

インターネットITSアーキテクチャを用いた車載システムの構築

遠山 祥広[†]

佐藤 雅明[†]

植原 啓介[†]

村井 純[‡]

Implementation of an In-Vehicle System for the InternetITS Architecture

Yoshihiro Toyama[†]

Masaaki Sato[†]

Keisuke Uehara[†]

Jun Murai[‡]

概要

自動車に関するシステムの高度化に関する研究が進む中、インターネットを用いて車外との通信を行なうことで、自動車の情報化をより一層押し進め、あらたなITSビジネスの創出を目指す機関としてインターネットITS協議会が設立された。本論文では、協議会が提唱するインターネットITSアーキテクチャについて述べ、その現状における問題点を挙げた。その問題点を解決するため、インターネットITSアーキテクチャに基づいたシステムを実際に構築した。その上で、各企業が作成したアプリケーションを動作させ、アーキテクチャの妥当性を示した。また、今回の実験から得られた問題点を整理した。さらに、インターネットITSアーキテクチャにおける情報提供形態や通信状態通知機能などの必要性を示し、インターネットITS基盤の実用化と普及に際する課題を明らかにした。

1. はじめに

1.1 背景

ITSと呼ばれる、交通システムの高度化に関する研究が広く行なわれている。自動車に関しても、車載センサを利用することで周辺の環境をセンシングしたり、過去の情報を利用して、渋滞を回避するシステム等の事例がある。

そのような中で、既に情報通信インフラとして広く普及しているインターネットを自動車で利用可能にする研究がある。インターネット自動車と呼ばれるこのシステムは、これまで固定端末中心に普及してきたインターネット環境を、移動体である自動車に持ち込むことによって、自動車と外部、または車車間での情報流通を可能にするものである。

このシステムを利用する例をあげると、自動車から速度情報と位置情報を集めることで、渋滞地図を生成することができる。また、最新のグルメマップや地図情報を自動車にダウンロードすることも可能となる。

インターネットを利用した自動車の情報化は、既に研究段階から実用に向けた開発段階に入ってきており [1]、各企業が開発を進めている分野である。

1.2 本研究の目的

本研究では、車両、車載機、サービスプロバイダ等の複数プレイヤーが活用できるオープンな自動車情報基盤の構築に向け、現状検討されているインターネットITSサービスが基盤に求める要件を整理し、その要件を満たす車載システムを構築することを目的とする。

また、この車載システム上で実際に複数のアプリケーションを動作させ、実フィールドで評価実験を行うことで、インターネットITS基盤の実用化と普及に際する課題を明らかにする。

2. インターネットITSアーキテクチャ

本章ではインターネットITS協議会について概説し、インターネットITS協議会が現在進めている共通基盤(インターネットITSアーキテクチャ)について述べる。

2.1 インターネットITS協議会

インターネットを使った自動車のITS(インターネットITS)とそのアーキテクチャ標準化を進める団体として、インターネットITS協議会 [2] が2002年に発足した。インターネットITS協議会は、ITSサービスを実現するための共通基盤の研究開発と、新たなITSビジネスの創出・活性化を目的としており、現在約100社が参加している産官学共同の団体である。

インターネットITS協議会においては、現状のITS・テレマティクスに見られるような個別シス

[†]慶應義塾大学院 政策・メディア研究科, Keio University Graduate School of Media and Governance

[‡]慶應義塾大学 環境情報学部, Keio University Faculty of Environmental Information

テムによるクローズな接続では、機器・サービスの選択肢が限られ、新規参入コストが高いという課題認識をしている。そこで、様々な機器・サービスの選択を可能にし、新規参入コストを削減するための共通基盤（インターネットITSアーキテクチャ）を構築することを目指している。

2.2 インターネットITSアーキテクチャの位置づけと意義

既存のITSは、個々のサービスが独立してシステムを構築しているため、構築やシステムの協調動作のためのコストが高い。また、交通という枠に特化した仕組みのため、他分野での発展性の無さから、多くの価値ある情報が破棄されているのが現状である。

インターネットを基礎としたインターネットITSでは、自動車の車両・車種や各種システムに依存する事の無い情報の収集・蓄積・利用が可能な基盤の構築を目指す。そのためには、自動車の持つ情報を様々な分野へ利用することを前提にして、自動車の持つ情報を定義していく必要がある。

インターネットITSでは、これまでWIDEプロジェクト[3]のiCAR WG[4][5][6]が提唱してきた、自動車の持つ情報を整理し、各種の情報を「名前」と「形式」からなる「エントリ」として一意に定義するデータ辞書モデル[7][8]に基づく形で、自動車の持つ情報の正規化を行う。このモデルに基づいた自動車情報の保持・取得が統一した方式で行える統合環境を構築することが出来れば、システム開発のコストを低減させることが可能となる。

2.3 インターネットITSアーキテクチャの構成

インターネットITSアーキテクチャ[9]はアプリケーションデータアーキテクチャ、車載アーキテクチャ、ネットワークアーキテクチャの3つから構成されており、それぞれが「車両情報とその取扱いに関する定義」「車内の機器構成の定義」「通信の手法の定義」となっている。

図1に示す通り、各アーキテクチャ仕様は、お互いに情報をやりとりするためのインタフェースを備えている。この3つのアーキテクチャを総合してインターネットITSアーキテクチャと呼ぶ。

データアーキテクチャは、自動車の持つ情報の定義を行なっている。センサの種類やベンダの違いを吸収し、どんな車種であっても一台の車両として外部から認識することを可能とする。また、単位・精度を定義することにより情報を収集した際の加工コストを低減することが可能となる。

ネットワークアーキテクチャでは、既に情報通信インフラとして広く普及したインターネットを

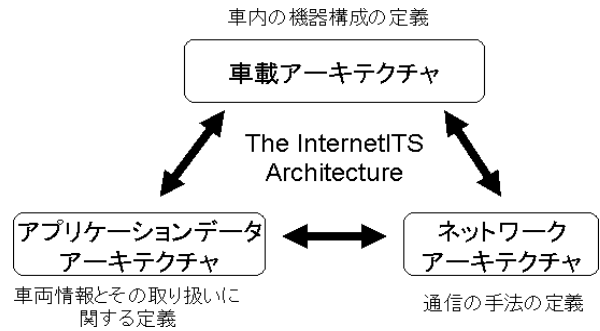


図1: インターネットITSアーキテクチャの3本柱

用い、車車間もしくは車外との通信を行なうことを定義している。これにより、インターネット上に既に存在している各種サービスを自動車内でも利用することが可能となる。

車載アーキテクチャを定義することにより、機種やベンダに依存しない機器構成を実現することが可能となる。これにより、従来とは異なるマルチベンダ環境を実現する。

2.4 現状の課題

現在、前述したように独立してそれぞれの機能の定義がなされ、それぞれに仕様が存在している。しかし、その仕様はこれまで机上でのみ検討され、実用性や実現性は検証されてこなかった。さらに、全ての仕様を満たした実装がなく、実働試験を行なったこともなかった。

今回の実験では、実際に動くシステムを構築することで、以下の3点を確認する。

1. 仕様の妥当性・矛盾点の確認

アーキテクチャを実際にシステムとして作り上げることで、仕様上と実装上の矛盾点の洗い出しを行なう。それにより、仕様の妥当性を検証する。

2. 各機能のインタフェースの過不足の確認

仕様に盛り込まれているインタフェースを利用して、実際のシステムを構築可能か検証する。また、アプリケーションに関しても現在の仕様で構築可能かを検証する。

3. 想定アプリケーションの実現性の確認

現在、インターネットITS協議会が想定しているアプリケーションがいくつかあるが、それらを実際に仕様書に基づいて作成する。そのアプリケーションを実際に実行可能か、実車環境での使用に耐えうるかを検証する。

3. アプローチ

本章では、協議会の仕様に基づいたシステムを実装する手法について述べる。

3.1 機器構成

協議会におけるインターネットを用いた通信の基盤は、IPv6 をベースとしており、車内のインターネット接続は全て IPv6 を用いて行なうこととしている。図 2 のように、インターネットへの接続環境は、モバイルルータと呼ばれる機器により提供される。このモバイルルータには携帯電話網や無線 LAN(802.11b)、PHS など複数の通信メディアが搭載され、それらを切り替えて使用し外部へと接続されることが想定されている。

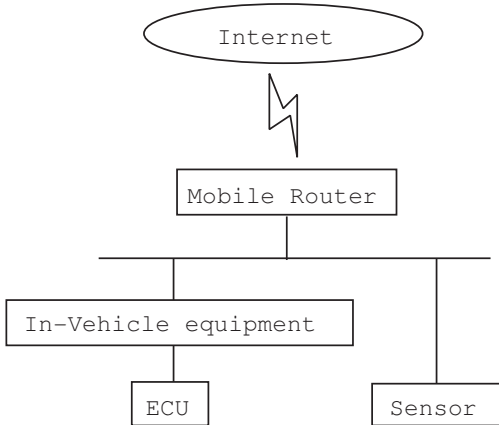


図 2: インターネット ITS アーキテクチャのハードウェア構成

また、カーナビゲーションを主とする車載機は、モバイルルータを利用したインターネット接続性を利用して、車外と通信を行なう。また、車両の ECU(自動車制御用コンピュータ)と通信することで自動車の実情報を利用することが想定されている。その他に車内に設置された各種センサを利用することも想定されている。

本研究では、車両 ECU と通信する機材を別に用意し、それを介して車両の情報を利用する手法を採用した。これは、カーナビゲーションに ECU との通信機能を付加するよりもコストがかからず、実験という意味では問題がないと判断されたためである。

3.2 プログラム実行環境

協議会の仕様においては、全ての車載アプリケーションは Java のプログラムであることが求められている。これは、さまざまなベンダが作る車載機の全てで動作することが求められているからであり、協議会の示すアプリケーションプラッ

トフォームとして必要な事柄である。全てのアプリケーションは Java VM 上で動作することが想定されている。アプリケーションの他に車両情報等のさまざまな情報を蓄積するデータベースが車載機には搭載される。

4. 実験システム的设计

今回の実験システムは、全て協議会の仕様に基づいて作成された。車載機に搭載される機能を PC とカーナビゲーションに分割して実装することで、カーナビゲーションを仮想的にインターネットに接続する構成とした。図 3 にその詳細を示す。

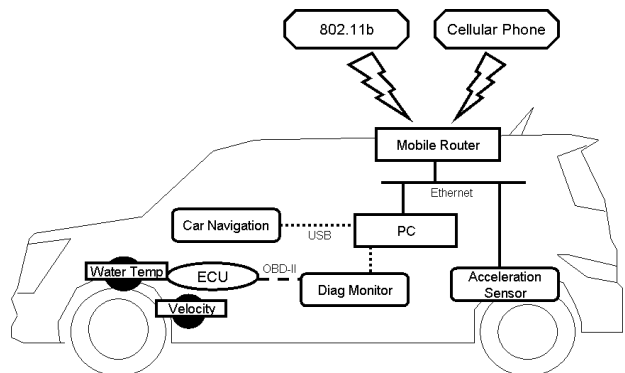


図 3: 実験システムのハードウェア構成

図 2 と図 3 の対応を述べる。モバイルルータは仕様どおり設置し、通信メディアは 802.11b の WiFi と携帯電話網 (EVDO) を利用した。車内は、モバイルルータによって Network Mobility[10][11](NEMO) 環境が提供されており、車内の機器類と車外との通信は、NEMO を利用した IPv6 による通信である

車載機は車載機用 PC、カーナビゲーション、ダイアグモニターの 3 つに機能分割し、それぞれを接続することで実現した。ダイアグモニターについては、後述する。

車載機用 PC とカーナビゲーションの接続は、USB によって行なっている。PC にはカーナビゲーションを外部操作するプログラムを組み込み、カーナビゲーションの操作を PC から行なうことができる環境を構築した。これにより、PC が外部と通信を行ない、その結果に基づいて目的地等のカーナビゲーションの設定を行なうことが可能となっている。

ダイアグモニターと呼ばれる機器と車載機用 PC は、USB で接続されている。ダイアグモニターは自動車と CAN[12] 通信を行ない、自動車の ECU の情報を閲覧することができる機器である(図 4)。CAN は車両制御のための車内 LAN 用プ

ロトコルとして国際標準となっているものである。自動車との接続には、OBD-II[13][14]と呼ばれるインターフェースを利用する。OBD-IIは2000年以降全ての車両に導入が義務づけられており、それ以降販売された全ての新車に装備されている。本実験においては、ダイアグテスターのプログラムと通信することで、自動車の情報を正確に利用することを可能とした。



図 4: ダイアグモニター

追加のセンサとして、今回はIPセンサ(図5)を利用して加速度の計測を行なった。これはSNMPを利用して情報を取得できるセンサである。



図 5: 加速度センサ (IP センサ)

この構成により、自動車 ECU からの車両の実情報と外部センサを同時に利用した「インターネット ITS アーキテクチャ」を構築することができた。次章では詳しい実装を述べる。

5. 実験システムの実装

5.1 実装環境

本実験においては、モバイルルータを日本電気株式会社、株式会社 KDDI、株式会社 IAL の 3 社

から提供を受けた。各ベンダが作成したモバイルルータの実装は異なるが、全てインターネット ITS アーキテクチャのネットワークアーキテクチャに基づいて実装されているため、どのモバイルルータを使用しても問題がないことが実証された。

ダイアグテスターとカーナビゲーションは株式会社デンソーの物を使用した。車載機用 PC は表 1 に示すものを使用した。

表 1: 実装環境

OS	Windows XP SP2
Java	JDK 1.5
Database	Oracle lite 10g
アプリケーション環境	Knopflerfish OSGi

また、以下の図 6 に実装のソフトウェア構成を示す。

Applications		Vehicle Data Manager	
Oracle API	Navi API		
Oracle lite 10g	knopflerfish OSGi	SNMP	DiagMonitor API
	Java VM		DiagMonitor
Windows XP			

図 6: 実験システムのソフトウェア構成

本実験システムは、OSGi[15] のひとつの実装である knopflerfish[16] を利用し、アプリケーションの動的な変更・改修を可能にしている。

車両情報の利用順序は図 7 のとおりである。

まず自動車の情報を取得するためのダイアグモニターとの通信はデンソー製のソフトウェアに改良を加えることで実現した。これはダイアグモニターと通信を行なうためのソフトウェアである。同一ホスト内において TCP を利用し、統合環境のプログラムとダイアグモニターが通信を行っている。ダイアグモニター API は、10 秒毎にデータの要求をダイアグモニターに送信する。データ要求に対して、車両情報を API に返す。その際の通信プロトコルは独自プロトコル (csv による順序決め打ち) である。情報取得間隔については、今回利用したダイアグモニターのハードウェアの制限により、10 秒が限度であった。

カーナビゲーションから車両の位置情報を取得するために、OSGi 上にカーナビゲーションとの通信を仲介する Navi API を実装し、それを介して位置情報の取得を行なっている。カーナビゲーションへの目的地設定等も、この Navi API を利用して行なわれている。Navi API は 1 秒毎に情報

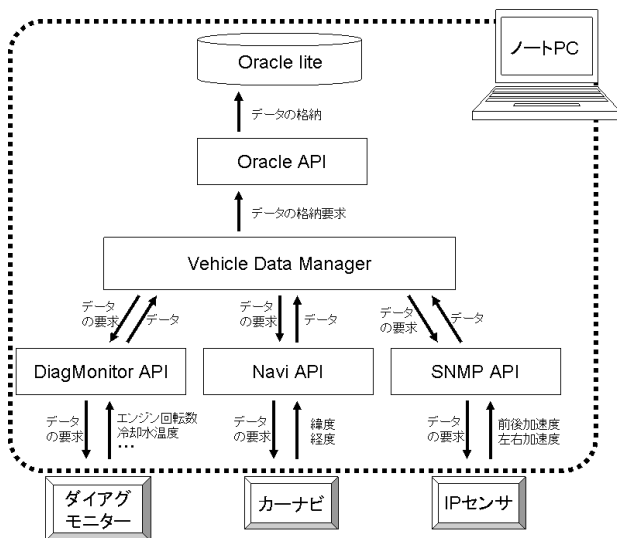


図 7: 実装システム図

の要求をカーナビゲーションに送信し、情報を取得する。その際のプロトコルはカーナビゲーションに依存するため、独自プロトコルとなっている。

また、加速度センサとの通信は SNMP を介して行なわれる。これは、Windows 用の実行ファイルである SNMP を用意し、これを外部プロセスとして発行している。これにより、加速度情報の取得を行なっている。

車両情報統合プログラム (Vehicle Data Manager) は、それぞれの API を介して情報の取得を行ない、アプリケーションデータアーキテクチャに基づいた単位・精度へと変換し、データベースへと情報を格納する。

データベースとの接続は、Oracle DB(Oracle lite) の利用を一括して担当する Oracle API を実装し、それを利用して取得した情報をデータベースに保存している。アプリケーションは保存された情報を、同じく Oracle API を利用して取得している。その情報を元にさまざまなサービスを実現する仕組みとなっている。

5.2 取得情報

本実験において、アプリケーションが利用する情報はダイアグテスター・IP センサ (加速度センサ)・カーナビゲーションの 3 つから取得されたものである。取得情報は以下の表 2 の通りである。

6. 実験概要

本実験では、さまざまな会社の作成したアプリケーションをひとつの環境で実現した。図 8 にトップメニュー画面を示す。

表 2: 取得情報一覧

名称	情報源	単位・精度	説明
前後方向 加速度	IP センサ	0.001G	車内に設置された IP センサから SNMP を利用して取得する。車両の前後方向の加速度を示す。
左右方向 加速度	IP センサ	0.001G	車内に設置された IP センサから SNMP を利用して取得する。車両の左右方向の加速度を示す。
緯度・経度	カーナビゲーション	0.0001 度	接続されたカーナビゲーションから取得する現在の車両位置情報。測位系は WGS84。
速度	ダイアグテスター	1km/h	現在の車両のメータ表示の速度を示す。
エンジン回転数	ダイアグテスター	1rpm	現在の車両のメータ表示のエンジン回転数を示す。
冷却水温度	ダイアグテスター	1	現在の冷却水の温度を示す。
ABS 制御フラグ	ダイアグテスター	on/off	急ブレーキを踏んだ際の Anti-lock Braking System (ABS) の作動状態を示す。
パーキングブレーキ	ダイアグテスター	on/off	パーキングブレーキの作動状態を示す。



図 8: アプリケーション選択画面

6.1 実装アプリケーションの詳細

車両デジタル診断

車両デジタル診断は、ダイアグテスターから取得された故障情報(ダイアグコード)を元に、自動的に車両の故障を判定する。故障が発見された際には、周辺の整備工場を検索し、ユーザに整備工場での整備を勧めるというアプリケーションである。今回は実装を行っていないが、車両に保存された過去の整備データを参照し、部品の経年劣化を自動で検知したり、また、整備工場においても他の工場で行なわれた過去の整備データを参照することも想定されている。

ナビポケット

ナビポケットは、カーナビゲーションの行き先設定を簡易にするアプリケーションである。ユーザは事前に経由地や行き先などをインターネット上のサーバに登録しておく。乗車した際に、登録しておいた経路をダウンロードし、自動的にカーナビゲーションの設定を行なうアプリケーションである。

天気情報

このアプリケーションは、現在地及び目的地の天気予報を表示するものである。天気の情報、インターネットを利用し(財)日本気象協会から提供された最新情報である。

駐車場検索

駐車場検索は、目的地に近づくと自動的に起動し、目的地付近のパーキングを自動検索・表示するアプリケーションである。駐車場情報には値段も含まれており、ユーザの要望にあわせた駐車場検索が可能である。また、最新の満空情報を取得することも可能で、駐車場についてから満車だったというようなこと

を防ぐことができる。今回は実装を行っていないが、一部のパーキングにおいて駐車場の予約を行なうシステムの開発が進められており、予約システムとの連動も視野に入れている。

プローブ

プローブには大別して3つの機能がある。1つ目は、危険道路情報の送信である。走行中に急ブレーキ情報(ABS情報)が検知された場合、その場所と時間をサーバへ送信する。サーバでは、この情報を集約することにより、急ブレーキが多く踏まれた地点を危険地点として認識することが可能となる。2つ目は、危険道路情報のダウンロードである。1つ目の機能で生成された情報をダウンロードし、経路上に危険地点があった場合、ドライバーに事前に警告する。これにより、ドライバーは危険地点において、より安全な運転を心がけることが可能となる。3つ目はエコドライブ診断である。駐車時に、運転の仕方から急発進など環境負荷が高い運転がなかったかを診断し、その運転の仕方についてのアドバイスを行うものである。これによりドライバーは、より環境を考慮した運転を心がけるようになることが期待されている。3つ目のエコドライブ診断は、今回は実装していない。

グルメ検索

このアプリケーションは、従来のカーナビゲーションについているグルメ検索と基本的には同一である。ただし、利用する情報はインターネットを介して取得された最新情報であり、駐車場検索と連動することで、より快適に利用することができる。また、駐車場とお店の2点が含まれた地図を事前に登録しておいたメールアドレスに送信することも可能である。これにより、駐車場から迷うことなく目的地へ到着することができ、これまで駐車場までであったナビゲーションを目的地まで延長することが可能である。

SS 会員アプリ

このアプリケーションは、サービスステーション(ガソリンスタンド)の会員向けアプリケーションである。具体的には、ガソリンスタンドを検索する場合、会員であるサービスステーションを自動検索し、優先的に表示を行なう。さらに、駐車場に設置された無線LANアクセスポイントを利用し、車両の情報をサービスステーションのサーバへと送信する。それを

利用し、給油口の位置や油種にあわせたレーンへと自動で誘導する。また、送信された車両情報を元に、適切な広告を車両に対して配信する。今回の実験では行っていないが、料金の決算などを自動化する研究も行なわれている。

6.2 実験手法

本実験では、上記のアプリケーションを実装した機器を実際の車両に搭載し、リアルタイムの車両情報を利用した実験を行なった。また、SS 会員アプリの実験のために、サービスステーションと仮定した地域に無線 LAN アクセスポイントを設置した。

走行したコースは、虎ノ門から浜松町周辺を周回するルートを選択した。図 9 は、実際に実験車両の位置情報を元にプロットしたデータを地図上に重ね合わせた物である。テストは 2006 年 1 月上旬から行ない、全てのアプリケーションを同時に実行した最終試験は 2006 年 3 月 27 日から同 29 日まで行なった。



図 9: 走行コース

コース上における通信品質は、約 2 時間でコースを 2 周して計測した。その間の利用通信デバイスは表 3 の通り、約 92% が携帯電話網を利用し、7% が無線 LAN、通信不能状態は約 0.2% となった。無線 LAN のアクセスポイントは、コースの

表 3: 通信デバイス利用率

利用デバイス名	利用率
携帯電話網	92.776%
無線 LAN	6.997%
通信不能	0.227%

南端に一ヶ所設置し、半径約 50m の範囲で電波が受けられる電波強度に設定した。

以下の表 4 に取得したデータの一部を示す。このように今回の実装により、車両の位置情報や車両の状態を示す値を取得することができた。このデータを利用して、前項のアプリケーションを動作させた。

表 4: 取得データ (一部)

時刻	東経	北緯	速度	エンジン回転数	冷却水温度
15:14:52	139.7555	35.6435	6	751	86
15:15:02	139.7556	35.6437	18	1664	86

7. 評価

2.4 章で挙げた 3 点についての評価を行なう。

7.1 仕様の妥当性・矛盾点の確認

今回の実験を行ない、インターネット ITS 協議会の提唱する仕様に矛盾点は確認されなかった。だが、全て Java による実装を行なうべき、という点に関しては問題なかったものの、OSGi を利用することで OSGi にまつわる問題が多発した。ほとんどは高負荷によるフリーズ現象であり、OSGi を利用する点は若干の問題があることが確認された。

しかし、最終的に全てのアプリケーションの動作が確認されたため、一定の妥当性が認められる結果となった。ネットワークに関しては、複数のベンダが作成したモバイルルータを利用したが相互に交換しても問題なく動作し、ネットワークアーキテクチャと車載アーキテクチャの仕様の妥当性が確認された。

データアーキテクチャの部分に関しても、アプリケーションのアルゴリズムに依存せず、かつ、アプリケーションの実行に必要な要求を満たすデータを提供することができた。だが、情報の取得間隔については、ハードウェアの制限から最短で 10 秒かかる情報が存在した。情報取得間隔について、

仕様上の記述はないが10秒ではアプリケーションの応答が遅すぎることが判明した。実際に利用する際には、3秒程度まで短縮する必要があると感じた。

7.2 各機能のインターフェースの過不足の確認

機能の面において以下の3点の不足点・問題点が発見された。

- 情報提供形態

現在の実装では、車両情報を取得する際に、細かな設定ができない。例えば、10分前から5分前までのエンジン回転数の推移、という情報が欲しい場合は、10分間の情報を取得し、必要な部分だけアプリケーションが抜き出す、という作業が必要である。情報をデータベースに蓄積するAPIは十分であったが、あらゆる利用形態に応じた情報提供APIが必要である。また、そのことにより車載機の負荷低減を図ることができる。

- 通信状態通知機能

現在の仕様においては、モバイルルータが現在利用している通信手段(通信メディア)の情報が隠蔽されている。SS会員アプリを実装するために今回は、特殊なコードをモバイルルータに搭載して対応したが、今後似たような機能を必要とするアプリケーションがあると予想される。利用している通信メディアや通信状況をアプリケーションに通知する機能が仕様に盛り込まれる必要があると考える。

- 突発的な通信の切断への対応

インターネットITS協議会の提唱する環境において、現在の技術では通信が突然切断されたり、メディアの切り替え時に通信が不安定になる可能性がある。今回実装を行なったいくつかのアプリケーションにおいて、そういった場合にアプリケーションの動作が不安定になることがあった。自動車に搭載するアプリケーションを実装する際は、そういった可能性を考慮し、通信が不安定であっても動作するような処理が必須であることがわかった。

- アプリケーションの操作インターフェース

また、今回はPCを利用することで仮想的に標準化アーキテクチャに乗っ取ったプラットフォームを実装したが、操作は基本的にキーボードとマウスによるものであった。運転者が操作する可能性がもっとも高いことから、音声認識等の操作インターフェースを標準化プラットフォームとして定義する必要がある。

- 個人情報保護

いくつかのアプリケーションには、車両を特定した上で行なわれるサービスが存在している。自動車の情報には個人情報が多く含まれているため、これらの情報をインターネットを介してやりとりする際には、十分な注意が必要である。また、どの情報が個人情報として保護すべきか、一概に判断することが難しい。例えば、単一では個人特定できないものの、ある特定の情報と合わせることで、個人の特定につながる情報なども存在する。そのため、標準化プラットフォームにて個人情報の定義を行ない、それらを交換する手法も定義する必要がある。もしくは交換の際のポリシー等をあらかじめ決定しておくことが望まれる。

7.3 想定アプリケーションの実現性の確認

このアーキテクチャを利用した実際の車両の情報を利用したアプリケーションを作成し、実行した。仕様に基づいた設計を行なうことで、想定されていたアプリケーションが正しく動作することが確認された。これにより、現在想定されているアプリケーションが仕様上、問題なく実現できることが確認された。

8. 結論

本実験では、インターネットITS協議会の提唱するインターネットITSアーキテクチャを実際のシステムまで落とし込んだ設計を行い、それに基づいた車載システムを構築した。その上で、いくつかの企業が実際に作成したアプリケーションを動作させることで、インターネットITSアーキテクチャの実現性を示した。

この結果、本実験システムはインターネットITS協議会成果報告会においてデモシステムとして利用していただいた。

また、本実験によって情報提供形態や通信状態通知機能の必要性など、解決すべき問題点を明らかにした。今後は、それら問題点をインターネットITS協議会に提供し、問題点の見直しと解決を図っていく予定である。

謝辞 本実験システムを構築するに当たり、多大なご協力をいただいた、インターネットITS協議会事務局ならびに参加企業の方々に深く御礼申し上げます。

参考文献

- [1] 遠山 祥広, 塚田 学, 植原 啓介, 砂原 秀樹, 村井 純. インターネット自動車のテストベッド構築と評価, November 2004. 情報処理学会研究報告 第6回コビキタスコンピューティングシステム, pp.37-43.
- [2] インターネット ITS 協議会, <http://www.internetits.org/>, August 2006.
- [3] WIDE プロジェクト, <http://www.wide.ad.jp/>, August 2006.
- [4] iCAR ワーキンググループ, <http://www.wide.ad.jp/project/wg/iCAR.html>. August 2006.
- [5] K.Uehara and Y.Watanabe and H.Sunahara and J.Murai. "The Internet-CAR -Internet Connected Automobiles-", Proc. of INET'98, July 1998.
- [6] K.Uehara and H.Sunahara and J.Murai. "The InternetCAR network architecture: Connect vehicles to the Internet using IPv6", in Proceedings of ITST2005, pp 187-190, June 2005.
- [7] ISO TC204/WG16/SWG16.3. *Vehicle Probe Data for Wide Area Communications*, ISO Committee Draft 22837.
- [8] 佐藤雅明, 植原啓介, 村井純, "インターネット自動車における情報取得 API の設計と実装", 情報処理学会第15回モバイルコンピューティングとワイアレス通信研究会 第3回高度交通システム研究会 合同研究会, pp137-pp144, November 2000.
- [9] インターネット ITS 協議会. インターネット ITS プラットフォーム仕様 第2版, April 2006.
- [10] Devarapalli Vijay and Wakikawa Ryuji and Petrescu Alexandru and Thubert Pascal. *Network Mobility (NEMO) Basic Support Protocol*. RFC3963.
- [11] Ryuji Wakikawa and Koshiro Mitsuya and Keisuke Uehara and Ernst Thierry and Jun Murai, "Basic Network Mobility Support for Internet ITS", IPSJ Journal Vol.44, No.12, pp2925-2395.
- [12] ISO TC22/SC3. *Road Vehicles -Controller area network(CAN)*, 2003, ISO 11898-1
- [13] ISO TC22/SC3. *Road Vehicles -Diagnostic systems-Part 1: Physical layer*, 2000, ISO 14230-1
- [14] OBD-II, <http://www.obdii.com/>, August 2006.
- [15] OSGi Alliance, <http://www.osgi.org/>, August 2006.
- [16] Knopflerfish Open Source OSGi, <http://knopflerfish.org/>, August 2006.