

自動車ネットワークサービスのサービス連携アーキテクチャの提案と評価

M2008MM022 水谷 拓人
指導教員：青山 幹雄

1. はじめに

近年、サービスの再利用、連携を容易にする技術としてサービス指向アーキテクチャ(SOA: Service-Oriented Architecture)が注目されている。そこで、車載システムと外部システムをネットワークを介してサービスとして連携し、新たな機能を提供することが求められている[1]。

本稿では、車載システムが外部システムとネットワークを介したサービス連携可能なアーキテクチャを提案する。

2. 自動車ネットワークのサービス連携

自動車ネットワークサービスのサービス連携には図1で示すように車載システム内の連携と外部システムとの連携の2種類が存在する。車載ネットワーク内のシステム連携を車内サービス連携、車載システムと外部システムのサービス連携を車外サービス連携とする。

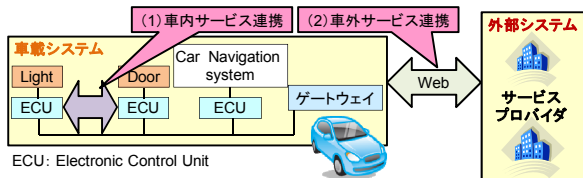


図 1: サービス連携

(1) 車内サービス連携

車内サービス連携とは、自動車の機能を制御している車載システム内の ECU 間のみでサービス連携が行われることである。

(2) 車外サービス連携

車外サービス連携とは、自動車の外に存在するシステムと車載システムとが連携することである。外部システムが車載システムのサービスを利用する場合と車載システムが外部システムのサービスを利用する場合の2つの形態がある。

3. 車外サービス連携の問題

現状の車外サービス連携の問題点と車外サービス連携へSOAを適用する上での問題点として以下の2点が挙げられる。

(1) サービスインタフェースとプロトコルが固有

現状での車外サービス連携を行う場合、サービスを提供するベンダやテレマティクスサービスプロバイダごとにインタフェース、通信プロトコルが固有である。そのため、サードパーティサービスの利用が困難であり、異なるサービス間のサービス連携も困難である。そのため、本稿では車外サービス連携にSOAの技術を適用することで複数の車外サービスと連携を可能にする。

(2) 組込みシステムの処理能力の制約

現状の車載ECUでは処理能力の制約が厳しいためWebサービスで用いられるメッセージを変換せず、個別の車載ECUやネットワークへ送信することはできない。また、新た

にWebサービスのメッセージプロトコルに対応する車載ECUやネットワークを個別に提供することは組込みシステムの処理能力やリソースの制約があるため困難である。

4. 関連研究

現在の車外サービス連携を実現するために以下の提案が行われている。

4.1. NGTP (Next Generation Telematics Protocol)[4]

NGTPのアーキテクチャは現在のテレマティクスサービスに加え、コンテンツプロバイダやe-Callサービスなどの公共のネットワークへの接続を可能とするように拡張されている。また、ディスプレイと呼ばれるサービス配信機構を置き、サービスの提供と配信を分離する。しかし、サードパーティサービスとの連携についての拡張性は示されていない。

また、NGTPの仕様では、SOAのようなプラットフォーム独立性、疎結合性などは示されていない。また、独自のAPIの標準化を行い、Webへ接続する構造であるため、Webサービスとの連携は十分に考慮されていない。

4.2. SCSN(Smart Car Sensor Network)[6]プラットフォーム

図2で示すSCSNプラットフォームはSCSNドメインとメーカ側のドメイン間のゲートウェイの役割を果たす。SCSNドメインのセンサ情報に加え、IEEE1394やBluetoothを使用する外部システムと車載システムの連携が可能である。

SCSNプラットフォームはOSGiを基盤としたプラットフォームである。このプラットフォームではソフトウェアの部品としてVS(Vehicle Service)がある。このVSのようなソフトウェア部品はbundleと呼ばれOSGiフレームワーク上に実装されることで機能を実現する。また、CANゲートウェイの機能を搭載することでCANプロトコルに柔軟に対応できる。また、VSI(Vehicle Service Interface)ゲートウェイもVS同様にbundleとして実装されている。VSIゲートウェイはVSの情報をCANゲートウェイへ提供する。CANゲートウェイとVSIゲートウェイ間ではTCP/IP stackを用いる。そして、さらに通信処理方法にpublish/subscribeモデルを適用することで車載システムからのデータ受渡しを減少させている。

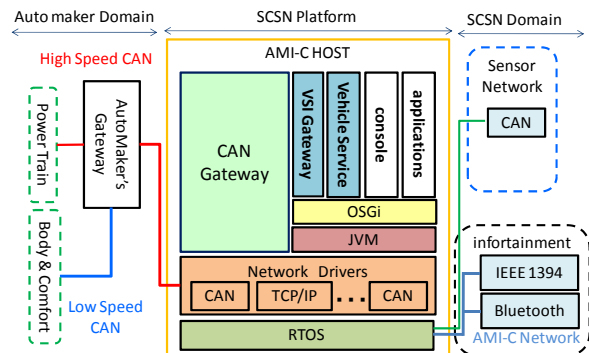


図 2: SCSN アーキテクチャ

4.3. OSGi(Open Service Gateway Initiative)[5]

OSGi[5]は家電製品をインターネットに接続し、パソコンや機器間で相互にサービスを提供するための技術仕様である。OSGi フレームワークは Java 言語を前提とした技術である。図 3 で示すように Java 仮想マシン上で動作する実行環境と Java で記述された bundle と呼ぶ各種アプリケーションソフトウェアやミドルウェアで構成される。そのため、bundle は OSGi フレームワークを実装する異なるプラットフォーム間で再利用を可能である。

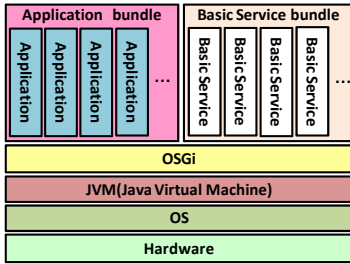


図 3: OSGi の構成

4.4. 車外サービス連携の比較

3 つのテレマティクスサービスの比較を表 1 に示す。既存のテレマティクスサービスではテレマティクスプロバイダ固有のインタフェースやメッセージプロトコルを提供する。本稿では Web サービス標準の技術を用いることで、車載システムとテレマティクスサービスとのサービス連携を容易にする。

表 1: 車外サービス連携の比較

要素技術	現在のテレマティクス	NGTP	SOA
通信プロトコル	SMS, HTTP, SMTP	HTTP	HTTP, SMTP
メッセージプロトコル	固有	固有	SOAP, POX
サービスインタフェース	固有	固有 (API を標準化)	WSDL
サービスの動的発見	無し(固定)	NA	サービスレジストリを利用
サービスの組合せ	無し(固定)	NA	BEPL などを用いて組合せ
Web コンテンツの利用	なし(クローズ)	NA	可能
セキュリティ (認証/暗号化)	固有(認証)	共通(認証/暗号化)	共通(WSSecurity 仕様) [End to End セキュリティ] (認証/暗号化)

5. 車外サービス連携への SOA の適用

5.1. 車載サービスブローカーアーキテクチャ

車載システムへ SOA の技術を導入する形態を示す。車載システムのサービスプロバイダとしての振る舞いと、サービスリクエスタとしての振る舞いについて説明する。

(1) 外部システムのサービスの利用

車載システムが外部システムのサービスを利用する場合、システム内の機器が外部システムのサービスを利用することになる。

(2) 車載システムのサービスの利用

外部システムが車載システムのサービスを利用する場合、外部システムから車載システムの各機器の機能が利用可能となる。例として、NGTP でも利用されている窓の開閉やハザードランプ等の点灯と消灯、ドアロックの on と off 等がある。

6. 車載サービスブローカーの提案

本稿ではメッセージ/プロトコル変換を行う車載ゲートウェイを車載サービスブローカーとして提案する(図 4)。車載サービスブローカーでメッセージ/プロトコル変換を行うことにより、SOAP/POX のメッセージによる車外サービス連携を可能にし、現在のテレマティクスサービスや NGTP よりも柔軟なサービス連携を実現可能にする。

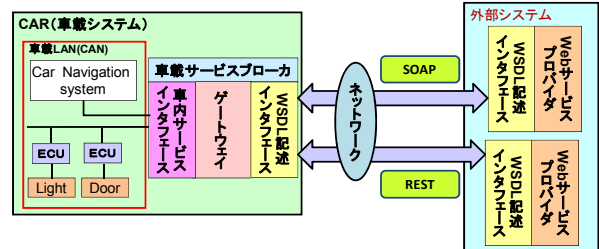


図 4: 車載サービスブローカー

メッセージ/プロトコル変換を行う車載サービスブローカーに 4.3 節で説明した OSGi フレームワークを用いた車載ゲートウェイのアーキテクチャを提案する。OSGi フレームワークはゲートウェイとしてアーキテクチャが提案されている。HTTP の機能を持つ bundle など通信に関連した bundle がデフォルトで提供されている。そのため、Web サービス等の外部システムと通信を行う機能を車載サービスブローカーの実現が容易である。また、OSGi フレームワークでは組込みシステムへの対応も考慮されており、組込みシステムでも実装可能となっている。

6.1. 車載サービスブローカーでのメッセージの送受信

車外サービス連携に SOAP, POX[2] のメッセージを用いるため、車載サービスブローカーでは SOAP, POX の解析を行う必要がある。そのため、OSGi フレームワークには XML パーサや SOAP/POX processor/proxy の機能を持つ bundle を利用する。また、車載サービスブローカーの SOAP/POX processor/proxy では外部システムとの通信プロトコルに HTTP や SMTP を用いるため、HTTP サーバ bundle や SMTP サーバ bundle を実装し外部システムと車外サービス連携を実現する(図 5)。

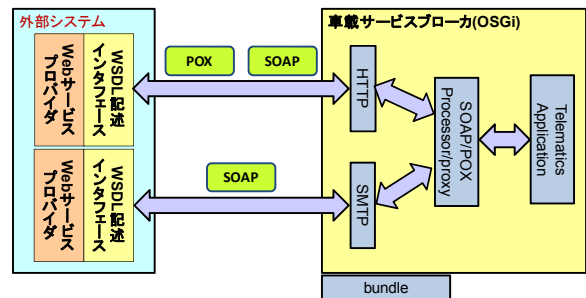


図 5: SOAP/POX の送受信

6.2. テレマティクスサービスの実現

上記で述べたようにテレマティクスサービスの利用形態として車載システムからのサービスの利用する形態が考えられる。この利用形態では車載サービスブローカーへ接続される車載 LAN に MOST, 接続する機器はカーナビゲーションシステムが考えられる(図 6)。

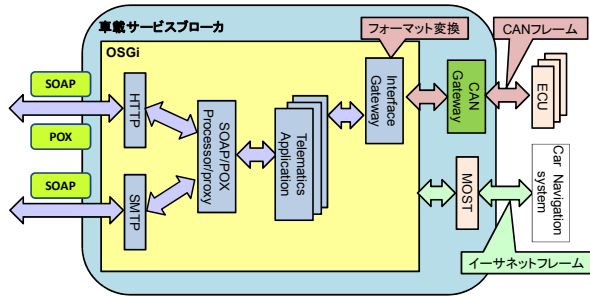


図 6:内部のメッセージ変換処理の流れ

ただし、現在の MOST ではイーサネットフレームをそのまま用いることが可能であるため、車載サービスブローカでのメッセージ変換を行わずにメッセージ送信することが可能である。

6.3. 車載サービスブローカのアーキテクチャ

NGTP のようなテレマティクスサービスを実行するためには CAN 等で接続された ECU にメッセージを送信する必要がある。そのため、車載サービスブローカ内で各車載 ECU に対応したメッセージにメッセージ/プロトコル変換機能が必要である。この機能を実現するため、車載サービスブローカでは OSGi フレームワーク上の bundle とは別に CAN Gateway 等の車載 ECU とリアルタイム性を保って通信可能な機能を実装する。

車載サービスブローカでテレマティクスサービスを実現する場合、OSGi フレームワークにテレマティクスサービスの機能をもつ Telematics Application bundle を実装する。ただし、Telematics Application bundle では 車載サービスブローカの車内 ECU との通信を行う。Telematics Application bundle では Can Gateway へ直接フォーマット変換したデータを渡さず、Interface Gateway でフォーマット変換を行う。

メッセージ/プロトコル変換を行う車載ゲートウェイとして図 7 に示す車載サービスブローカのアーキテクチャを提案する。OSGi フレームワークを用いることで、機能の追加、変更を容易にする。

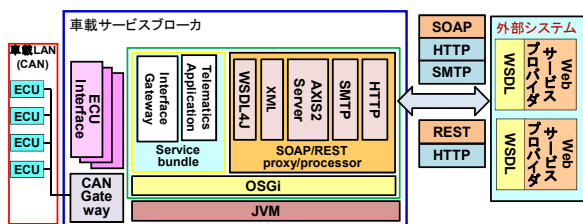


図 7:車載サービスブローカのアーキテクチャ

6.4. メッセージ/プロトコル変換

車載サービスブローカにテレマティクスサービスのリモートセキュリティシステムの機能を適用する。

(1) うっかり通知

うっかり通知ではまず車載 LAN 内の Telematics Application bundle からドアの状を取得する。ECU の状態は車載サービスブローカの CAN ゲートウェイへ通知され、CAN ゲートウェイからフォーマット変換されたデータを Interface Gateway bundle が受け取り、Telematics Application bundle へ車載の情報が渡される。さらに

Telematics Application bundle よりサービスプロバイダへ SOAP/POX メッセージが送信され、車載の情報をサービスプロバイダへ受け渡すことができる。そして、そのメッセージをもとにサービスプロバイダはユーザへうっかり状態を通知する(図 8)。

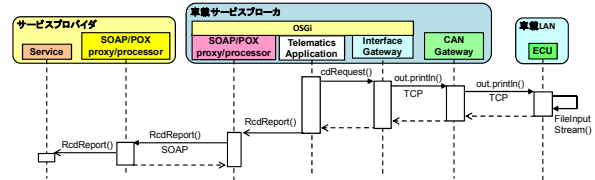


図 8:うっかり通知での処理の流れ

(2) リモートドアロック

ユーザのリモートドアロックサービスを実行した場合、まずユーザはサービスプロバイダのシステムにログインし、リモート操作を送信する。そして、サービスプロバイダは操作受付完了メールを送信し、リモート操作命令を実行する。車載サービスブローカの Telematics Application bundle を実行し Interface Gateway bundle と連携を行い CAN Gateway へデータを受け渡す(図 9)。

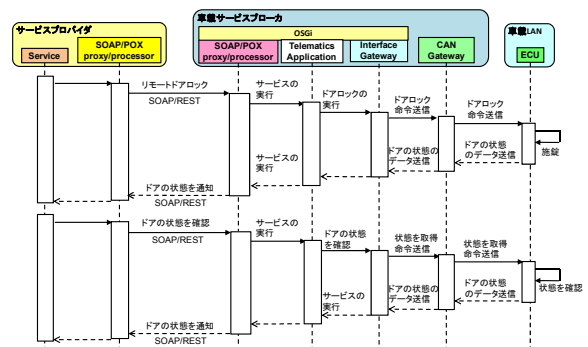


図 9:リモートロックサービスの処理の流れ

7. プロトタイプ

オープンソースの OSGi フレームワークを用いた車載サービスブローカのプロトタイプの実装を以下に示す。

(1) OSGi フレームワーク

本稿では OSGi フレームワークに knopflerfish[2]を用いた。HTTP-Server 等の knopflerfish で提供される Basic Service bundle を利用する。

(2) SOAP/REST Processor/Proxy

SOAP/REST によるサービス実現するために knopflerfish で提供されている Axis2-Server と Web サービスの処理を行う bundle を利用する。

(3) Telematics Application bundle

リモートセキュリティサービスのリモートドアロックサービスを実装する。また、車載システムの情報を外部システムへ返す機能を実装する。

(4) Interface Gateway bundle

Interface Gateway bundle は Telematics Application bundle と連携して車内へ情報を出力する。また、車内の情報を取得し、Telematics Application bundle へ車内の情報を渡す機能を実装する。

8. 車載サービスブローカの評価

8.1. Web サービスとの相互運用性

本稿では外部の Web サービスと相互運用可能な車載サービスブローカの提案を行った。提案した車載サービスブローカによって車載サービスが外部の Web サービスと相互運用可能であるか実装したプロトタイプを用いて検証を行った。

車載サービスブローカでは OSGi フレームワークの Axis2 の機能を実装可能である。そのため、車載サービスブローカ内にサービスを実装し、そのサービスを外部システムから実行することが可能である。

プロトタイプでも OSGi フレームワーク上で Web サービスとして実装されるサービスが実行でき、独立して動作するプログラムとのプロセス間通信による動作も可能である。

そのため、テレマティクスサービスでのリモートセキュリティシステムのリモートドアロックや車載の状態を取得するサービスのように、外部システムからの公開した車載サービスを実行可能である。

OSGi フレームワークからでもうっかり通知のように自動車自身が車載の状態を検出し、車載の状態を外部システムに通知するサービスのように車載側から外部システムのサービスを実行することが可能である。そのため、車載サービスブローカでは外部システムのサービスも利用することができる。

このように、車載サービスブローカでも外部システムと車載サービスブローカ間の双方向の通信が可能である。そのため、テレマティクスサービスの同様のサービスを実現できるといえる。

8.2. メッセージ変換

車載サービスブローカでは SOAP メッセージを受信し、サービスの実行をする。そのため、ドアロックを要求するにはサービスを実行する記述により車載サービスブローカのサービスを実行する。

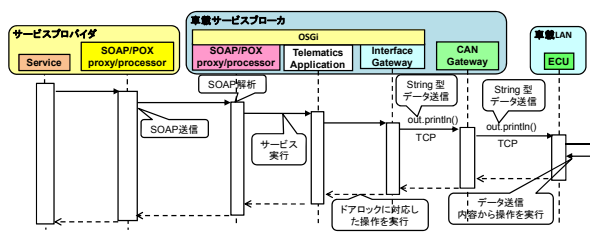


図 10: メッセージフォーマット変換

プロトタイプでは ECU の操作を行うデータとして何を送信するかは Interface Gateway bundle が決定する。そして、CAN Gateway ヘデータを受け渡し、CAN Gateway から String 型のデータの送信を行い ECU の操作を行う。サービスが実行されてからの処理の流れを図 10 に示す。

このように Telematics Application bundle のサービスと Interface Gateway bundle の機能を用いて SOAP メッセージの要求を別のデータ型としてフォーマット変換を行い、テレマティクスサービスを実現可能である。しかし、本稿で提案するメッセージフォーマット変換では SOAP メッセージによるサービスの実行によってメッセージフォーマット変換を行

う、アプリケーションに依存する。そのため、あらたなサービスを実行する場合には bundle のアップデートを行うサービス、機構が必要である。

9. 今後の課題

9.1. 軽量な SOA メッセージの利用

POX での軽量なメッセージはセキュリティの観点からテレマティクスサービスのような車載システムのサービスに用いるには脆弱であるため、セキュリティの強化ができる SOAP の軽量メッセージが必要である。

9.2. 車外サービス連携におけるセキュリティ

車外サービス連携において Web サービスの SOAP メッセージを用いることで WS-Security によるエンド to エンドのセキュリティが可能であることを述べた。しかし、SOAP は SOAP のヘッダなどによる処理時間が長くなる。そのため、車外サービス連携でのセキュリティを強化する場合、組込みシステムのリソース制約を考慮したセキュリティが必要である。

10. 考察

OSGi フレームワーク上では Telematics Application bundle と Interface Gateway bundle の bundle が連携する。そして、Interface Gateway bundle と CAN gateway で連携することでメッセージ/プロトコル変換を実行する。Telematics Application と Interface Gateway bundle を分離することで各 bundle の再利用性は高くなる。

ただし、アプリケーションに依存した、テレマティクスサービスであるため、サービスに対する柔軟性を考慮する場合は bundle をアップデートできるような機構が必要である。

11. まとめ

本稿では自動車ネットワークの車載システムと外部システムとのサービス連携可能なアーキテクチャを SOA を用いて実現する方法を提案した。また、OSGi を基盤とし、メッセージ/プロトコル変換を可能にする車載サービスブローカを提案した。車載サービスブローカの妥当性を確認するためプロトタイプ開発を行っている。

謝辞

本研究のご支援頂いた、(株)デンソーの岩井明史氏と佐藤洋介氏に感謝する。

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] Knopflerfish, <http://www.knopflerfish.org/index.html/>
- [3] L. Richardson, and S. Ruby, RESTful Web Service, O'Reilly, Jul. 2008.
- [4] NGTP (Next Generation Telematics Protocol), <http://www.ngtp.org/>.
- [5] OSGi(Open Service Gateway Initiative) Alliance, <http://www.osgi.org/>.
- [6] P. Park, et. al., An OSGi Based In-vehicle Gateway Platform Architecture for Sensor Extensibility and Interoperability, Proc. IEEE COMPSAC 2009, Jul. 2009.