

インターネット自動車システムにおける情報取得APIの設計と実装

佐藤 雅明[†] 植原 啓介^{††} 村井 純^{†††}

インターネット自動車システムでは、自動車をインターネット上のノードとして捉え、情報化を行なう。また、通信とサービスを分離することで、自由な自動車情報の取得や自動車への情報提供、および自動車情報の利用を支援する。

本稿では、インターネット自動車システムによる自動車情報取得アプリケーションインターフェイスを設計・実装した。その際、自動車の停止の際にインターネットに設置された管理システムに情報を送る事で、車載計算機の電源断という状態を検知できるシステムを構築し、このシステムとの連係を行なった。その結果、本APIを用いることにより、特に意識することなくインターネットから自動車の情報を容易に取得することが可能となった。また、従来は取得することができなかった電源断中の自動車の状態を取得することが可能となった。

Design and Implementation of Application Interface for InternetCAR system

MASAAKI SATOU,[†] KEISUKE UEHARA^{††} and JUN MURAI^{†††}

In InternetCAR system, car is recognized as a node on the internet and get computerized. Also by separating networking and services, this system will support unrestricted aquirement of car information, provision of information to car, and utilization of the car information.

In this paper, the application interface which will aquire car information on InternetCAR system is designed and implemented. Also the system which will detect the shutdown of the machine by sending information to the controlling system on the internet on automobile stoppage is developed and hooked to InternetCAR system. In consequence, with the use of this API, it is now possible to get car information from the internet transparently. Also it became possible to get the state of the car when the machine is shutdown. This service was never available before.

1. はじめに

現在、自動車を取り巻く環境は大きく変化し、情報化されつつある。自動車の情報化、交通に関する総合的な情報交通システムは、様々な道路交通問題の根本的な解決手段として注目されている。

その実現形態として、自動車の情報化自体の研究が進められ、道路交通に関する総合的な情報通信システムである高度道路交通システム (Intelligent Transport System:ITS) の構築が世界規模で盛んに行なわれている。

しかし、ITS における自動車情報化の目的は、道路

交通問題の解決であり、これは交通のシステム化を主としている。自動車の動きを把握し、適切な情報を与えることで、交通流を制御するこの手法では、必ずしも自動車を単体として捉えてはいない。社会システム自体の情報化が盛んな現状では、自動車の持つ情報は様々な分野で有益に利用できる。交通という閉じた情報化は得策でない。

情報の収集・取得用の通信基盤をサービスごとに構築することは、コストの増大を招く。また、情報をサービスに特化した形で収集すると、設計用途以外への応用が難しい。これは、システム全体の情報化を進める上で非効率的である。

我々は、こうした問題を解決するため、インターネット自動車システム¹⁾を提案している。インターネット自動車システムでは、自動車をインターネット上のノードとして捉え、情報化を行なう。また、インターネット自動車システムは、通信部分を基盤システムとしてサービスから独立させることで、自由な自動車情

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

^{††} 慶應義塾大学 SFC 研究所
Research Institute at SFC, Keio University

^{†††} 慶應義塾大学 環境情報学部
Faculty of Environmental Information, Keio University

報の取得や自動車への情報提供、および他システムからの自動車情報の利用を支援する。

2. インターネット自動車システム

2.1 現 状

現在、慶應義塾大学インターネット自動車コンソーシアム²⁾、WIDE プロジェクト³⁾ インターネット自動車分科会では、インターネットに接続された自動車を基盤とした移動体通信環境に関する研究を行なっている。

インターネット自動車システムは、自動車の情報化において、通信とサービスを分離している。自動車への情報の提供や、自動車が持つ情報を他システムが利用できる基盤システムを独立させることで、自由な情報の交換を支援している。

また、自動車をインターネット上の一つのノードとして捉え、インターネット上から取り扱える環境を構築している。このため、インターネット自動車システムでは、インターネット上で動作する様々なソフトウェアを利用できる。これはインターネット自動車システムの大きなメリットである。

自動車は移動するため、一般的なインターネットとは異なる通信環境を持つ。有線通信機器の利用が可能な状況は限られており、通常は無線通信機器による接続性を確保しなければならない。しかし、無線通信環境は、基本的には有線に比べ低速高遅延であり、また、現状ではあらゆる環境で利用可能な無線通信機器は存在しない。そのため、安定した接続性を確保するためには、状況に応じて複数の通信機器を切替えて利用する必要が生じる。インターネット自動車システムでは、通信機器を通信状態に応じて自動的に切替える機構⁴⁾⁵⁾を用い、可能な限りの接続の継続を実現している。

地理的移動に伴い、ネットワーク上での移動も起こる。インターネット上では、IP アドレスによって通信ノードの識別を行なっている。しかし、IP アドレスはノードが接続しているネットワークに依存しているため、移動や複数通信機器を切替える際に、IP アドレスの変更が起こる可能性がある。IP アドレスの変化が起こると、移動中の自動車を一意に特定することが不可能になる。そのため、インターネット自動車システムでは Mobile IP⁶⁾ という移動透過性を持つ移動体通信用のプロトコルを利用している。Mobile IP を用いることで、自動車は常に不変の IP アドレスによる識別が可能となり、移動に妨げられずに継続した通信を行なうことができる。

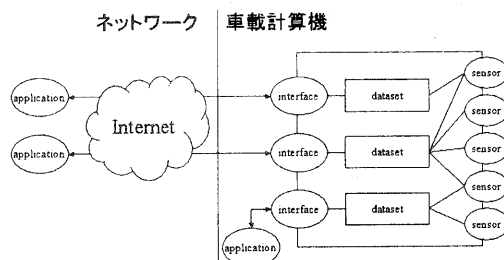


図 1 システム独自のインターフェイスによる情報収集
Fig.1 interface

世界の自動車の保有台数は約7億台である。これらを全てインターネットに接続しようとなると、現在のIP version 4 ではアドレスの枯渇を招く。そのため、インターネット自動車システムでは、システム全体のIP version 6 への移行を進めている。

2.2 現状の問題点

現状のインターネット自動車システムにおける問題点を以下に述べる。

2.3 自動車情報インターフェイスの不統一

現状における問題点として、自動車の持つ情報に対するアクセス手段が統一されていない点が挙げられる。現在、自動車の情報をインターネットから利用する場合には、システム毎に独自の通信インターフェイスを車載計算機とインターネット上のノードに用意している。現在の車載計算機上の情報システムを図1に示す。

この状態では、新規アプリケーションの実現には自動車上のシステム、インターネット上のノードのシステム双方を構築しなければならない。また、車載計算機が持つ情報がサービスに依存しているため、複数の自動車の情報を画一的に取得することは難しい。

これまでの研究で、自動車をノードとしてインターネット上から認識し、接続性の確保および通信の継続が可能となった。今後、インターネット自動車システム上で、誰もが自由にアプリケーションを開発できる環境を提供するためには、自動車の持つ情報に容易にアクセス可能とする、統一されたインターフェイスが必要である。

統一した情報の利用インターフェイスが確立されれば、それらを用いることでサービスシステムの開発コストを大幅に下げることが可能となる。また、情報の

加工・生成や交換、共有や再利用が容易になり。情報環境全体の価値を高めることができる。

2.4 車載計算機の電源断による通信の断絶

自動車に搭載されている車載計算機の電源断時の通信の断絶も問題である。車載計算機は、自動車の駐車等により電源供給が断たれ、頻繁に電源断が起こる。この時、インターネット上のノードと、車載計算機との通信は成立せず、自動車の持つ情報は一切取得することができない。この際の挙動は、無線通信機器の電波不感地帯に入っている時と同じである。そのため、アプリケーションは車載計算機の電源断を検知することが出来ない。自動車との通信が不成立だった場合、インターネット側からは、自動車が一次的に通信不能な状態にあるのか、それとも車載計算機の電源が断たれているのかを判断することができない。

通信状態による一次的な通信の断絶と駐車による通信の断絶は、情報として大きな隔りがある。また、“駐車”という状態を利用したいアプリケーションの実現は不可能である。この情報が取得できれば、例えば移動時のトンネルや地理的な問題による一次的な通信不通であれば通信を続けるが、駐車している車両、つまり電源が断たれているのであれば通信間隔を空けて自動車との通信回復を待つ、というようなアプリケーションが実現できる。また、駐車とその地理位置情報を関連付けて取得できれば、自動車の駐車位置を検出することが可能となる。

3. 情報取得 API の設計

前述の問題を解決する為に、本稿ではインターネット自動車システムにおける情報取得アプリケーションインターフェイス（以下情報取得 API）の設計を行なった。以下にその設計を述べる。

3.1 自動車情報インターフェイスへのアプローチ

自動車の持つ情報へのインターフェイスについては、車載計算機上に情報管理システムを構築し、統一したデータセットと通信インターフェイスを用いることとした。これにより、アプリケーションは自動車の状態に左右されない情報取得が可能となる。

自動車を持つ情報は、その種類、形式、単位等が統一されていない。また、自動車の種類や型式により、取得できる情報が異なる。これらを整理し、自動車情報のデータセットを作成する。自動車は、このデータセットに基づいた情報を持つとし、車載計算機上にデータセットを一元的に管理する情報管理システムを構築する。

情報管理システムは、センサから取得できる情報を

このデータセットに合わせて正規化する機能を有する。また、情報管理システムは、シンプルなメッセージセットからなる情報 API を介し、アプリケーションへ自動車の持つ情報を提供する。インターネット自動車システムでは、自動車の情報の取得に関しては、すべてこのインターフェイスを利用する。

この結果、アプリケーションは車載計算機上に新たな情報システムを構築することなく、自動車情報を取得できる。また、データセットとその正規化の部分を拡張することで、新しい情報の取得が可能となる。

データセットの形式は情報管理システムの大きなポイントである。どのような構造とするかや書式による性能の差異も生じてくる。しかし、本稿の焦点は情報取得 API である。従って、データセット作成時に、汎用性と拡張性を考慮はするが、データセット自体の妥当性や形式には触れず、規格化されたデータセットと、それに対する管理、および情報 API による情報提供機能を、車載計算機が持つ機能とした。

3.2 自動車の電源断検知へのアプローチ

電源断時の車両の情報をする方法としては、以下のようなアプローチが考えられる。

- 自動車のエージェントをインターネット上に用意し、それを介して通信を行なう
- インターネット上に状態管理システムを構築し、

電源断時にはこの機構が自動車の状態を保持する。前者の方法は、インターネット上に自動車の情報を保持するエージェントを作り、自動車は定期的に情報をエージェントに送信する。この場合、インターネット上のアプリケーションは、エージェントと通信することで自動車の情報を得る。自動車本体が通信不能な状態でも、最後に通信を行なった時点の自動車情報が取得できる。

しかし、システムとしてエージェントに依存する部分が出来てしまう。エージェントによって、自動車をノードとして捉え、自由な情報の交換を目指すシステム全体の設計に対して影響を与えることは好ましくない。

後者の方法は、インターネット上に自動車の状態を管理する機構を作り、自動車の駐車等による電源断をあらかじめ検知した車載計算機が、メッセージを送る。インターネット上のノードは、自動車との通信が不成立だった場合に、状態管理システムに問い合わせ、自動車の状態を得ることが可能となる。

この方法では、自動車内の情報管理システムに電源断時のシーケンスを追加する必要がある。また、自動車との通信不成立時にアプリケーションが状態管理シ

システムに自動車の状態を取得に行く必要が生じる。

しかし、本稿の場合では、情報取得 API を設計する上で、車載計算機の情報管理システムと、メッセージセットにこれらを組み込む事ができる。また、自動車とインターネットノードの通信にも影響を与えない。したがって、本稿では後者の方法を選択した。

3.3 設 計

上記のアプローチを基に、情報取得 API の設計を行なった。アプリケーションはこのインターフェイスを用いることで、自動車の状態に左右されない透過的な自動車情報の取得が可能となる。

情報取得 API は、自動車に搭載されている車載計算機上で自動車情報を管理する情報管理システム、インターネット上で電源断中の自動車の状態を保持する状態管理システム、及びこれらに対して情報取得を行なう為のメッセージセットである情報取得ライブラリからなる。

情報管理システム

情報管理システムは、図 2 のように、自動車上の各種センサから取得した情報を、データセットとして保持している。センサ情報は、情報管理システムが一元的に管理する。アプリケーションが、自動車の情報を取得する場合には、すべて情報管理システムのインターフェイスを通じて行なう。また、データセットは拡張が容易な設計とし、新たなデバイスからの情報も、データセットの拡張のみで対応できる。自動車の扱う情報が変化しても、データセットのみを更新すれば良い。このデータセットについては、ASN1.1 によって記述し、自動車の情報を MIB⁷⁾ として扱うことや、XML によって記述し、http によって情報交換を行なうという設計も考えられる。

状態管理システム

情報管理システムは、自動車の“駐車”または“エンジンの停止”を検知すると、電源断が起こると判断し、電源断メッセージを状態管理システムに通知する。状態管理システムは、自動車からの電源断メッセージを受けると、その自動車のエントリの状態を電源断とし、確認メッセージを情報管理システムへ通知する。情報管理システムでは、この確認メッセージを受け取るか、タイムアウトを待った後、終了処理を行ない電源を切る。この電源断処理の流れを、図 3 に示す。

車載計算機の電源が再投入された場合には、情報管理システムは状態管理システムへ状態の消去メッセージを送信し、状態管理システムはその自

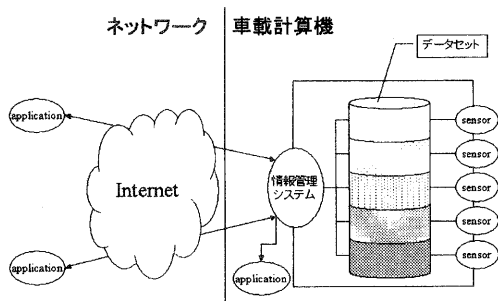


図 2 情報管理システムの概要
Fig. 2 design of information management system

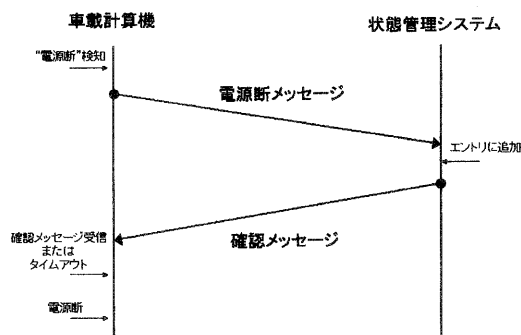


図 3 電源断処理の流れ
Fig. 3 power off process

動車のエントリを消去し、確認メッセージを送信する。それを受け、情報管理システムは通常の動作に移行する。

情報取得ライブラリ

アプリケーションが情報を取得する場合には、自動車情報取得ライブラリを利用する。このライブラリは、データセットに対応するメッセージセットである。アプリケーションは必要なデータセット部分に対応するメッセージに、自動車の ID、必要であれば状態管理システムの ID、および通信不成立の判断条件を与える。

このライブラリは、アプリケーションには情報管理システム、状態管理システムとの通信を隠蔽し、どちらと通信した場合でも同じように自動車の持つ情報をアプリケーションへ渡す。

情報取得時の流れ

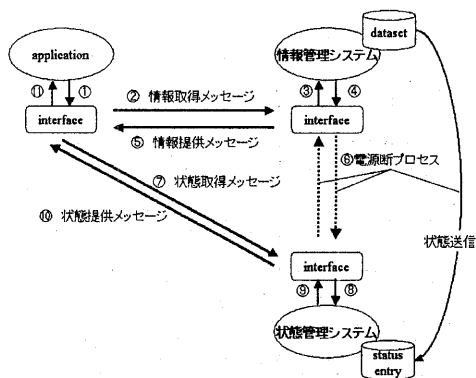


図 4 情報取得時の挙動

Fig. 4 behavior of Information acquisition

図 4 に、情報取得 API による情報取得時の流れを示す。

- (1) アプリケーションは、データセットの必要な部分に対応するメッセージセットに、取得したい自動車の ID と、状態管理システムの ID、および通信不成立条件を与える。
- (2) 指定された自動車の情報管理システムに対し、メッセージが送信される。
- (3) 車載計算機との通信が成立し、情報管理システムがメッセージを受けとる。
- (4) 情報管理システムは該当データ部分を提供メッセージに与える。
- (5) 車載計算機からアプリケーションに対し、情報提供メッセージが送信される。通常はここまでで処理が終了する。
- (6) 情報管理システムが電源断を検知した場合には、図 3 に示した処理が行なわれる。
- (7) 条件に従い、車載計算機との通信が不成立した場合には、自動的に状態管理システムに自動車の状態取得メッセージを送信する。
- (8) 状態管理システムは、状態取得メッセージを受けとり、管理する状態エントリに対し、該当 ID を検索する。
- (9) 該当する ID があった場合のみ、状態管理システムは状態提供メッセージを送信する。
- (10) 情報管理システム、状態管理システムどちらからメッセージが返ってきた場合でも、透過的にアプリケーションへ取得情報を返す。

なお、自動車との通信が不成立し、且つ状態管理システムのエントリに該当 ID が見つからない場

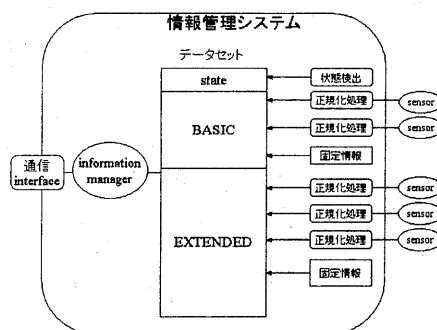


図 5 情報管理システムの実装

Fig. 5 implementation of information management system

合には、自動車の状態は“不明”である。状態管理システムは、情報管理システムの代わりはせず、アプリケーションは自動車との通信不能として処理を行なう。

自動車の電源断による挙動の変化を必要としないアプリケーションでは、ライブラリに状態管理システムを与えなければ、(7)-(10)の動作は行なわれない。

自動車の電源断を検知した時に自動車の通信が不可能だった場合には、情報管理システムはタイムアウトまで電源断メッセージの送信を繰返し、それでも確認メッセージが得られない場合には、そのまま電源断を行なう。この場合、状態管理システムには自動車の状態が残らないため、アプリケーションは自動車の状態を取得できない。これは、事実と異なる状態を状態管理システムが返す事で、実際の自動車とインターネット上のノードとの通信に干渉するのを防ぐためである。状態管理システムによる電源断の判断や予測等は一切行なわない。

4. 情報取得 API の実装

本章では前述の設計に基づき、情報管理システム、状態管理システムおよび情報取得ライブラリからなる情報取得 API の実装について述べる。

4.1 情報管理システム

情報管理システムの実装は、図 5 のように行なった。本実装では、自動車のデータセットは、基本セットとなる“BASIC”と拡張セットである“EXTENDED”に分けて整理を行なった。これはインターネット自動車として最低限保持する情報の定義と、車両の個体差や将来的なデータセットの拡張に、情報管理システム

を対応させる為である。また、データセットは C 言語の構造体とし、各データについては、C 言語の変数として定義した。

基本セットについて情報管理システムは必ず対応し、そのデータは保持する。拡張セットについては、状況に応じて情報管理システム上で選択を行なうことができる。情報管理システムは、各センサやファイルなどから取得できる情報を、データセットに定義されている形式へ正規化する。基本データセットは以下の通りである。

carid 車両の識別子を示す。本実装では FQDN で車両を表す。

status 車両の状態を示す。INTEGER で 0 なら電源断、1 ならば動作中を表す。

timestamp 情報の更新時間を示す。unsigned long である time_t で表す。

latitude 車両の緯度を示す。値を"度.分.秒"で表し、それぞれ INTEGER で示す。南緯は負の値で表す。

longitude 車両の経度を示す。値を"度.分.秒"で表し、それぞれ INTEGER で示す。西経は負の値で表す。

altitude 車両の高度を示す。INTEGER で表し、単位は m とする。

speed 車両の速度を示す。INTEGER で表し、単位は km/h とする。

拡張データとしては、これに加えてヘッドライト情報とワイパ情報、および自動車の車幅、車高、車重を示す"width"、"height"、"weight"を加え、この3つを固定情報とした。固定情報とは、自動車の移動や時間の変化に関係ない情報と定義した。拡張データセットは以下の通りである。

light 車両のヘッドライト点灯状況を示す。INTEGER で 0 なら消灯、1 ならスモール、2 なら通常点灯、3 ならハイビームをそれぞれ表す。

wiper 車両のワイパ動作状況を示す。INTEGER で 0 なら停止、1 ならインターバル、2 なら low、3 なら hi のポジションであることをそれぞれ表す。

width 車両の車幅を示す。単位は mm とする。

height 車両の車高を示す。単位は mm とする。

weight 車両の重量を示す。単位は kg とする。

これらデータフォーマットの形式をまとめたものを、表 1、および表 2 に示す。

車載計算機としては、インターネット自動車システム用に開発した車載ハードウェア SIC2000⁸⁾を用いた。主な仕様は表 3 の通りである。

表 1 BASIC データセット

Table 1 "BASIC" Car information data set

BASIC	
データ名	形式
CarID	string(FQDN)
status	unsigned int
timestamp	unsigned long time_t
Latitude	signed short.signed short .unsigned short
Longitude	signed short.signed short .unsigned short
altitude	signed long
speed	signed short

表 2 EXTENDED データセット

Table 2 "EXTENDED" Car information data set

EXTEND	
データ名	形式
light	signed short
wiper	signed short
width	unsigned short
height	unsigned short
weight	unsigned short

表 3 SIC2000 本体ハードウェア構成

CPU	VR4300
メインメモリ	DIMM (64MB)
起動プログラム格納装置	EPROM (512KB)
オペレーティングシステム格納装置	FROM (8MB)
システムバス	ISA バス
シリアルインタフェース	RS-232C(4 個), RS-422
ネットワークインタフェース	10BASE-T
拡張スロット	PCMCIA 4 スロット

SIC2000 は自動車での利用を考慮し、DC12~24V の電源で動作する。また、電源を 2 系統持ち、自動車のイグニッションキーとの連動による電源制御が可能である。また、内部コンデンサにより外部からの電源供給が断たれても数分間の動作が可能である。

本実装では、これを基に、イグニッション OFF の状態が 30 秒続いたら電源断が起こると見なし、情報管理システムは電源断処理を行なう。電源断メッセージを送信する状態管理システムについては、既知のインターネット上の固定ノードとした。電源断メッセージには、自動車の IP アドレス、タイムスタンプ、位置情報および固定情報を送ることとした。タイムアウトに関しては、1 分とした。

実装には C 言語を用い、コンパイラは gcc ver-

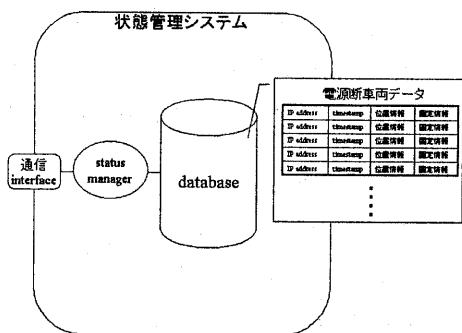


図 6 状態管理システムの実装
Fig. 6 implementation of status management system

sion2.95.2、開発には FreeBSD4.1R を用いた。SIC2000 の OS は NetBSD 1.4.1L であるため、クロスコンパイルを経て使用した。

4.2 状態管理システム

状態管理システムの実装は、図 6 のように行なった。

実装には C 言語を用い、コンパイラは gcc version2.95.2、開発には FreeBSD4.1R を用いた。本実装で状態管理システムで管理する情報は、自動車の IP アドレス毎に、IP アドレス、time stamp、緯度、経度、高度、および固定情報とした。今回の実装では、状態と共に地理位置情報と電源断時間を関連付けて保持することで、アプリケーションは情報取得 API を用いることで、電源断の位置と時間を得ることが可能である。また、固定情報に関しては、車載機の電源状態に関わらず取得が可能である実装とした。

5. 情報取得ライブラリ

情報取得ライブラリの実装は、C 言語のライブラリとした。実装には C 言語を用い、コンパイラは gcc version2.95.2、開発には FreeBSD4.1R を用いた。

このライブラリは以下のメッセージから構成されている。

car_info_acquisition_basic 自動車の基本データセット情報全てを取得するメッセージ。

car_info_acquisition_all 自動車の基本及び拡張データセット情報全てを取得するメッセージ。

car_status_acquisition.all 自動車の状態情報を取得するメッセージ。通常は自動的に行なわれる為選択しない。

car_info_acquisition 自動車データセットの任意の 1 情報を取得するメッセージ。

これらメッセージに、情報取得を行ないたい自動車の IP アドレス、状態管理システムの IP アドレス、通信不成立の条件を引数に与える。car_info_acquisition に関しては、さらにデータ名を与える。今回の実装では判断不成立の条件は時間とし、与えられた時間内 (msec で指定) に応答がない場合には、状態管理システムに問い合わせを行なう。

今回の実装では、状態管理システムからの応答には、電源断時の時刻および地理的位置、固定情報が含まれる。図 7、図 8、および図 9 に情報取得 API の処理の流れを示す。

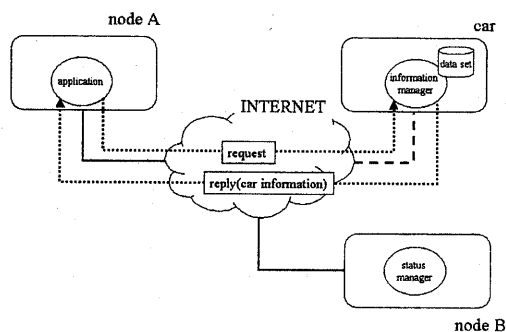


図 7 自動車との通信確立時
Fig. 7 3

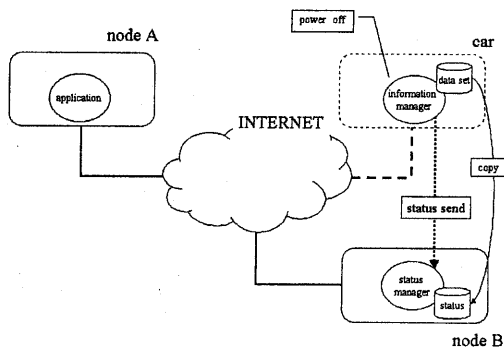


図 8 自動車の電源断による状態管理システムへの状態通知
Fig. 8 4

6. 考 察

本実装を利用して自動車の情報を取得する簡単なア

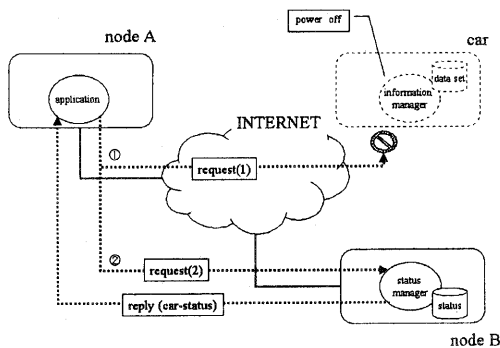


図 9 自動車との通信不成立時の自動車状態の取得
Fig. 9 5

アプリケーションを構築し考察を行なった。実験には、前節で述べた車載計算機である SIC2000 を搭載したインターネット自動車を利用した。通信機器には 9.6kbps の PDC-P と 2Mbps の 802.11 準拠の無線 LAN の 2 種類を用いた。楕円状の大学構内の周回道路のを無線 LAN エリアと PDC-P エリアに分かれるように、無線 LAN 基地局を設置し、それぞれについてインターネット上のノードから自動車の情報を取得した。ネットワーク環境はアプリケーションと状態管理システム、および無線 LAN は同一セグメント上にあり、PDC-P のダイヤルアップポイントは別セグメントに存在する。この間の RTT は平均 2.2msec 程度であった。実験に用いた計算機の仕様を表 4 に示す。

表 4 実験環境

	状態管理システム	アプリケーション
CPU	Pentium Pro(200)	MMX PentiumII(266)
OS	FreeBSD 3.4	FreeBSD 4.1 + KAME + PAO
MEMORY	130Mbyte	196Mbyte
NIC	10BASE-T	100BASE-T

RTT は、無線 LAN 時で平均 15msec、PDC-P 時で平均 724msec であった。この環境において実験を行なったところ、無線 LAN、PDC-P の双方で自動車の情報を取得することが出来た。また、自動車の電源断状態の取得が出来た。この時、無線 LAN 使用時のパケットロス率は 1%未満であった。また伝送速度も十分に早く状態管理システムへの通知も正常に動作したため、この環境下ではほぼ完全に自動車の状態を取得することが可能である。PDC-P での利用時には、パ

ケットロスが 9%程度あったが、電源断状態の取得も可能であり、十分に実用に耐える結果となった。

7. まとめと今後の課題

本実装により、アプリケーションが、統一したインターフェイスにより自動車から情報取得を行なうことができた。また、車載計算機が電源断である自動車の情報を取得することができた。

今後の課題としては、データセットを再設計し、規格化・標準化を行なうことが挙げられる。今回の実装結果を基に、広く使えるデータセットをインターネット自動車システムで利用していく必要がある。書式として、ASN.1 や XML 等を検討することも課題である。また、状態管理システムの動的更新を実現することで、例えば駐車場に入ったら、その駐車場の状態管理システムに登録する、といったことが可能となる。しかし、アプリケーションが通信時に状態管理システムのアドレスを取得する必要があるため、検討が必要である。

謝辞 本研究を進めるにあたり、議論や実験等に協力して頂いた慶應義塾大学 環境情報学部 徳田・村井・楠本・中村研究室の皆様へ感謝致します。

また、本論文を執筆する上で様々な議論や貴重な助言を頂きました WIDE プロジェクト、インターネット自動車プロジェクトの皆様へ感謝の意を表します。

参 考 文 献

- 1) Uehara, K., Watanabe, Y., Sunahara, H., Nakamura, O. and Murai, J.: InternetCAR - Internet Connected Automobiles -, INET'98(1998).
- 2) インターネット自動車コンソーシアム
<http://www.sfc.wide.ad.jp/InternetCAR/>
- 3) WIDE Project. <http://www.wide.ad.jp/>
- 4) 砂原秀樹、比良木貴史、植原啓介、尾家祐二: 移動対端末装置における通信インターフェイスの自動選択機能の実現, DiCoMo, 1997.
- 5) 湧川隆次、植原啓介、田村陽介、徳田英幸: MIB-socket: 移動型計算機におけるネットワークエンティティの状態変化に対応する管理機構の設計と実装, 情報処理学会 システムソフトウェアとオペレーティングシステム研究会, 1999.
- 6) C.Perkins.: IP Mobility Support, RFC:2002, 1996.
- 7) M.Rose., K.McCloghrie.: Concise MIB Definitions, RFC:1212, 1991.
- 8) 杉本信太、植原啓介、三屋光史朗、村井純: 車載コンピュータへの BSD の応用、第 3 回プログラミングおよび応用システムに関するワークショップ (SPA2000), 2000.