特集　車載向けサービスプラットフォームの構築と評価*

Design and Evaluation of an Automotive Service Integration Platform

岩 井 明 史
Akihito IWAI

Automotive electronics control systems, which are operated by various types of in-vehicle information/function, continue to become large in scale and complex according to the widespread use of the sophisticated driver-assist-systems, etc. in the market. Moreover, outside the vehicle, various communication services are growing widely, and those services have become to be closely linked to vehicles through mobile information devices. The fusion of in-vehicle information/function and out-of-vehicle information services progresses towards service integration systems. Nowadays, the automotive industry is striving to realize a reliable and safe service integration system. This paper presents a newly developed automobile-oriented service integration platform based on SOA (Service-Oriented Architecture) which is platform architecture to integrate the information service and communication service via the Web. Additionally, we present evaluation results of this functionality and real-time performance from an architectural point of view regarding the orchestration between in-vehicle and out-of-vehicle services.

Key words：Cloud computing, SOA, Service-Oriented Architecture, REST, Automotive software, AUTOSAR, Embedded software, Telematic services

1．はじめに

近年，ACC（Adaptive Cruise Control）やLKS（Lane Keeping System）等の高度な安全運転支援システムの普及に伴い，車載電子制御システムは，益々，大規模化・複雑化の一途を辿っている．更に，最近では，車外の交通インフラやiPhone, Android等のモバイル情報機器とも繋がり，クルマと車外の複数の情報サービスとが密に連携・融合する，所謂，サービス統合システムへと発展している．

また，将来的には，低炭素かつ安全・快適・便利な新モビリティ社会の実現に向け，クルマは，PHV（Plug-in Hybrid Vehicle）/EV（Electric Vehicle）の電動化と合わせ，家・ビル・店・街等あらゆる社会の構成要素と繋がり，サービス連携範囲が，更に拡大していくと予測されている．

本稿では，そうした状況において，今後の広範囲かつ付加価値の高い車内外のサービス連携の仕組み実現に向け，IT業界の標準技術の一つであるSOA（Service-Oriented Architecture）に着目し，サービス統合システムの核となる安全かつ信頼性の高い車載向けのサービスプラットフォームのアーキテクチャを提案し，その妥当性を評価したものである．

2．研究背景

本章では，本研究の背景となる自動車分野でのサービス統合システムの課題について述べる．

2.1　自動車分野でのサービス統合システムの課題

車内と車外の夫々の機能やサービスを連携・統合した自動車分野でのサービス統合システム（**Fig. 1**）を構築する場合，車内の車載電子制御機器（ECU：Electronic Control Unit）と車外のパソコンやモバイル情報端末等の情報機器との特性の違いに着目する必要がある．両者は，夫々の機器に求められる特性がかなり異なる．例えば，一般的に，車載電子制御機器の方が，情報機器に対して，リアルタイム性（時間制約），安全性，信頼性，品質，コスト等の非機能要件が厳しい．また，開発プロセスや製品サイクルも，車載電子制御機器の方が長い．従って，情報機器と同様の常時接続を前提とした，クルマの情報通信ネットワークシステム，即ち，自動車分野でのサービス統合システムを構築しようとした場合，情報機器の場合とは異なる課題が生じる．

＊2013年7月18日　原稿受理

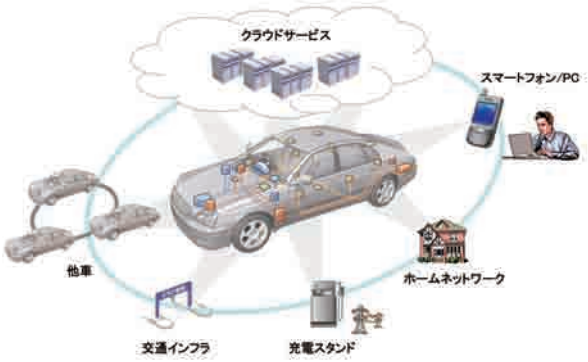

Fig. 1　Service Integrated System in Automotive Domain

(1) リアルタイム性の保証（時間制約の厳守）

車載電子制御システムは，時間制約が厳しいリアルタイム組込みシステムである．センサで取り込んだ信号を，決まった時間内に処理し，アクチュエータを適切なタイミングで制御しなければならない．もし，何らかの影響で決められた時間を過ぎてしまった場合は，制御性が損なわれ，最悪，システムが誤動作する場合がある．つまり，サービス統合システムにおいては，例えば，車外の情報機器からインターネット経由で，センサ情報を取得，或いは，アクチュエータを制御しようとした場合等，リアルタイム性の保証（時間制約の厳守）が課題となる．

(2) 安全性・信頼性の保証

車載電子制御システムは，人命に関わるクリティカルシステムである．そのため，システム構築上，安全性・信頼性が最も重視される．サービス統合システムにおいて，信頼性が不十分な車外の情報機器・情報インフラと，高い安全性と信頼性が要求される車載制御機器をシステム統合する場合，全体のシステムとしての安全性・信頼性の確保が課題となる．つまり，車外の情報機器が故障した場合においても，車載制御機器には，何ら影響を与えない仕組み（Isolation）が必要となる．情報セキュリティ面においても同様である．悪意を持ったクラッカーからの不正アクセスに対して，防御するファイヤウォールなどの情報セキュリティの仕組みも不可欠である．

(3) 車外情報システムの短期製品サイクルへの追従

情報機器は，車載制御機器と違って，製品サイクルが早い．また，ソフトウェアのアップデートが容易であり，製品リリース後も機能アップが可能である．インターネット上のサービスプロバイダから，オンデマンドでサービスも利用できる．これは，上述のリアルタイム性・安全性・信頼性が，車載制御機器に比べ，あまり厳しくないという理由もあるが，情報通信分野での技術が多く標準化されていることが大きい．例えば，インターネット上のサービスの場合は，既に，HTTPやFTP等の通信プロトコルが標準化されているため，誰でも簡単に世界中のサイトにアクセスできる．SOAPやREST等のWebサービスの高位プロトコルも標準化されているため，サービス連携も容易である．しかし，一方，自動車分野においては，この様なサービス系の標準化は，まだ始まったばかりである．Telematics向けのNGTP（New Generation Telematics Protocol）[1] や，Genevi [2] 等があるが，普及段階には至っていない．

3．車載向けサービスプラットフォームによる解決アプローチ

3.1　狙い・目的

本研究では，自動車分野でのサービス統合システムの課題を解決するために，車載電子機器向けの制御系プラットフォームと情報機器向けの情報系プラットフォームに，サービスレイヤを追加した新しい車載向けサービスプラットフォーム（開発コードネーム：ASPF）の構築を考案した．その際，制御系プラットフォームは，現在，自動車業界標準のAUTOSAR [3] を前提とした．また，情報系プラットフォームは，IT業界の標準であるWindowsやLinuxを想定した．この車載向けサービスプラットフォームは，大規模・複雑化するサービス統合システムにおいて，車内外のサービスを安全かつ迅速に連携・統合する事を目的としている．また，車載電子制御システムで，重要なリアルタイム性の保証も狙いとしている．

3.2　SOA（Service-Oriented Architecture）アプローチ

数年前より、通信サービスと情報サービスとをWebを介してシームレスに統合するための基盤アーキテクチャであるSOA（Service-Oriented Architecture）[4] [5] の研究が盛んになっている．SOAでは，全ての機能を”サービス”として扱うとともにサービス間の相互作用を高い抽象度で標準化する事により，物理的なシステム構成や下位の通信プロトコルを隠蔽し，高い相互運用性を実現する基盤となっている．例えばSOAP，WSDL，BPELはそれぞれ，通信，サービスインター

Table 1　Specific Requirements on Service Integration Platform for Automotive (Summary)

分類	要求
サービス連携・統合	– 車内のAUTOSAR SWC同士が双方向連携できる. – 車外の情報サービスとAUTOSAR SWCが，時間制約内にかつ安全に双方向連携できる. – 様々なユーザーニーズや車両状況に応じた最適なサービスプロセスを定義できる. – 通信途絶中や電源遮断時等の不安定なネットワーク状態でも，サービスプロセスが継続可能.
情報共有	– 複数のクルマのユーザー情報・車両情報を，車外より通信回線を経由して安全に呼び出し，一括管理・共有できる. – クルマの販売後にユーザー情報・車両情報が，車外より安全に追加・削除・変更が可能.
その他	– 車載側の端末コストが低い. – 障害に対する耐故障性が高い

フェース，ビジネスプロセスを記述するための標準であり，W3CやOASISなどの標準化団体により管理されており，既存のIT系システムとの親和性が良い．最近では，リアルタイムSOAと称し，時間制約があるシステムへの適応のため，リアルタイム性の拡張に関する研究も進んでいる[6]．本研究では，このSOAをレファレンスとして，車載向けサービスプラットフォームを構築した．

3.3　車載向けサービスプラットフォームに対する要求

今回，複数のサービス統合システムを想定し，車載向けサービスプラットフォームに対する要求を抽出した（**Table 1**）．その際，特に，既存のSOA技術には不足しているリアルタイム性や信頼性の観点を中心に分析した．

4．ASPFプロジェクト

本章では，上述の車載向けサービスプラットフォーム（開発コードネーム：ASPF）についての基本コンセプトとアーキテクチャ，及びコア技術を解説する．

4.1　基本コンセプト

本研究プロジェクト（ASPFプロジェクト）にて構築した車載向けサービスプラットフォームASPFは，車外の情報システムのライフサイクルに合わせて，動的かつ自律的に進化するクルマを基本コンセプトとしている．将来的には，ASPFは，クルマの購入から10年間，ソフトウェア更新のみで車外のライフサイクルに追従でき，また，ユーザーの多様化に対応するため，機能最小セットのクルマを購入し，ニーズに合わせ，必要な機能を拡張できることを目指している．

4.2　ASPFサービス

ASPFで扱うサービス（ASPFサービス）は，通常のWebサービスとは異なる特徴を有している．クルマは移動体であり，移動した場所や時間によって，ドライバもしくは同乗者に必要なサービスは変化する．ASPFは，Webサービスのような，いつでも，どこでも利用できるサービスではなく，状況とニーズのマッチングによって，いまここで最適なサービスを提供可能とする事を特徴とする．今後，自動車ビジネスにおいて，クルマのユーザーに対して，移動による付加価値を提供することが重要と成ってくる．

4.3　ASPFアーキテクチャ

(1) 前提とするシステム構成

実際に，サービス統合システムを実現する場合，クルマとインターネットの接続形態や，車両内のネットワーク構成等，様々なシステム構成が考えられる．今回は，一般的な以下のシステム構成を前提とした．

* クルマは，インターネットには直接接続されない．インターネット上のサービスを利用する場合は，専用の通信回線を経由する．
* クルマは，専用の車載情報端末（本研究では，ASPF Gateway）を介して車外のサービス（インターネット上のサービスと専用の通信回線上のサービス）と連携する．
* 車載情報端末は，車内の基幹バスを経由して各車載電子制御機器（ECU）と接続され，車両内ネットワークを構成している．
* 車両内ネットワークは，CANやLIN，FlexRay等の車載向け通信プロトコルで構成されている．

(2) 機能アーキテクチャ

SOAをレファレンスとした機能アーキテクチャを**Fig. 2**のように定義した．各層は，SOAのリファレンスモデルに対応している．横断的な機能として，各サービスを統合するフレームワークであるASS（Automotive Service Space）とQoSやSecurity管理，システム診断，リカバリー等の非機能要件を含んだ高信頼性OSがある．機能アーキテクチャは，ASPFのソフトウェアアーキテクチャを，ASPFの機能面から抽象化したものであり，実際の各機器に実装されるソフトウェアアーキテクチャとは異なる．

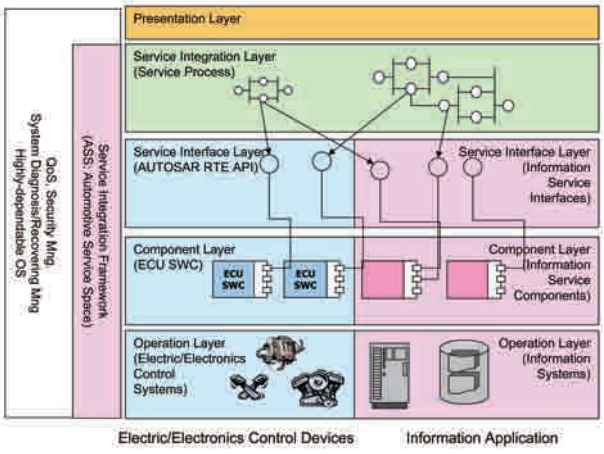

Fig. 2　Function Architecture of ASPF

(3) コンセプトアーキテクチャ

コンセプトアーキテクチャとは，車内外のサービスが連携するASPF Platformの仕組みを概念的に表現した構成図である（**Fig. 3**）．Service Processでサービス統合されたアプリケーションが，SPM（Service Process Manager）によって起動されると，構成要素である車外のサービスを，ASS（Automotive Service Space）から，SOAP，REST又は専用サービス連携プロトコルASP（Automotive Service link Protocol）のインターフェイスを介して呼び出す．また，同時に車内のECUソフトウェアコンポーネント（ECU SWC）を，ASSから，AUTOSAR VFBアダプタを経由してAUTOSAR VFBのインターフェイスを介して呼び出す．本研究において，コアとなる技術が，これらASS, ASPとSPMである．VFBとは，Virtual Functional Busの略であり，AUTOSARで定義されているAUTOSAR SWCを結合する仮想的な機能バスである．

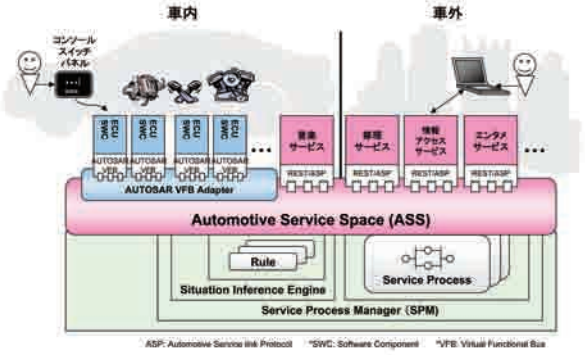

Fig. 3　Concept Architecture of ASPF

4.4　ASPF Platformのコア技術

(1) ASS（Automotive Service Space）

現在，ネットワークシステムにおいて，複数の機器に分散した情報やサービスを，共有化する技術としてJava SpacesやCORBA等がある．しかし，車載向けサービス統合システムの様な、時間制約が厳しく，かつ，高い安全性や信頼性が要求されるシステムにおいては，そのままでは扱えない．従って，本研究では，新たに，これら既存技術をベースに，情報サービス共有空間であるASS（Automotive Service Space）を開発した．ASPFが扱うあらゆるサービス（通常のWebサービスも含めたASPFサービス）は，このASSを介して連携・統合される（**Fig. 4**）．また，QoS確保のため，サービスに優先順位を付与し，重要サービスを優先的に実行させる機構を有している．

サービスへのアクセスは，Public/Subscribeセマンティックス [7] で行われる．実際のASSは，データベースを用いて，複数の機器やサーバへ分散して実装される．また，本研究では，以下の車載向け機能を拡張した．

* AUTOSAR VFB通信機能（AUTOSAR VFBアダプタ）
* サービス優先制御
* 通信途絶対応

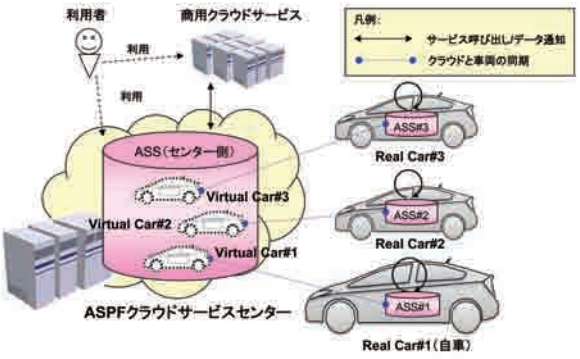

Fig. 4　Role of ASS（Automotive Service Space）

(2) ASP（Automotive Service link Protocol）

ASPは，REST/NGTPのサブセットであり，REST [8] やNGTPと互換性がある．RESTに対して，データ符号化技術を利用したデータ圧縮により軽量化を図っている．REST（Representational State Transfer）とは，HTTPを利用した汎用通信プロトコル仕様であり，サーバが管理するリソース（プロトコル上で扱う論理的な資源）を，URLとして表現する特徴を持つ．また，HTTPプロトコル上の操作（PUT，GET，POST，DELETE）に応じてリソースの操作を行うことになる．通常はリソースの更新（PUT），リソースの削除（DELETE）,リソースの取得（GET）が利用される．

ASSは，ユーザー情報，自動車情報，ECUサービス情報，車外サービス情報を管理するので，これらをそれぞれRESTのリソースとして扱い，URLで表現する．URLには，ユーザーID，自動車ID，サービスID，インターフェイスIDを設定することで，ASSが管理するリソースを一意に特定することができる．例えば，ECUサービス呼出を行うためのURLは，以下のように記述する．

URL記述例：

`http://aspf.jp/api/001/vehicles/{自動車ID}/services/{サービスID}/interfaces/{インターフェイスID}/call`

(3) SPM（Service Process Manager）

SPMは，ASPFサービスのコンセプトを実現するサービスプロセス管理モジュールである．クルマの運転状況に応じたサービスプロセスの起動条件を記述したルールファイル，それを選択し，対応するサービスプロセスを起動するエンジン（SIE：Situation Interface Engine）で構成される．サービス提供者が，予めこのルールファイルを記述しておけば，SPMがクルマの運転状況とルールファイルを照合させ，自動的に必要なサービスを起動する仕組みとなっている．本研究では，ルールファイルを予め全てをサービス提供者が記述する方法を取っているが，推論エンジンや機械学習等の推論機構を応用すれば，もっと自律的にサービスを起動する仕組みができると考えている．

また，本研究でベースとした技術は，Apache ODE, BPEL（Business Process Execution Language）である．BPELは，サービスプロセスの記述に利用している．

5．プロトタイプとケーススタディ

ASPFアーキテクチャのコンセプト検証をするために，実際にプロトタイプを行いケーススタディを実施した．本ケーススタディでは，主に機能性及び原始的な振る舞いに着目し，アーキテクチャ性能を評価した．**Fig. 5**はASPFが実装された乗用車である．本車はデンソー設立60周年を記念し製作された自社製電気自動車の復刻版である．**Fig. 6**は，Androidスマートフォンである．

Fig. 5　Prototype DENSO Electric Vehicle

Fig. 6　Android Smart Phone for Remote Control

5.1. プロトタイプシステム

本プロトタイプシステムの構成を**Fig. 7**に示す．車内と車外の 2 つのサブシステムから構成されている．各構成部品の仕様の抜粋を**Table 2**に示す．

Table 2　Specifications of Prototype System

装置	CPU	メモリ	ソフトウェア
D1: Mobile Terminal	QUAL COMM MSM7201a, 528MHz	Flash 512MB, SRAM 192MB	Android R1.6
D2: ASS Server	Intel Core DUO2 3GHz	2.5G	Debian5.0 Sun Java Ver. 1.6.0_14 Apache Tomcat Ver. 6.0.20, SQLite
D3: ASPF Gateway	VIA C7 1GHz	1GB	Debian4.0 Linter
D4: ECU1	CUSTOM CPU	512KB	AUTOSAR Basic Software Release 3.0
D5: ECU2	CUSTOM CPU	512KB	AUTOSAR Basic Software Release 3.0

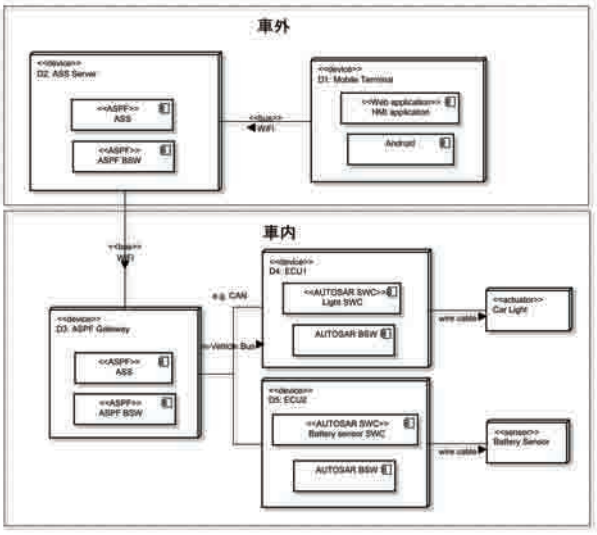

Fig. 7　Configuration of Prototype System

(1) 車内サブシステム：ASPFゲートウェイ，2つのECU，ライト駆動器とバッテリセンサで構成．ASPFゲートウェイは，車内サービスと車外サービス間のメッセージ変換を行う．ASPFゲートウェイには，ASS等，数種類の必要なソフトウェアが搭載されている．また，2つのECUには，AUTOSAR BSW（Basic Software）が実装されており，標準的な車内ネットワーク（例：CAN，FlexRay）に接続されている．ECU1では，ライトSWC（Software Component）が，BSWを仲介して，ケーブル経由でライト駆動器に繋がっている．ECU2では，同様に，バッテリセンサSWCが，BSWを仲介して，ケーブル経由でバッテリセンサに繋がっている．

(2) 車外サブシステム：ASSサーバとモバイル端末（Androidスマートフォン）で構成．ユーザーは，Androidスマートフォンから，リモートで車内サブシステムのASPFゲートウェイを介して各ECUにアクセスできる．ASSには，AUTOSARアダプタが内蔵されている．また，他のASPF BSWが，ASSサーバに搭載されており，車内サービスと車外サービスの統合を行うサービスプロセスを動作させている．
ただし，今回SPM（Service Process Manager）は，スタブとして実装した．また，ルールベースのリポジトリには，商用の組込みデータベースを使用し，通信断絶時の修復及びトランザクション処理を実行させる．

5.2　利用シナリオ

以下二つの利用シナリオで動作させ評価した．

シナリオ1：ヘッドライト遠隔制御

サービスタイプ：車外サービスから車内サービスを呼び出し

サービス記述：ユーザーがリモートでモバイル端末（Androidスマートフォン）から車のヘッドライトを点灯／消灯

シナリオ2：バッテリ充電容量モニタ

サービスタイプ：車内から車両情報を車外に提供

サービス記述：ユーザーがモバイル端末（Androidスマートフォン）からリモートで車内のバッテリセンサ情報をモニタ

(1) シナリオ1の振る舞い

シナリオ1のシーケンス図を**Fig. 8**に示す．一旦，モバイル端末（D1）のHMIアプリ（P1）が，ユーザーの"ライト点灯"ボタンの押下イベント（E1）を検出すると，ASSゲートウェイに対し，"ライト点灯"サービス（M1）を呼び出す．この呼び出しAPIのURLは，次のように記述される．

```
http://aspf.jp/api/001/vehicle/PRI0001/services/SwcLoc/interfaces/MainLight/call
```

上記URLで，PRI0001は，第5章で示したように，“Car ID”，SwcLoc は“Service ID”，MainLightは“Interface ID”である．

ASSサーバ（D2）のASSが，このサービス呼び出しを受信すると，ASSリポジトリの中の対応するライト点灯サービスプロセス（P2）を検索し，見つかればそのサービス（P3）を起動する．その際，"ライト点灯"サービスプロセスは，BPELで記述される．また，ASSはサービスプロセス構成を分析し，ASPFゲートウェイ(D3)から，ASPプロトコルを使い，車内の"ライト点灯"サービス（M2）を呼び出す．ASPFゲートウェイでは，ASSが"ライト点灯"サービスをASSリポジトリに書き込み（P4），ECU1（D4）が認識可能なVFBメッセージに変換する．更に，ASPF BSWは"ライト点灯"メッセージ（M3）をECU1にVFBプロトコルを使い送信する．最後に，ECU1内のライトSWCはヘッドライトを点灯させる．

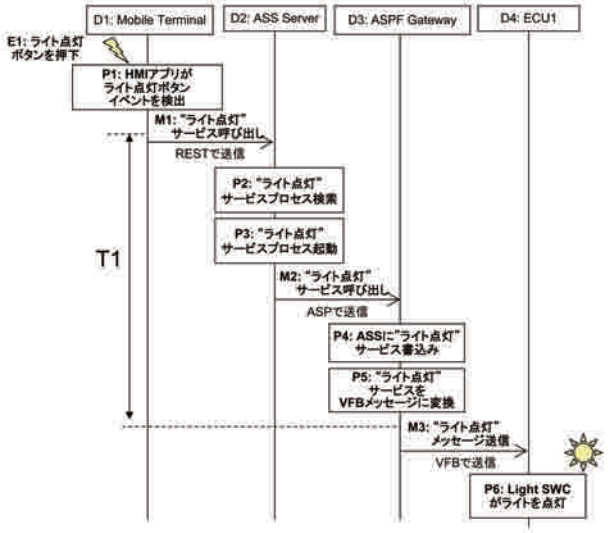

Fig. 8　Remote Control of Vehicle Head-lights

(2) シナリオ2の振る舞い

シナリオ2のシーケンス図を**Fig. 9**に示す．シナリオ1と同様に，"バッテリセンサ情報"サービスプロセス(P5)が起動され，ASSとASPプロトコルより動作する．詳細は，省略する．

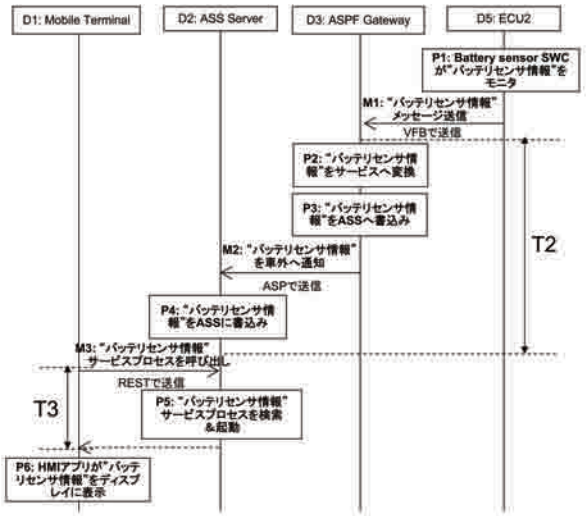

Fig. 9　Battery Charging Monitor

5.3　性能評価

ASPFアーキテクチャの性能を，単体性能，及び，システムレベルの性能の2つの観点から評価した．ただし，本稿においては，紙面の都合上，単体性能の評価結果は割愛し，システムレベルの性能評価結果についてのみ記述する．なお，今回の評価は，主なサービス呼び出しシーケンスのオーバーヘッドをプロトタイプシステムにおいて上述の二つのシナリオを用いて計測した．

Table 3と**Fig. 10**にプロトタイプシステムの性能統計値を示す．その際，T1,T2,T3の計測時間は，**Fig. 8**と**Fig. 9**のシーケンス図の中で定義している．

Table 3　End-to-End Response Time of Service Call

シナリオ	最小	最大	平均	標準偏差
T1	139	544	162.8[205.1]	15.9[120.7]
T2	91	524	192.9[212.4]	131.1[135.4]
T3	105	433	259.3[266.6]	93.2[90.3]

注記：[]は初回の計測を含めた統計量を示す

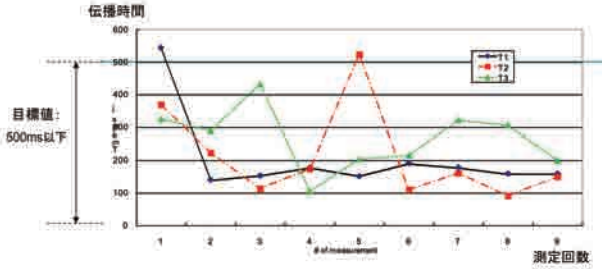

Fig. 10　Propagation Time in Prototype System

(1) T1は，車外のサービス或いはアプリ（今回は，AndroidスマートフォンのHMIアプリ）から車内サービス（今回は，ECU1のライト点灯／消灯サービス）を呼び出す時間である．即ち，オーバーヘッドは，ASSサーバの検索時間，ライト点灯／消灯サービスの呼び出し時間，ASPFゲートウェイのASSに対する要求書き込み時間，ASPFプラットフォームでのVFBメッセージ変換時間の総合計時間となる．

(2) T2は，車内サービス（今回は，ECU2のバッテリセンサモニタ）での車両情報を呼び出す時間である．即ち，オーバーヘッドは，データ変換時間，ASPFゲートウェイからASSに書き込む時間，ASSサーバからASSに書き込む時間の総合計時間となる．

(3) T3は，バッテリ状態を呼び出し，ASSサーバに公開する応答時間である．通常，サービスは応答時間内に周期的に呼ばれる（この評価では600msに一回）．その際，ASSサーバ内のASSのサービス検索時間は含まれる．本評価では，サービス統合のオーバーヘッドは，実利用での妥当性を鑑み，500ms以下を目標とした．然しながら，**Table 4**の分散及び標準偏差を見てみると，初回測定時（a）では，T1,T2,T3全てにおいて大きな遅延が見られる．特に，T2のばらつき（標準偏差）が大きい．詳細データを解析した結果，次の原因が想定される．

(a) 初回測定時の長い応答時間：
ディスクからメモリへのミドルウェアの初回ロードに時間が掛かっている．

(b) T2の大きいばらつき（標準偏差）：
主な原因は，ASSサーバの性能（書き込み時間）のばらつきが大きい事による．本プロトタイプシステムにおいて，ASSは，ASPFゲートウェイとASSサーバの両方にあり，ASPFゲートウェイは，組込みDBのLinterを使い実装されているが，ASSサーバ側は，ビジネス系で良く使用されているフリーのSQLiteを使っている．組込みDBは，書き込み時間が短くかつばらつきが少ないが，SQLiteは，書き込み時間が長くかつばらつきが大きい．T2の場合では，T1と比べた場合，ASSサーバでの書き込みがあり，SQLiteの書き込み時間のばらつきが原因となって，T2の測定時間のばらつきを引き起こしている．

その為，初回のロード時間の改善，及び，高性能の商用DBを使用することでT2の最大値とばらつきを減少させる事が期待できる．また，今回，簡単なシナリオを使い評価を実施したが，今後は，実際のシナリオを用いた評価が必要となる．

6．考察

本ケーススタディにおいて，これらのシナリオが期待通りに動作するか否かをASPFアーキテクチャに基づいたプロトタイプシステムで確認した．

(1) ASPFアーキテクチャの実現性
本研究の最大の成果は，リアルタイム性・信頼性を確保しつつ，車内外のソフトウェアプラットフォーム間をシームレスに接続するSOAベースの仮想サービスプラットフォームのコンセプトを検証した事である．ASPFアーキテクチャのコア技術は，ASP（ASPF Service link Protocol）とSOAP/REST対応と共に，ASS（Automotive Service Space），SPM（Service Process Manager）及びSIE（Situation Inference Engine）となるが，これら個々の技術は，既存分野では実装されて来た．然しながら，自動車ソフトウェアシステムへ適合する際には，これら技術の融合が課題となる．本研究では，複数の組込みシステムに分散された複雑かつセーフティクリティカルな自動車分野において，これら技術の融合の実現性を実証した．今日，車内の複数のECUは，異なるベンダーによって個々に開発されているが，より多くの機能が，複数ECUに跨る協調動作が必要となっている．今後，自動車ソフトウェアは，情報社会においての役割の拡大と同様，更なる燃料消費や排気ガスの低減に向け，車外のサービスや社会インフラとの協調が期待されている．

(2) プロトタイプ性能
今回の二つの利用シナリオを用いた実車でのプロトタイプにおける一連の性能評価で，ASPFアーキテクチャの振る舞い特性を実証した．今後更なる検討は必要だが，プロトタイプ性能は，アーキテクチャの実現性を示すのに十分良好な結果が得られた．

(3) 従来研究との比較
従来研究[4) 9) 10)]や関連研究[5) 11) 12)]と比較し，ASPFアーキテクチャの以下二つの利点を主張できる．

* 車内サービスと車外サービスとのシームレスな統合
* AUTOSARアーキテクチャの様な従来アーキテクチャと進化的な協調

(4) 実験から，ASPFコンセプトが実現可能かつ効率的である事を確認した．更なる評価は実施中である．

7．今後の課題

今回，ASPFアーキテクチャの実現性の確認，及び，シンプルなシナリオを用いて性能を評価した．今後，非機能面に着目し，リアルタイム性能，安全性，及び，信頼性を検討していく．更には，SPMの為のコンテキスト／シチュエーションアウェアな検索及び起動の詳細メカニズムの検討も行う．

8．おわりに

本稿において，自動車分野でのサービス統合システムの課題分析を行い，この課題を解決するSOAをベースとした車載向けサービスプラットフォーム（ASPF）を提案した．また，自動車分野での実ユースケースを想定した複数のシナリオを用いたプロトタイプ環境を構築し，本アーキテクチャの実現性を確認した．今後，更に安全性・信頼性・セキュリティ等の非機能要件を検討していく．

＜参考文献＞

1）NGTP, Application Services Layer, Version 1.0, Jan. 2008, http://www.ngtp.org/.
2）GENIVI, http://genivi.org/.
3）AUTOSAR, http://www.autosar.org/.
4）青山 幹雄，サービス指向アーキテクチャの誕生と進化，ソフトウェアエンジニアリング最前線2008 （情報処理学会ソフトウェアエンジニアリングシンポジウム2008論文集），近代科学社, Sep. 2008, pp. 9-16.
5）T. Erl, Service-Oriented Architecture, Prentice Hall, 2005.
6）M. Panahi, et al., A Framework for Real-Time Service-Oriented Architecture, Proc. of IEEE CEC 2009, Jul. 2009, pp.460-467.
7）P. Th. Eugster, et al., The Many Faces of Publish / Subscribe, ACM Computing Survey, Vol. 35, No. 2, Jun. 2003, pp. 114-131.
8）L. Richardson, et al., RESTful Web Services, O'Reilly, 2007.
9）A. A. Ahson and M. Ilyas (eds.), Cloud Computing and Software Services, CRC Press, 2010.
10）L. Baresi, et al., Hybrid Service-Oriented Architectures: A Case-Study in the Automotive Domain, Proc. of ACM SEM 2005, Sep. 2005, pp. 62-68.
11）L. Bocchi, et al., Service-Oriented Modelling of Automotive Systems; Proc. of IEEE COMPSAC '08, IEEE, Jul./Aug. 2008, pp. 1059-1064.
12）M. Broy, et al. (eds.), Automotive Software-Connected Services in Mobile Networks, LNCS Vol. 4147, Springer, 2006.

＜著　者＞

岩井　明史
（いわい　あきひと）
電子基盤システム開発部
先行技術開発室
基盤ソフトウェア技術の
研究開発に従事