

インターネット自動車用 アプリケーションソフトウェア試験環境の構築

植原 啓介[†] 砂原 秀樹^{††} 村井 純^{†††}

インターネット自動車用のアプリケーションソフトウェアは、インターネットに接続された自動車という特殊な環境下で利用されるため、現在は実フィールドを使って試験されている。このため、試験にかかるコストが非常に大きい。本研究では、自動車用アプリケーションの置かれている環境をモデル化し、このモデルに基づき環境シミュレータと自動車エミュレータから成る試験環境を構築した。構築した試験環境は、インターネット自動車用アプリケーションソフトウェアの試験を行うのに、十分スケラブルで機能的であった。

Design and Implementation of a Testing Environment for InternetCAR Application Software

KEISUKE UEHARA,[†] HIDEKI SUNAHARA^{††} and JUN MURAI^{†††}

Application softwares for InternetCAR are used in the special environment where automobiles are connected to the Internet. For that reason, the softwares are tested in actual automobiles. Therefore, the cost for testing is very high. The model of the environment for InternetCAR applications is constructed. The testing environment is developed based on the model in this research. The testing environment consists of environment simulator and automobile emulator developed based on the model. Developed testing environment for InternetCAR application softwares was sufficiently functionable and scalable for testing.

1. はじめに

現在、高度交通システム (Intelligent Transport Systems: ITS) の分野、特に自動車内マルチメディアシステムは、新しい流れを迎えている。カーナビゲーションシステムに代表される車載計算機を計算機ネットワークに接続することにより、より柔軟な情報環境を構築しようという試みである。この流れは、計算機ネットワーク上で流通する情報を車内で利用できるばかりではなく、自動車を含めた新たな計算機ネットワーク環境の実現を可能にする。

インターネット自動車プロジェクト^{1),2)}でも、現在、自動車をインターネットに接続し、自動車の情報化を支援するためのプラットフォームを構築する研究を進めている。このようなプラットフォームを構築することに

より、多くの人々が自動車を利用したアプリケーションを低コストで実現することが可能となり、自動車の情報化を加速することが可能となる。インターネット自動車プロジェクトでは、自動車をインターネットに接続するためのネットワーク機構、インターネットに接続された自動車を利用するためのアプリケーションアーキテクチャなどについて研究を進めている。

本稿では、インターネット自動車プロジェクトにおける成果の1つである、アプリケーションソフトウェア試験環境について述べる。自動車の情報化を加速するためには、インターネットによる社会の情報化がそうであったように、一般の人々がアプリケーションのアイデアを出し、これを具現化できる環境が必要である。このためには、アプリケーションソフトウェアの開発コストを十分に下げなければならない。この問題を解決すべく、インターネット自動車プロジェクトでは、自動車用アプリケーション実現のためのプラットフォーム構築を行ってきた。これにより、アプリケーションソフトウェア開発で一番コストが大きな部分は、アプリケーションの試験となった。現在、自動車を利用したアプリケーションでは、実車両を使用した試験

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

^{††} 奈良先端科学技術大学院大学
Nara Institute of Science and Technology

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

が必要となる。これは、自動車用のアプリケーションが、1) 自然環境の中を 2) 社会秩序に従って 3) 移動する 4) 多数の 5) 自動車という特殊な環境を対象として構築されているためである。しかし、開発の初期段階から自動車を用いて試験を行うことはコスト的に難しい。そこで、開発したアプリケーションを可能な限り机上において試験できる環境の構築が望まれている。本研究では、このような試験環境を提供するために、アプリケーションが置かれている環境モデルを構築し、これを利用した試験環境を開発する。

本稿では、初めに現在の問題点を洗い出し、本研究の目標を定める。次に自動車が置かれている環境について考察し、システムのモデルを構築する。残りの部分では、システム的设计・実装について述べ、実際に実装したシステムの評価検討を行う。

2. 研究の目標とその前提

本章では、研究の目標について述べた後、その前提となるインターネット自動車用アプリケーションの特殊性について述べる。

2.1 現状の問題点と本研究の目標

インターネット自動車プラットフォームは、自動車を利用したアプリケーションを低コストで提供することを可能とし、自動車の情報化を加速することを目標としている。しかし、実際にアプリケーションを構築するためには、インターネット自動車プラットフォームが提供するアプリケーションプログラミングインターフェイス (Application Programming Interface: API) に沿ってアプリケーションソフトウェアを作成し、その試験を行う必要がある。現状では、作成したアプリケーションソフトウェアを試験するためのコストが非常に大きい。

現在、インターネット自動車用アプリケーションソフトウェアは、次のような手順で開発される。

- (1) アプリケーションの設計
- (2) インターネット自動車 API を使ったアプリケーションソフトウェアの実装
- (3) 1 台の自動車を使った機能試験
- (4) デバッグ ((3) へ)
- (5) 複数の自動車を使った性能試験
- (6) デバッグ, 調整 ((3) または (5) へ)
- (7) 運用

これらの開発手順の中で、(3) から (6) までの繰返しのコストが特に大きい。これは実際に自動車にソフトウェアをインストールし、稼働させる必要があることに起因する。しかし、実際にはそれぞれの試験に

も段階があり、必ずしもはじめから実車両を用いる必要はない。API の試験やある程度の動作確認に関しては、机上で行える環境が提供されるべきである。

もう 1 つの問題として、複数の自動車を使った試験が難しいことをあげることができる。アプリケーションの開発のために多数の実車両をあらかじめ準備しておくコストは非常に高く、一般の人々がアプリケーションを構築する妨げとなる。また、複数の車両を用意できたとしても、通信環境や取得可能なセンサ情報、各自動車の挙動などの組合せが多く存在し、これらを考慮した試験を行うことは事実上不可能である。

本研究では、このような問題を解決するためのインターネット自動車用アプリケーションソフトウェア試験環境を構築することを目標とする。

2.2 インターネット自動車用アプリケーションの特殊性

インターネット自動車用アプリケーションの定義は、「自動車内と自動車外で動作するソフトウェアがインターネット自動車プラットフォームを介して協調動作するアプリケーション」である。これらのアプリケーションは、次に示すようないくつかの特徴を備えている。

はじめに、自動車という特殊な環境の中で動作するアプリケーションである点をあげることができる。自動車環境では、運転手や同乗者がユーザとなりうる。このため、自動車側のユーザが積極的に介在するようなアプリケーションは考えにくい。

次に、アプリケーションが自然環境や社会秩序などの周辺環境に強く依存することをあげることができる。自動車は潜在的に 100 以上のセンサ情報を持っており³⁾、これらの情報を有効利用したアプリケーションを構築することができる。また、自動車は道路上を秩序を持って移動し、道路状況などにその挙動が左右される。

3 つ目として、自動車は本質的に移動するものである点をあげることができる。この特徴のため、先に述べた周辺環境の変化はさらに複雑になる。自動車自身が移動するため、通信環境などの本来ならそれほど変化しない周辺環境も大きく変化することを考慮しなければならない。

最後に、社会に存在する自動車の膨大な数をあげることができる。1997 年現在、日本国内には約 7300 万台⁴⁾ の自動車が存在している。インターネット自動車用アプリケーションでは、多数の計算機 (自動車) が利用可能となることを考慮しておく必要がある。

本研究では、上記の事項をふまえた試験環境を実現するための一手法を提案する。

3. モデル構築

本章では、前章で整理したアプリケーションの特徴を基に、アプリケーションが置かれている環境のモデルである iCAR モデルを構築する。その後、各部を構築するためのアプローチについて述べる。

3.1 iCAR モデル

前章であげた特徴を整理すると、インターネット自動車用アプリケーションは、環境情報（自動車外部の情報）、車両状態（自動車内部の情報）、通信環境（車外とのインタフェース）を利用していることが分かる。また、ユーザの介在はあるものの、車内における操作は限られたものとなる。つまり、構築する試験環境には、環境情報提供機能、車両状態提供機能、通信環境制御機能の3つが必要となることが分かる。

これらの機能は、以下の2つの情報を基にして実現することが可能である。

- 自動車周囲の情報
基本的には気象などの自動車とは直接関係のない情報。ただし、渋滞状況などの情報も自動車周囲の情報と見なす。
- 自動車内部の情報
自動車がなんらかのセンサで取得した情報やブレーキやライトのオンオフ、アクセル開度などの自動車の走行に関わる情報。

環境情報提供機能は、「自動車周囲の情報」を自動車のセンサを通して（センサ誤差などを加味して）取得した「自動車内部の情報」に加工し、これをアプリケーションソフトウェアに提供する機能にほかならない。また、車両状態提供機能は「自動車内部の情報」のうち「自動車周囲の情報」ではないものをアプリケーションソフトウェアに提供する機能である。通信制御機能は「自動車周囲の情報」のうちの通信に係るものを考慮し、アプリケーションソフトウェアが行う通信を制限する機能である。

以上のことをふまえると、今回構築する試験環境は図1に示すようなモデルで示される。これを、インターネット自動車のアプリケーションが置かれる環境をモデル化したものという意味で iCAR モデルと呼ぶ。iCAR モデルでは、自動車周囲の情報を環境シミュレータで、自動車内部の情報を自動車エミュレータで生成している。さらに、自動車エミュレータは、環境シミュレータに対して位置情報とともに環境情報を要求し、この情報を利用して自動車のセンサ情報を生成している。このため、iCAR モデル上には2種類の自動車が存在することになる。1つは道路状況を生成す

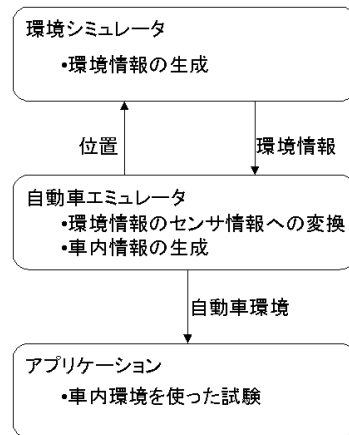


図1 iCAR モデル
Fig. 1 iCAR model.

るための雑音生成用の自動車（環境シミュレータとして実装される自動車）、もう1つは実際にインターネット自動車として扱われる自動車（自動車エミュレータとして実装される自動車）である。前者は、単なる道路状況を生成する手段であるため、後者の自動車と比べると単純なモデル化で表現することが可能である。また、アプリケーションソフトウェアは、自動車エミュレータ部からのみ情報を取得して動作する。

本研究では、iCAR モデルに基づいて試験環境を構築するものとする。

3.2 環境シミュレータ構築のアプローチ

iCAR モデルにおける環境シミュレータは、さまざまな自動車周囲の情報を生成する。それぞれの情報は、本来、論理モデルに基づいたシミュレーションを行ったうえで生成される必要がある。しかし、実際には厳密に自動車周囲の環境をシミュレーションする必要はなく、インターネット自動車用アプリケーションの試験が可能な程度に現実的な情報が生成できればよい。現段階では精度の悪い論理モデルを使っているが、将来、より負荷が少なく精度の高い論理モデルが構築されたときに、そのモデルへの移行が簡単に行える構造になっていることが重要となる。

一方、インターネット自動車においては、自動車に接続するセンサの加減により、取得できる周囲の情報の種類も増減する。このことを鑑み、環境シミュレータ部は生成できる自動車周囲の情報を簡単に追加できる構造になっている必要がある。

今回は、上記の2つの条件を満たすため、複数のシミュレーションモジュールで環境シミュレータを構成する。各シミュレーションモジュールは1つの現象について独立にシミュレーションを行う。環境シミュ

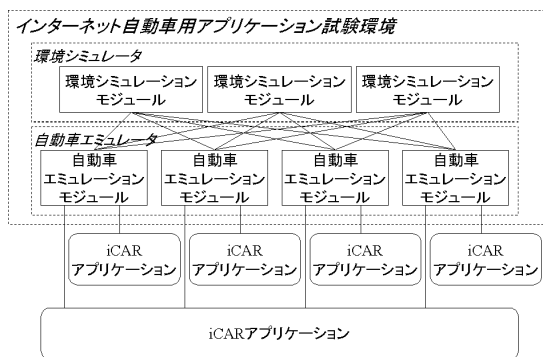


図 2 システムモデル

Fig. 2 System model.

レータをシミュレーションモジュール別に分割することにより、各分野で研究されている論理モデルを基に、独立して各モジュールを構築・改良することが可能となる。

3.3 自動車エミュレータ構築のアプローチ

iCAR モデルにおける自動車エミュレータは、環境シミュレータからの情報を利用し、個々の自動車の状況をエミュレーションする部分である。アプリケーションソフトウェアと試験環境の唯一のインターフェースでもあり、アプリケーションに渡されるすべての情報はここに集約される。

自動車エミュレータ部は自動車の挙動を可能な限り忠実にエミュレーションする必要がある。アプリケーションが 1 台の自動車の挙動に着目することも考えられ、それぞれの自動車が一貫性を持った挙動をしなければならない。このため、自動車エミュレータ内部では、人間の行動モデルを構築し、それに従って走行する必要がある。

今回のアプローチでは、個々の自動車を独立してエミュレーションする手法をとる。本来、前の自動車がブレーキをかけたために後続車両もブレーキをかけるなどの関係がある。しかし、このような関係は、インターネット自動車の絶対数が増え、しかも、大多数の自動車の情報が取得できる場合にのみアプリケーションにとって有効な関係である。このため、システムの簡素化を重視し、個々の自動車を独立したエミュレーションモジュールとして構築する。

3.4 システムモデル

以上をまとめると、インターネット自動車用アプリケーションの試験環境のシステムモデルは図 2 に示すようなものとなる。

図では、3 つのシミュレーションモジュールが独立に動作している。また、各車両の自動車エミュレーシ

ョンモジュールが、それぞれのシミュレーションモジュールから情報を取得し、センサ誤差などを加味したうえでアプリケーションに提供している。実際の運用では、各シミュレーションモジュールがシミュレーションする地域、各自動車エミュレーションモジュールが走行をエミュレーションする地域を一致させておく必要がある。また、各シミュレーションモジュールおよびエミュレーションモジュールの時間の進み方も合わせておく必要がある。時間に関してはアプリケーションの実装方法の問題もあり、実時間に合わせるのが最適である。一方で、試験時間を短縮したい場合などに時間を圧縮する手法は試験の効率化に有効である。

4. システム設計

本章では、前章で述べたモデルに従って実際にシステムを設計する。初めに、システムモデルを実現するために必要となる各部間のインターフェースについて述べた後、個々の部位の詳細な設計について述べる。

4.1 各部間のインターフェース

初めに、アプリケーションソフトウェアと自動車エミュレーションモジュール間のインターフェースについて検討する。アプリケーションの試験では、試験のためのアプリケーションソフトウェアの変更は最小限にとどめる方が望ましい。そこで、今回の実装では、アプリケーションソフトウェアと自動車エミュレーションモジュールのインターフェースは、実際のインターネット自動車 API をそのまま利用するものとする。現在、インターネット自動車プラットフォームでは、自動車が取得した情報を Management Information Base (MIB) として定義し、Simple Network Management Protocol (SNMP) で提供することを検討している。つまり、本試験環境でもこの仕様に沿い、アプリケーションソフトウェアに対して SNMP で情報を提供する。アプリケーションソフトウェアは、自動車の IP アドレスと目的の MIB を指定することにより、自動車の情報を取得することが可能となる。

環境シミュレーションモジュールと自動車エミュレーションモジュール間のインターフェースは試験環境に閉じたインターフェースであり、性能的に著しく問題がなければ自由に実装することができる。今回は要求応答型の独自のプロトコルを策定するものとする。

4.2 環境シミュレータの設計

今回の実装では、交通流シミュレーションモジュール(交通流シミュレータ)、気象シミュレーションモジュール(気象シミュレータ)、無線通信環境シミュレーションモジュール(無線通信環境シミュレータ)

の3つを実装する。前者の2つを実装することにより、道路に関するもの、空間に関するものの双方の構築可能性を検証することができる。また、無線通信環境シミュレータを実装することにより、自動車エミュレータの通信制御部の実装が可能となる。本節では、各シミュレーションモジュールの詳細設計について述べる。

4.2.1 交通流シミュレータ

まず、環境シミュレータの中でも特に重要である交通流シミュレータの設計について述べる。交通流シミュレータは、地図情報を基にシミュレータ時間で道路状況をシミュレーションする。実際には道路のトポロジによって交通流を適当に増減させ、現実に近い交通状況を生成する。

交通流シミュレータを構築するにあたり、交通流のモデルを作成する必要がある。今回構築すべき交通流シミュレータは、1台1台の自動車を正確にシミュレーションする必要はなく、大方の交通状況を生成できればよい。そこで、交通流に関して以下のような前提を置くものとする。

- 地図上には混雑している地域が存在する。
- 交差点における車両の入出量は保存される。
- 各道路の情報は速度で代表される。
- 車両の増減は道路の適当な場所で発生する。
- 道路には車線数、幹線などの属性が存在する。

これらの前提をふまえ、今回作成するシミュレータを2つの部分に分解する。概念図を図3に示す。2つの部分のうち、片方はシミュレーションする地域を矩形に区切り、その中の状態を大まかにシミュレーションするマクロシミュレーション部、もう片方は矩形内の各道路の状況をシミュレーションするミクロシミュレーション部である。

今回の設計では、マクロシミュレーション部はセルラオートマトンによって構成する。各矩形(セル)の状態を、自動車の混雑度(矩形における自動車の存在数/矩形における存在可能な自動車の総数)で表現する。あるセルの時刻 t_1 の状態は、注目するセルとその周りのセルの時刻 t_0 の状態によって決定される。マクロシミュレータ内の状態変化の単位時間は、現実社会での渋滞の変化がそれほど早くないため、10分~数10分程度で十分である。

一方、ミクロシミュレーションの方は道路地図を有効グラフとして表現し、その中を流れる粒として自動車をモデル化し、シミュレーションを行う。図4に示すように、交差点に流入出する自動車を確率的に計算し、道路状況をシミュレーションする。交差点では、

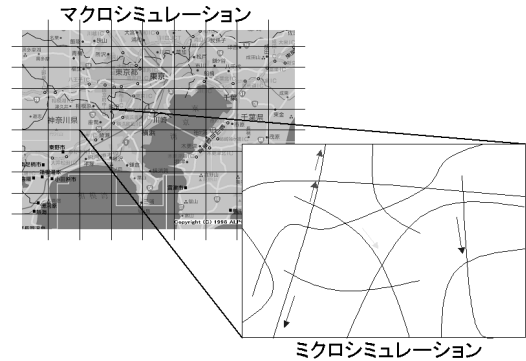


図3 交通流シミュレータ—マクロシミュレーションとミクロシミュレーション

Fig. 3 Traffic simulator—macro simulation and micro simulation.

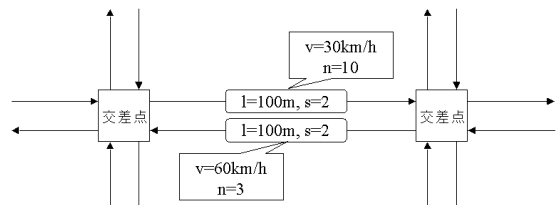


図4 ミクロシミュレーション部の考え方

Fig. 4 Micro simulation model.

式(1)が成り立つ必要がある。式で $flow_{in}$ は交差点への流入量を、 $flow_{out}$ は交差点からの流出量を表している。また、 l は道路の長さを、 v は道路を走る自動車の平均速度を、 n は存在している自動車の台数を示している。

$$flow_{in} = \sum \frac{v \cdot dt}{l} n = flow_{out} \quad (1)$$

また、道路の車線数を s とすると、式(2)が成立することが知られている⁶⁾。本シミュレータでも、各道路の平均速度の算出にこの式を利用するものとする。

$$v = f(l, s, n) \quad (2)$$

4.2.2 気象、無線通信環境シミュレータ

気象シミュレータおよび無線通信環境シミュレータは、双方とも空間の情報を生成するシミュレータである。それぞれ、大気の状態などを考慮した複雑な論理モデルが存在するが、本システムはインターネット自動車用アプリケーションの試験環境であることをふまえ、より簡単なモデルを利用するものとする。

気象シミュレータに対する要求は、現実的なサイクルで雨量や気温が変化する環境の生成である。また、雨が降っている場所の移動なども考慮する必要がある。気象は、基本的にエリアによって状態が決定する事象

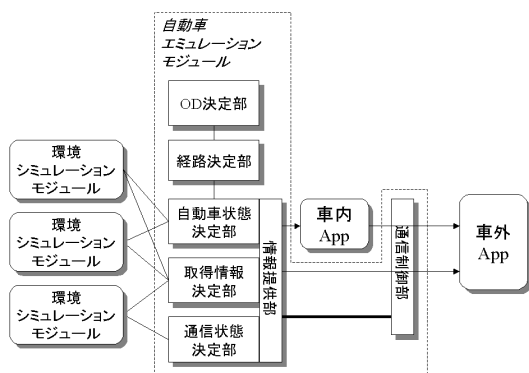


図5 自動車エミュレーションモジュールブロック図

Fig.5 Block diagram of car emulation module.

であり、今回はこの特長を利用し、交通流シミュレータのマクロシミュレーション部のようにセルに区切って状態を決定することにする。実際には、時刻 t_0 のセルの値とその周りのセルの値、新しく算出した乱数の3つを使って、時刻 t_1 の状態を決定する。再計算の時間間隔も交通流シミュレータのマクロシミュレーション同様、10～数十分間隔とする。

無線通信環境シミュレータも、気象シミュレータ同様、エリアごとに環境情報を生成する。しかし、無線通信環境は時間による変化がほとんどないため、シミュレータ起動時に各エリアの状態を決定し、その後は自動車エミュレータからの要求に応える動作だけを行う。

4.3 自動車エミュレータの設計

本節では、自動車エミュレーションモジュールの実装について述べる。

自動車エミュレーションモジュールは、環境シミュレータからの情報を基に、個々の自動車の挙動を模倣する部分である。モジュールの構成を図5に示す。モジュールは7つの部分から成っており、互いに協調して動作している。OD決定部では、自動車の出発点と目的地を決定する。これを受けて経路決定部が経路を決定し、実際の走行を車両状態決定部でエミュレーションする。取得情報決定部は環境エミュレータから受け取った環境情報にセンサ誤差を加え、自動車が持つセンサ情報を生成する。通信状態決定部は無線通信環境シミュレータから得た情報を基に、車のインターネットアクセス線の通信状態を決定する。以上の決定部によって作られた情報は、情報提供部と通信制御部を通してアプリケーションソフトウェアの試験に利用される。

車両エミュレータは、試験で使う自動車の台数分だけの自動車をエミュレーションしなければならない。

今回は、拡張性を考慮し、1台につき1つのプロセスを割り当てることにした。これにより、多数の自動車を利用して試験を行う際にも、計算機を何台使用するかを問わず、必要な分のプロセスを起動することで対応できる。

4.3.1 OD決定部と経路決定部の設計

本来、OD情報を決定するためには、人間の行動モデルや人口分布などを考慮しなければならない。しかし、このような社会全体を忠実に再現できるモデルを仮定するのは現実的ではない。そこで、今回の設計では、OD情報を乱数により決定するものとした。出発点と目的地を乱数で決定し、各々の点から一番近い道路上の点を改めて出発点と目的地とする。

出発点と目的地が決まれば、経路を決定することができる。経路を選択する方法は、すでに、カーナビゲーションシステムなどでさまざまな方法が用いられている。しかし、今回は、1) 利用している地図の制限により、幹線道路が分からない、2) 負荷軽減のため、システムの単純化を考慮する必要がある、という2つの理由により、出発点から目的地までの最短経路を経路決定部の出力とすることとした。

4.3.2 車両状態決定部の設計

車両状態決定部は、経路決定部と交通流シミュレータの出力を受けて、自動車の走行状態を管理する部分である。時刻が進むごとに自動車を経路にそって前進させ、自動車の情報(位置を含む)を更新する。車両状態決定部は以下のような手順で実行される。

- (1) 交通流シミュレータから道路の平均速度を取得し、単位時間分の移動距離を算出する。
- (2) 車両の位置を経路に従って移動する。
- (3) 目的地に達したら、OD決定モジュールと経路決定モジュールを呼び出し、新たな経路を算出する。
- (4) 単位時間待機し、(1)に戻る。

経路決定部、車両状態決定部および交通流シミュレータは同じ道路地図を共有しなければならないことに注意が必要である。

4.3.3 取得情報決定部の実装

取得情報決定部は環境シミュレータから取得した情報にセンサ誤差を加える部分である。降雨情報を取得した場合は人間の感覚にあった関数を利用してワイパの動作強度に反映したり、外気温情報の場合は自動車に設置されている外気センサの特長を踏まえてアプリケーションに提供したりする。設計としては、アプリケーションに提供する情報ごとに生成関数を持ち、これに従って値を算出する。

4.3.4 情報提供部, 通信状態決定部, 通信制御部の実装

通信状態決定部は, 無線通信環境シミュレータから通信メディアごとの電波状態を取得し, 走行速度などを鑑みながら各通信メディアの通信状況を遅延, 誤り率, 帯域として算出する. 算出は, あらかじめ用意された関数によって行う.

情報提供部は, 車両状態決定部や取得情報決定部の実装が算出した情報をアプリケーションソフトウェアに提供するためのインタフェースである. エミュレータ内に蓄積されている情報を MIB の形に整形し, SNMP 要求に従って応答を返す.

情報提供部が提供するインタフェースは 2 つ存在する. 片方は車載計算機上で動作するソフトウェアに情報を提供するためのインタフェース, もう片方は, 車外で動作するアプリケーションに情報を提供するためのインタフェースである. 前者は普通の SNMP による通信として扱われるが, 後者は通信状態決定部で算出された誤り率や遅延, 帯域で制限されたインタフェースとして扱われる. 後者の通信制限を行う部分が通信制御部となる. 本来, 通信制御部はアプリケーション独自の通信にも対応する必要があるが, 今回は実装方法の制限 (アプリケーションとしての実装) により, この機能は割愛する.

5. 実装

本章では, 前章の詳細設計を受けて行った実装について簡単に説明する.

まず, 実装にあたり, 地図の準備が必要となる. 今回の実装では, 実際の道路を用いて試験環境を構築できるように, 地図情報として (財) 地図センター発行の「数値地図 2500」を利用できる構造とした. 「数値地図 2500」は, 2500 分の 1 の紙地図に記載されている情報と同等な情報が入っている数値地図である. ここから, 道路のトポジモデータとして格納されている情報を抜き出し, 交通流シミュレータおよび自動車エミュレータで利用した.

インターネット自動車用アプリケーション試験環境を構築するにあたり, 今回はプラットフォームとして FreeBSD 4.x を選択した. また, 環境シミュレータの実装では各シミュレータモジュールを, 自動車エミュレータでは個々の自動車のエミュレータを 1 つのプロセスとして実装した. これにより, プロセスを追加することにより, 新たな周囲情報を追加したり, 自動車の数を増加させたりすることが可能となる. さらに, 各モジュール間のインタフェースは TCP/IPv4, v6

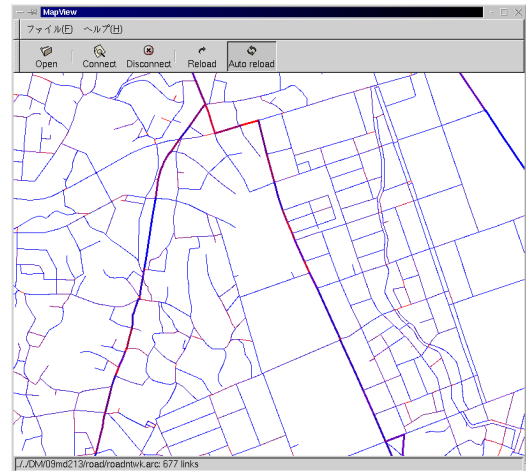


図 6 MapView

Fig. 6 MapView.

で実装した.

今回の実装では, 環境シミュレータの出力はテキストベースの要求応答型インタフェースとしている. このため, ユーザ (アプリケーション開発者) は各シミュレータの内部の情報を, telnet などのアプリケーションを利用することにより, 簡単に取得することができる. しかし, テキストで得られる数値だけでは, 道路地図上の情報を感覚的に理解することは難しい. これを支援するために, 道路の混雑状況などの情報を視覚的に表示するソフトウェア MapView を開発した. MapView のスクリーンダンプを図 6 に示す.

6. 評価と考察

本章では, 本研究で開発したインターネット自動車用アプリケーションソフトウェア試験環境について評価と考察を行う.

6.1 設計・実装によるモデルの評価と考察

今回開発した試験環境は, 環境シミュレータと自動車エミュレータから構成されている. この構造は, それぞれのアルゴリズムの変更や機能追加を容易にしている. これは, 将来, 現時点では予想していないようなアプリケーションが考案された場合にも素早い対応が可能であることを示している.

モデル構築時にも述べたように, 自動車に装備されているセンサは増減する. また, 車種や型式によっても搭載されているセンサは異なる. このことに対応するため, 本実装では環境シミュレータの内部をさらにモジュール化し, 自動車周囲の環境を個々にシミュレーションするような構造とした. 今回の構築でも, それぞれのモジュールをバラバラに開発することが可

能であった。この構造化により、既存のシステムに変更を加えることなく、新たな周囲情報の生成を可能とした。しかし、一方で、周囲情報間の協調が難しいという問題がある。また、モジュールを追加し、その情報を利用するためには、自動車エミュレータ側での対応が必要となる。今回の実装では、交通流シミュレータの作成に約 2 週間、気象シミュレータおよび通信環境シミュレータの構築に約 2 日人を必要とした。

今回の実装では、通信制御部は自動車の情報を提供するインターネット自動車 API に対してのみ機能する。しかし、実際にはインターネット自動車用アプリケーションは独自の通信を行うことが予想される。将来的にはアプリケーションが独自に行う通信に対しても通信制御が可能な機構を準備する必要がある。

また、今回作成した試験環境は、試験を行う際に試験に使う地域や地図情報、再計算を行う時間間隔など、多くのパラメータ設定を必要とする。これらのパラメータを一貫性を持って設定する作業は、アプリケーションソフトウェア開発者の大きな負担となる可能性がある。これは、システムモデルを細分化した結果である。この問題を解決するために、設定を簡単に行うためのツールを開発する必要がある。

6.2 規模性の評価

今回構築した試験環境の目標の 1 つは、大規模な環境での実験が可能なシステムとすることであった。今回構築したシステムがこの目標を満たしていることを確認するために、一番負荷の大きな交通流シミュレータの負荷を測定した。測定は、CPU が PentiumII 400 MHz、メインメモリが 128 Mbyte の PC で、OS に FreeBSD4.1.1 を用いて行った。また、シミュレーションの計算間隔は 3 秒とし、実時間シミュレーションを行った。

負荷測定は、数値地図 2500 の 1 枚の図葉である 2 km × 1.5 km の区域を 1 エリアとして起動し、シミュレーションエリアのエリア数と計算機の負荷の関係を測定した。今回の実装では、交通流シミュレータは、1 エリアに対して 1 つのプロセスとしてシミュレーションを行っている。このため、エリアの数が計算機上で動作するプロセスの数となる。負荷測定の手順は、以下のとおりである。初めに、3 秒おきに 1 エリアずつ 10 エリア分のシミュレータを起動する。その後、120 秒の遅延を置いて 1 分間のロードアベレージを測定する。これを 250 エリアまで繰り返す。250 エリア分の測定を 10 回行い、その平均値を今回の測定結果とした。測定結果を図 7 に示す。図では 10 回の測定における負荷の平均をエラーバーおよびベジェ

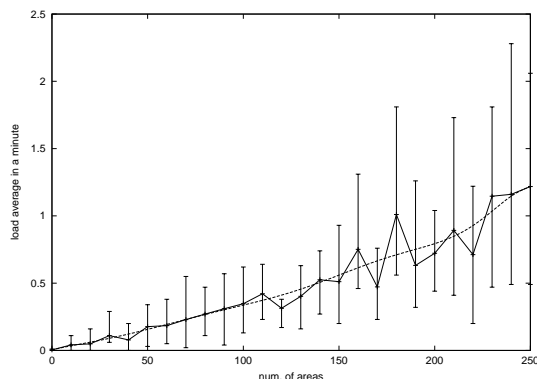


図 7 シミュレーション範囲 vs. ロードアベレージ
Fig. 7 Simulation area vs. load average.

曲線による近似とともに示している。この結果では、かなりな揺らぎはあるものの、200 単位強の範囲のシミュレーションでロードアベレージが 1 に達することが読み取れ、1 台の計算機でも 600 km² 程度の範囲をシミュレーション可能なが分かった。このときにシミュレーションエリア内に存在する自動車の数は 20 万台程度である。また、エリアの増加と負荷の増加が比例することがグラフより確認できる。

また、上記の測定では、各プロセスは約 2 Mbyte のメモリを消費した。より広範囲のシミュレーションを行うためには、十分なメモリ空間を用意する必要がある。

今回の設計では、各モジュール間のインタフェースに TCP/IP を用いているため、性能低下が予想される場合には、複数の計算機を用いることも可能であり、上記の結果は十分規模性を実現していることを示しているといえる。

さらに、TCP/IP を用いた場合のオーバヘッドについて考察する。今回の実装では、各モジュール間の通信をテキストベースのプロトコルで実現している。ここで利用されるコマンドは、大きい場合でも一度の情報やりとりで要求が 20 バイト程度、応答が 5 バイト程度である。600 km² 程度の範囲で、1000 台程度の自動車エミュレータを動作させる場合、シミュレータモジュールが 200 プロセス程度であるのに対して、自動車エミュレータは 1000 プロセス起動することになる。現在の実装ではシミュレータモジュール相互間の通信は、隣り合うセルだけで行われるため、ほぼ無視することができる。また、自動車エミュレータは 1 単位時間ごとに各種のシミュレータモジュールに問合せを 1 度ずつ行うため、交通流、気象、通信状態の 3 種のシミュレータモジュールを利用する場合には、1 単位時間あたり 3000 回の通信、75 Kbyte 程度の通信

量が必要となる。この結果は、現在の計算機では CPU コストの方が先に限界となり、通信のオーバーヘッドは無視できることを示している。ただし、この結果は 1 台の計算機を利用した場合であり、複数の計算機を用いる場合はネットワーク構成などに注意してシステム設計を行う必要がある。

6.3 試験環境に関する考察

本試験環境は、2.1 節で述べた開発手順を簡素化するために構築した。実際には 2.1 節の (3) から (6) の手順を机上で行うことが可能な環境の構築を目標とした。今回の実装では、完全に自動車の環境をエミュレーションできなかったため、机上での試験のみで開発を終了することができる環境を提供することはできなかったが、試験環境として十分有用な環境を実現できたことを確認した。

実際には、今回開発した試験環境を利用することにより、実車両を利用したデバッグの回数を 3 分の 1 程度に軽減することができた。また、机上における試験は簡単にできるため、試験の回数を増やすことが可能となり、アプリケーションの信頼性向上にも貢献することができた。

7. 結 論

本研究では、インターネット自動車用アプリケーションが置かれている環境のモデルを作成し、これに基づいた試験環境を構築した。これまでのインターネット自動車プロジェクトの研究により、自動車をインターネットに接続する環境が整い、アプリケーションの開発が強く望まれる。しかし、実際にはアプリケーションの開発コスト、特にソフトウェアの試験にかかるコストが高く、一般の人々がアプリケーションを開発することが難しかった。本研究により開発したインターネット自動車用アプリケーションの試験環境により、このコストを大幅に削減することが可能になった。

拡張性と規模性に重点をおいてインターネット自動車の特徴をとらえた試験環境を開発した。試験環境に必要な機能を注意深く解析することにより、機能を細分化できることが分かった。その結果、複数の独立して動作するモジュールとして試験環境を構築することが可能になり、拡張性を確保することに成功した。また、複数の計算機にまたがって 1 つの試験環境を構築することを可能とするために、それぞれのモジュール間インタフェースを TCP/IP を用いて定義した。これにより、規模性に富んだシステムを構築することができた。

今後は、より高性能・高機能化を図るとともに、よ

りユーザにとって使いやすいシステムにしていく必要がある。

謝辞 本研究を行うにあたり、研究の場を提供していただきました慶應義塾大学 SFC 研究所インターネット自動車コンソーシアムのメンバーの方々に感謝いたします。

論文の執筆にあたり、ご指導をいただきましたソニーコンピュータサイエンス研究所の寺岡文男博士に感謝いたします。

本研究を進めるにあたり、貴重な助言をいただきました WIDE Project⁷⁾ インターネット自動車分科会の皆様および rover 研究グループ⁸⁾ の皆様に感謝いたします。

研究を進めるにあたり、議論や実験などに協力していただいた慶應義塾大学環境情報学部徳田・村井・中村・楠本研究室の諸氏に感謝いたします。

参 考 文 献

- 1) Uehara, K., Watanaba, Y., Sunahara, H., Nakamura, O. and Murai, J.: InternetCAR—Internet Connected Automobiles, *INET'98* (1998).
- 2) 植原啓介, 湧川隆次, 佐藤雅明, 渡辺恭人, 砂原秀樹, 寺岡文男, 村井 純: 自動車情報化のためのインターネットを用いた通信システムの構築, *情報処理学会論文誌*, Vol.42, No.2, pp.286–296 (2001).
- 3) Aoki, K., Ueda, T. and Fujii, H.: Research and development and the proof test of the probe car, *ITS World Congress 2000* (2000).
- 4) 運輸省自動車交通局 (編): 数字でみる自動車, 社団法人日本自動車会議所 (1999).
- 5) Rose, M.T.: *The Simple Book — An Introduction to Internet Management*, 2nd edition, Prentice Hall (1994).
- 6) 社団法人交通工学研究会 (編): やさしい交通シミュレーション, 交通工学研究会 (2000).
- 7) WIDE Project. <http://www.wide.ad.jp/>.
- 8) rover 研究グループ. <http://www.sfc.wide.ad.jp/rover/>.

(平成 12 年 12 月 18 日受付)

(平成 13 年 5 月 10 日採録)



植原 啓介 (正会員)

1970年生。1995年電気通信大学大学院電気通信学研究科情報工学専攻(博士前期課程)修了。慶應義塾大学環境情報研究所研究員,慶應義塾大学環境情報学部助手,慶應義塾大学大学院政策・メディア研究科(後期博士課程)等を経て,現職,慶應義塾大学政策・メディア研究科特別研究専任講師。移动通信プロトコル,移動計算機環境等の研究に従事。修士(工学)。ACM,IEEE各会員。



砂原 秀樹 (正会員)

奈良先端科学技術大学院大学情報科学センター教授。1960年生。1983年慶應義塾大学工学部電気工学科卒業。1988年同大学大学院博士課程修了。1988年~1994年電気通信大学情報工学科助手。1994年より奈良先端科学技術大学院大学情報科学センター助教授を経て2001年より現職。工学博士。インターネット,モバイル/ユビキタスコンピューティング,大規模広域分散環境,並列処理,オペレーティングシステム,電子図書館に関する研究に従事。電子情報通信学会,ソフトウェア科学会,Internet Society,ACM,IEEE各会員。



村井 純 (正会員)

現職:慶應義塾大学環境情報学部教授。1955年生まれ。1984年慶應義塾大学工学部数理工学博士課程修了。1987年博士号取得。1984年東京工業大学総合情報処理センター助手,1987年東京大学大型計算機センター助手。1990年慶應義塾大学環境情報学部助教授を経て1997年より現職。1999年慶應義塾大学SFC研究所所長。1984年JUNETを設立。1988年WIDEプロジェクトを設立し,今日までその代表として指導にあたる。社団法人日本ネットワークインフォメーションセンター理事長。ICANNボード。著書に「インターネット」,「インターネットII」(岩波新書),監訳に「インターネットシステムハンドブック」(インプレス),「IPv6:次世代インターネットプロトコル」(プレントイスホール)等がある。