデューサ、コンシューマ
アラーム通知サービス	コンシューマ
車内検知サービス	コンシューマ

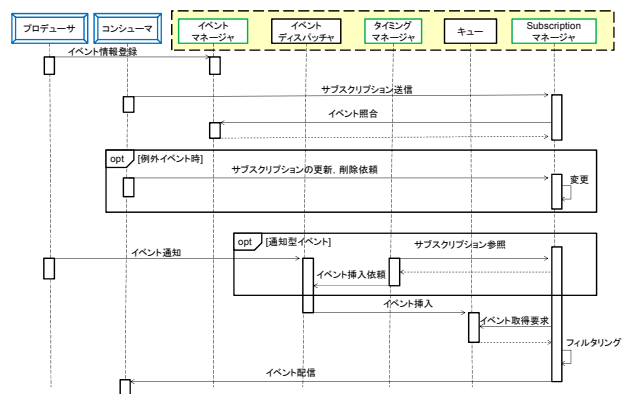


図 6 オートアラームにおける車載ブローカの振舞い

## 6.2. ウォーニング通知システムへの適用

ウォーニング通知システムでのユースケース分析の結果から抽出した機能は「センサ状態送信サービス」、「ウォーニング通知サービス」、「車両情報通知サービス」、「取り次ぎメール送信サービス」の四つである。各サービスとアクタの関係を表2にまとめ、提案を適用した際の振舞いを図7に示す。

表2 ウォーニング通知の構成サービスとアクタの関係

サービス名	対応アクタ
センサ状態送信サービス	プロデューサ
ウォーニング通知サービス	プロデューサ, コンシューマ
車両情報通知サービス	プロデューサ, コンシューマ
取り次ぎ送信サービス	コンシューマ

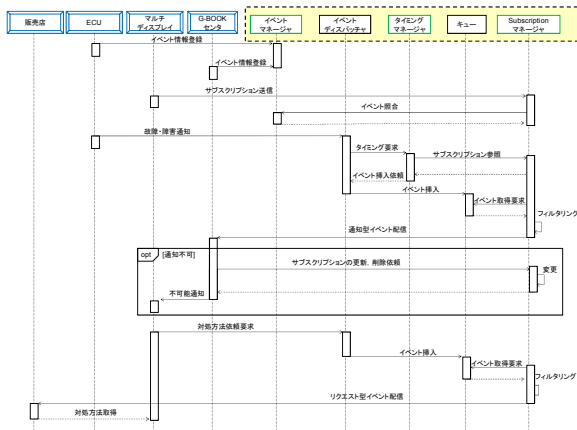


図7 ウォーニング通知における車載プロローカの振舞い

## 7. 提案アーキテクチャの評価

提案アーキテクチャをテレマティクスサービスの例題に適用した結果を用いて、多様なイベントへの対応、イベント配信のタイミングの観点から評価する。

### (1) 多様なイベントへの対応

車内外では多くのセンサやサービスからのイベント配信が行われている。例題ではセンサからのイベント通知やサービスからのイベント通知が行われている。それぞれがプロデューサ、コンシューマ、または両方の可能性がある。リクエストを期待するアクタ、リクエストを期待しないアクタと多様である。オートアラームシステムでは通知型のイベント配信のみの処理である。ウォーニング通知では、リクエスト型と通知型の両方からのイベント配信が行われている。イベントモデルを用い、イベント照合を行うことでいずれのケースでもイベントを識別することが可能となり、それぞれのイベントに合った処理を行うことが可能となった。

### (2) イベント配信のタイミング

センサなど時刻情報しか保持しないイベントに対して、

コンシューマのサブスクリプションの条件で指定した要求タイミングをタイミングマネージャが付加することで、コンシューマの要求時間内のイベント配信が可能となった。サービスなどあらかじめ時間情報を保持しているイベントに対しては、要求時間を付加する必要はない。サービスのサブスクリプションの条件に従い、Subscription マネージャがタイミングの管理を行うことで配信が可能となった。

また、例題では通知タイミングが変更された場合の処理として、サブスクリプションの変更などを行っている。これによりコンシューマの要求時間や通知タイミングの変更に対し、サブスクリプションの更新を反映することでイベント配信のタイミングを変更することで対応可能となった。

従って、センサやサービスなど保持する情報が異なるアクタからのイベント通知を統一的に管理、変更、削除を行うことによる、統一的なイベント配信が可能となった。

## 8. 今後の課題

### (1) プロトタイプを実装しての検討

実際にプロトタイプを作成し、サービスが指定したタイミング以内に利用可能であるかを検証する必要がある。

### (2) 例外処理の検討

例外イベントなどが発生した場合の例外処理について考慮されていないため検討する必要がある。その後、イベントを再送信する処理についても考慮する必要がある。

## 9. まとめ

本稿では、外部連携サービス指向アーキテクチャを提案した。提案アーキテクチャでは、車内外システムをサービスとみなし、標準技術を用いることでオープンな連携を提案した。シームレスな連携を実現するために Publisher/Subscribe モデルに基づく車載プロローカを導入し、イベント配信のタイミング管理を行った。提案アーキテクチャをテレマティクスサービスに適用し、有効性を示した。

## 参考文献

- [1] 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 2008 年秋季大会 学術講演会前刷集, No. 97-08, Oct. 2008, pp. 21-26.
- [2] P. T. Eugster, et al., The Many Faces of Publish/Subscribe, ACM Computing Survey, Jun. 2003. pp. 114-131.
- [3] OASIS, Web Services Base Notification 1.3 (WS-BaseNotification), 2006, [http://docs.oasis-open.org/wsn/wsn-ws\\_base\\_notification-1.3-spec-os.pdf](http://docs.oasis-open.org/wsn/wsn-ws_base_notification-1.3-spec-os.pdf).
- [4] 武野 佑基, ほか, SOA に基づく車載イベント駆動アーキテクチャの提案, 情報処理学会 第 74 回全国大会 講演論文集, Vol. 1, Mar. 2012. pp. 369-370.
- [5] トヨタ自動車, Lexus LS 電子技術マニュアル, 2006.