車載ソフトウェアのための イベント駆動サービス指向アーキテクチャの提案と評価

宮脇 聡[†1]　中道 上[†2]　青山 幹雄[†2]　佐藤 洋介[†3]　岩井 明史[†3]

南山大学大学院 数理情報研究科[†1]　南山大学 情報理工学部 ソフトウェア工学科[†2]　株式会社デンソー[†3]

1. はじめに

近年，車載ソフトウェアの再利用や拡張，ネットワークを介した連携を柔軟に行うためにサービス指向アーキテクチャ(SOA)の適用が研究されている[1]．本稿では，SOA を車載ソフトウェアへ適用するためにイベント処理を可能にするイベント駆動 SOA を提案する．

2. 車載ソフトウェアへの SOA 適用の課題

従来の SOA では，イベント処理とサービス処理が統合的に扱われていない．さらに，車載システムの多様なイベントへ対応する必要もある．

3. 関連研究

SOA の基盤技術である Web サービスにおいてイベントを通知する方法が定義されている[2]．しかし，イベント処理は検討されていない．

分散イベントシステムで共通のイベントを利用するために，タイプ定義を用いたイベントモデルが提案されている[3]．

4. 問題解決へのアプローチ

SOA の要求/応答型の処理にイベント処理を可能とするパブリッシュ/サブスクライブ型の処理を統合する．このため SOA の直接サービス起動とパブリッシュ/サブスクライブによる間接サービス起動の統一的な制御を行うためにサービスブローカを提案する．車載システムで発生するイベントをイベントモデルとして定義し，イベントの識別を可能にする．

5. イベント駆動 SOA の提案

イベント駆動 SOA は，イベントの情報を定義したイベントモデルとイベントを処理するサービスブローカから構成される．

5.1. アーキテクチャの構成要素

図 1 にイベント駆動 SOA のアーキテクチャを示し，処理の流れを説明する．センサは，検出情報をイベントとしてサービスブローカへ送信する．サービスブローカは，イベントをフィルタリングし，イベントをサービスへ配信する．サービスは，受信したイベントに基づきアクチュエータを制御する．

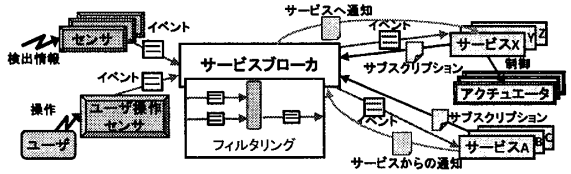

図 1 イベント駆動 SOA のアーキテクチャ

5.2. イベントモデル

イベントモデルは，次の 2 つのモデルから成る．

(1) イベント構造モデル：イベントを構成するイベントソースなどの構造の情報を示す．

(2) イベント発生モデル：イベントの発生するタイミングを示す．

イベントモデルに基づいてイベントの登録やサブスクリプションを行う．

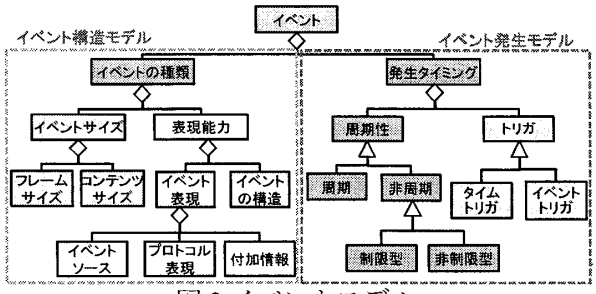

図 2 イベントモデル

5.3. サービスブローカの構造

図 3 に示すようにサービスブローカは，通知情報管理を行う PublisherRegistration マネージャ，イベント処理を行う Subscription マネージャ，サービスへのイベントなどの配信を行う Service マネージャから構成する．

(1) PublisherRegistration マネージャ：センサからのイベントの付加情報やサービスからの通知情報を管理する．イベントの付加情報は，イベントモデルに基づいて定義する．Subscription マネージャ，Service マネージャからのサブスクリプションを通知と照合する．

(2) Subscription マネージャ：ユーザ操作と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イベントのフィルタリングを行い，Service マネージャへイベントを配信する．ユーザ操作のイベントは，対象サービスへ通知される．イベントに関するサブスクリプシ

Event-Driven Service-Oriented Architecture Its Evaluation for In-Vehicle Software.

†1 Satoshi Miyawaki, Graduate School of Mathematical Sciences and Information Engineering, Nanzan University.

†2 Noboru Nakamichi, Mikio Aoyama, Department of Software Engineering, Nanzan University

†3 Yosuke Sato, Akihito Iwai, DENSO CORPORATION

ョンの管理も行う．
(3) Service マネージャ：Subscription マネージャからのイベントを受信し，それをサービスへ通知する．サービスに関するサブスクリプションを管理する．サービス連携では，サービスからの通知をフィルタリングし，他のサービスへ通知する．また，サブスクリプション変更要求を受信し，サブスクリプションの再開/停止を行う．

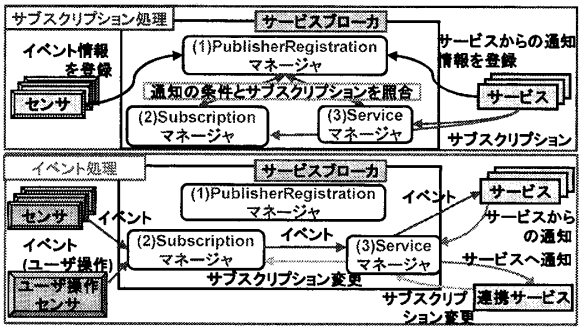

図 3 サービスブローカの構造

5.4. サービスブローカのイベント処理モデル

センサから発生する複数のイベントを優先度に基づきサービスブローカで処理する必要がある．これをイベント処理モデルで実現する(図 4)．イベントディスパッチャがイベントの優先度を判断し，優先度キューにイベントを挿入する．このイベントディスパッチャによりイベントの受信とイベントに基づく処理を分離可能である．

イベント処理モデルを用いてセンサからのイベント送信からサービスがイベント通知を受信するまでの流れを図 5 に示す．

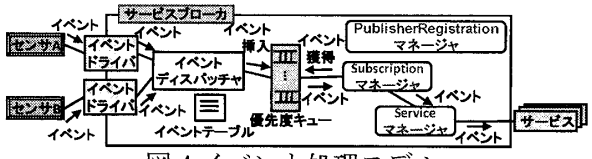

図 4 イベント処理モデル

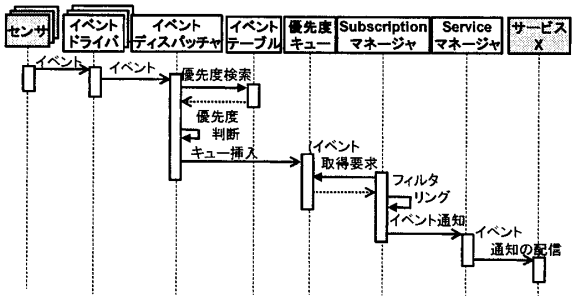

図 5 イベント処理の流れ

6. プロトタイプによるイベント処理の確認

図 4 に示した提案アーキテクチャのイベント処理モデルのプロトタイプを Apache Axis2 を用いて試作した．イベントが同時，並行して発生し，サービスへ通知するまでのサービスブローカのイベント処理を確認した．

7. 提案アーキテクチャの適用と評価

7.1. 提案アーキテクチャの例題への適用

提案アーキテクチャを自動車の CC (Cruise Control System)と ACC (Adaptive Cruise Control System)へ適用した(図 6)．

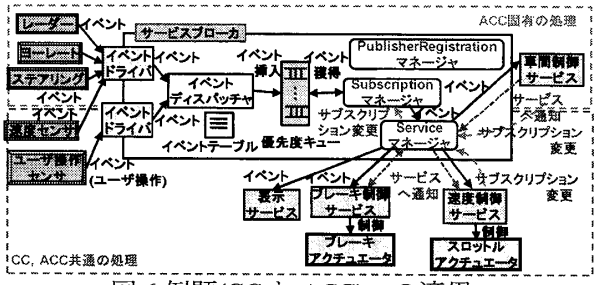

図 6 例題(CC と ACC)への適用

7.2. 評価

提案アーキテクチャが車載ソフトウェアのイベントに対応可能であるかを評価した．

(1)再利用性： CC から ACC へ拡張した場合，追加されたセンサとサービスは，イベント登録とサブスクリプションのみにより利用可能となる．CC のサービスを変更することなく ACC で再利用可能である．

(2)イベント(ユーザ操作とセンサ)の統一制御：速度センサなどのイベントとブレーキ要求などのユーザ操作による処理を Subscription マネージャで行うことで，2 つの処理を統一的に制御が可能となる．

(3)多様なイベントへの対応：イベントモデルを用いることで，レーダーセンサによる複数の異なるイベントを識別可能にする．またイベント処理モデルにより異なるタイミングで発生するイベントを統一的に処理可能となる．

8. まとめ

イベント(ユーザ操作とセンサ)の処理を統一的に制御するサービスブローカとイベントの情報を定義するイベントモデルから構成されるイベント駆動 SOA を提案した．プロトタイプを実装し，イベント処理を確認した．さらに提案アーキテクチャを例題へ適用し，妥当性を確認した．

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, Oct. 2008, No. 97-08, pp. 21-26.

[2] OASIS, Web Service Notification (WSN), ver. 1.3, 2006,http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsn.

[3] S. Rozsnyai, et al., Concepts and Models for Typing Events for Event-Based Systems, Proc. DEBS '07, Jun. 2007, pp. 62-70.