

ITS 通信システム評価統合シミュレータの設計と開発

疋田 敏朗^{†1} 高橋 幹^{†1} 吉岡 顕^{†1}

ITS(Intelligent Transport System)として、路車間通信、車々間通信を用いた様々なアプリケーションが検討されている。これらのアプリケーションに共通する課題として、多数の構成要素、技術分野が関連するため、単体での検証・評価が困難なことが上げられる。従来同様実車による検証を考えた場合、シナリオを限定した状態でも、数十～数百台以上といった大量の実験車を用いた大規模な実証実験が必要となる。実証実験は非常に高コストであり、開発の都度このような手法をとることは非現実的なため、シミュレーションによる評価が必要になると考えられる。

我々は「シミュレーション要素」をプラットフォーム上で連携させ、対象となるITS通信アプリケーションの系全体をシミュレートする「統合シミュレータ」の開発を行ってきた。統合シミュレータの開発には、時間・空間概念が異なる各種シミュレーションシステムの連携動作や大量の計算時間を必要とするシミュレーション要素の並列化や高速化が必要であった。

本発表ではアーキテクチャ設計・高速部分に着目して、統合シミュレータの開発・設計について報告するとともに、第一世代の開発・運用経験を元にして、現在開発中である第二世代統合シミュレーションシステムへの展望を発表する

Design and Development of Integrated Simulator Platform for Evaluation of Vehicular Communication Applications

TOSHIRO HIKITA,^{†1} TSUYOSHI TAKAHASHI^{†1}
and AKIRA YOSHIOKA^{†1}

Recently new vehicular safety systems, such as vehicle-vehicle/vehicle-infrastructure cooperative systems have been proposed for reducing traffic accidents. Effectiveness of the proposed systems can be tested by means of traditional field tests, however, such field tests prove to be extremely costly despite having low scalability and low repeatability.. If accurate and efficient vehicular communication simulation tools can be developed, then such tools would fill in the gap where field tests remain inadequate. Development and evaluation of complex vehicular communication applications would benefit tremendously from efficient simulations.

However, evaluation of such vehicular safety applications would require the use and coordination of multiple simulators, each having its own intrinsic timing and relationship attributes.

In this paper we describe the structure and operation of the Integrated Simulation Platform (ISP) which we have developed for the purposes mentioned above.

1. 背景

Intelligent Transport Systems (ITS) 分野で

は、道路側に設置した機器と走行車両間の通信を用いて事故低減を目指す路車協調型アプリケーション⁴⁾、走行車両間の通信による情報交換により事故低減を目指す車々間通信型アプリケーション⁵⁾が、多数検討されている。

これらのアプリケーションに共通する課題とし

^{†1} 株式会社トヨタ IT 開発センター 研究開発部
Toyota InfoTechnology Center, Co., Ltd.
{hikita,tsu-takahashi,yoshioka}@jp.toyota-itc.com

て、多数の構成要素、技術分野が関連するため、単体での検証・評価が困難なことが上げられる。従来同様実車による検証を考えた場合、シナリオを限定した状態でも、数十～数百台以上といった大量の実験車を用いたテストコース実験が現実の交通環境での大規模な実証実験（社会実験）が必要となる。開発の最終段階ではこのような手法での検証も必要であるがアプリケーションを実現するための通信機の開発、アプリケーション内部のパラメータ調整といった開発初期の段階から、その都度このような手法をとることは非現実的であり、シミュレーションによる評価が必須である。

通信を利用する ITS アプリケーションの有効性、成立条件を検証するためには、各時刻における車両位置を決定するマイクロ交通流シミュレーション、通信による情報が伝達可能かを評価するネットワークシミュレーション、送信側機器と受信側機器の通信可能性を評価する電波伝搬シミュレーションと、信号情報提供といった対象となる ITS アプリケーションのシミュレーションを組み合わせる必要がある。

上記の各シミュレーションは、それぞれ単体としては既に多くの研究開発が行われており、研究レベルのもの、市販のものを含めて多数のシミュレータが存在する。これらのシミュレータを用いれば、アプリケーションを構成する要素技術の一つ一つにの評価は可能である。しかし、それぞれ単体のシミュレーションではシステムあるいはアプリケーション全体を通しての効果を検証することは不可能であり、上記の「シミュレーション要素」が連携する形で対象となる系全体をシミュレーションする統合シミュレーションが必要である。

このような観点から、筆者らは通信を利用した ITS アプリケーションの評価を可能とする、マイクロ交通流、ネットワーク、電波伝搬、ITS アプリケーションの各要素を連携させた統合シミュレータを開発してきた^{3),2)}。

本発表ではアーキテクチャ設計・高速部分に着目して、統合シミュレータの開発・設計について報告するとともに、第一世代の開発・運用経験を元にして、現在開発中である第二世代統合シミュレーションシステムへの展望を発表する

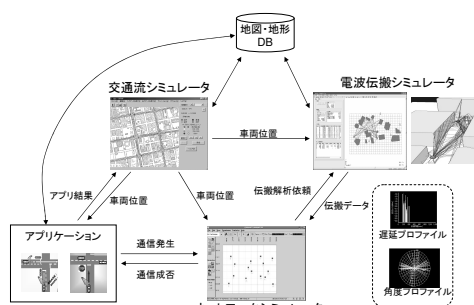


図 1 統合シミュレータにおけるデータの流れ

2. 各シミュレーション要素の連携

2.1 処理の流れと各シミュレーション要素間で受け渡すデータ

前節で述べたように、通信を利用した ITS アプリケーションの有効性、成立条件検証のためには、マイクロ交通流、ネットワーク、電波伝搬に加えて評価すべき ITS アプリケーションという4つの要素シミュレーションが連携する必要があります。図1に、各要素シミュレーションの間で交換すべきデータの流れを示す。

ここで、

- まず、マイクロ交通流シミュレーションによって各車両位置を求める。
- この各車両と路側機の位置関係、あるいは車両間の位置関係から、それらの間に発生すべき通信を ITS アプリケーションシミュレータが生成する。
- その通信発生イベントをネットワークシミュレータに入力し、さらに路側機や各車両位置を通信ノードの位置を入力することにより、通信の成否を求める。
- その際に必要な、送信ノードから受信ノードへの電波伝搬品質を電波伝搬シミュレータで求める。この際には、路側機、車両位置に加えて、回りの建物等の構造物および車両形状を入力し、レイトレースに利用する。
- ネットワークシミュレータでは、得られた伝搬品質からビットエラー率、パケットエラー率を算出し、これと乱数によりパケット到達すなわち通信の成否を決定する。
- ITS アプリケーションシミュレータでは、通信成否の情報を用いて、信号情報提供等のアプリの動作を決定する。

という形で全体のシミュレーションを行う。

2.2 たがい違い法による異なるシミュレーション要素の連携

本開発で対象とする3つのシミュレーション要素は、

- **マイクロ交通シミュレーション**
ある時刻の状態をもとに、きざみ幅分後の状態を求める time step 方式
- **ネットワークシミュレーション**
送信、受信といったイベントがいつおきるかを、時刻を連続的に管理しつつシミュレーションをすすめるイベント駆動方式
- **電波伝搬シミュレーション**
時間の概念はなく、ある瞬間的な時刻における伝搬状態を扱う

という、異なる時間の取扱いをする。そのため、これらを連携させるためには、前節で述べたプロセスに加えて時間管理方式の差異を考慮したプロセスを加える必要があり、図2のような手順となる。

- (1) 交通流シミュレーションにおいて時刻 $t(n)$ の車両位置を計算
- (2) これを ITS アプリ、ネットワーク、電波伝搬の各シミュレータに通知
- (3) ITS アプリシミュレータでは、時刻 $t(n-1)$ から $t(n)$ の間の時間経過、車両の移動等から通信発生イベントを抽出。このイベントリストをネットワークシミュレータに通知
- (4) ネットワークシミュレータでは、 $t(n-1)$ から $t(n)$ の間のシミュレーションを実施。以下の5から8までを通信イベント分だけ繰り返し
- (5) ある通信発生イベントについて、ISO 参照7層モデルにおける上位のアプリケーション層 (L7) から物理層 (L1) へ向かってシミュレーションを実施する
- (6) 物理層 (L1) シミュレーションにおいて必要となる電波伝搬パラメータを電波伝搬シミュレータにリクエスト
- (7) 電波伝搬シミュレータでは、リクエストされた条件についてシミュレーションを実施、結果を応答
- (8) ネットワークシミュレーションでは、得られた電波伝搬パラメータかパケットエラー率を算出し、物理層 (L1) からアプリケーション層 (L7) へシミュレーションを行い、

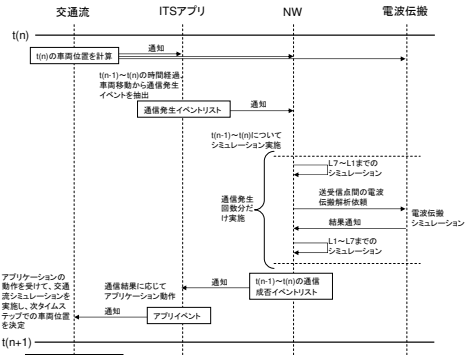


図2 たがい違い法による処理フロー

通信成否を算出

- (9) 全ての通信イベントについての成否を、ITS アプリシミュレータに応答
- (10) ITS アプリシミュレータでは、通信結果に応じてアプリケーションの動作を決定
- (11) 交通流シミュレータでは、ITS アプリシミュレータによって得られたアプリ動作から、次のタイムステップ $t(n+1)$ の車両位置を計算

なお、ここで、時刻 $t(n-1)$ から時刻 $t(n)$ 間の状態から計算されたイベントによってアプリケーションの動作状態を決定し、それを時刻 $t(n+1)$ の車両位置に反映させていることに注意が必要である。ここでは、必ず交通流シミュレータと ITS アプリシミュレータとの間には1タイムステップ分の時間遅れが発生していることになる。この差を埋めるためには、一般に反復計算をする必要があり、膨大な計算量となる。そこで、ここではたがい違い法 (staggered method)¹⁾ あるいは弱連成法と呼ばれる手法を採用した。これは、流体-構造問題等の連続体の連成問題のシミュレーションにおいてよく使われている手法であり、時間刻み幅がその錬成現象が本来持つ時間遅れよりも短い場合には十分に良好な近似結果を得られることがわかっている

3. 基本設計と実装

3.1 密結合と疎結合

複数の要素からなる錬成現象をシミュレーションする場合、

- それぞれの要素シミュレーションにおけるモデル化手法について、ひとつのソフトウェア

として密に組み合わせる方法 (密結合)

- それぞれ単体で動作する要素シミュレータを疎に組み合わせる方法 (疎結合)

の 2 種類の方法がある。

例えば、本開発でも利用しているネットワークシミュレータの Qualnet や ns-2 では、それ自体が一種のプログラミング環境になっており、この中で、電波伝搬モデルや通信ノードの移動モデルをプログラムすることが可能である。この機能を利用すれば、Qualnet あるいは ns-2 単独で、今回のターゲットである交通流—ネットワーク—電波伝搬を統合したシミュレーションを実現可能である。

一方、後者の考え方に基づいた場合には、マイクロ交通流、ネットワーク、電波伝搬の各シミュレータを用意し、これらの中で、必要な情報を通信等を用いてやり取りすることによって統合する形となる。

一般論として、密結合方式では通信等のオーバーヘッドがなく高速なシミュレーションが可能であるというメリットがあり、一方疎結合方式では、オーバーヘッドにより余分なシミュレーション時間を要するというデメリットがある。

一方、密結合方式では全体をひとつのプログラムとして作成するため、開発者が関連する要素シミュレーションの全てについての専門的な知識をもつ必要があるというデメリットを持つのにに対し、疎結合方式では、各要素シミュレーション部について単独で開発可能であり、分担しての開発が可能であること。全部あるいは一部に既開発のシミュレータをそのまま再利用可能であること

シミュレーションの規模、粒度、目的等に応じて、各要素シミュレーション部に求められる精度等の要求要件が変わってくる場合があるが、疎結合方式の場合、ある要素シミュレーション部に対し多種のモジュールを用意し、目的に応じて変更して利用することが可能であること

将来、それぞれの分野における研究開発の進展により新技術・新シミュレーション手法が開発された場合に、要素シミュレーション部を容易に新しいものに交換可能というメリットがある。

本開発では、上記のような過去、他者の成果、知見の再利用性、将来の拡張性を重要視し、疎結合方式を採用することとした。

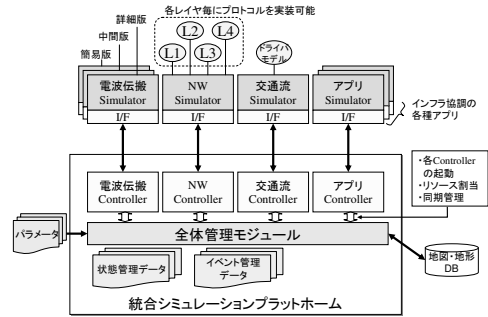


図 3 統合シミュレータのアーキテクチャ

表 1 使用したハードウェアとソフトウェア

	Platform	Propagation	Map
CPU	Opteron 254 (2.8GHz)		Xeon E5345 (2.33GHz) x2
Memory	8GB		
OS	Fedora Core 5	Fedora Core 6	CentOS 5
Softwares	ISP Platform Qualnet Automesh	PostGIS	UPPS

3.2 プラットフォーム構造の採用による拡張性の確保

前節でも述べた各要素シミュレーション部における拡張性をより高い次元で確保するため、本開発では、各要素シミュレーション間で直接データの交換をする構造ではなく、図 3 に示すようなプラットフォーム構造を採用した。

ここでは、全体の管理、各要素シミュレーション部との入出力、同期といった部分をプラットフォームと呼ぶモジュールが担当し、各要素シミュレーションモジュールは、コントローラと呼ぶインターフェースモジュールを介して、プラットフォームとのみ通信する。各要素シミュレーション部を交換する際には、このコントローラ部のみを変更するだけでよく、高拡張性を実現できるようにしている。

4. システムとシミュレーション

4.1 システム実装

システムは主な機能を 3 台のサーバに分散して運用するように設計を行った。これは疎結合構成を取ったため、各要素シミュレータがそれなりに計算力を要求し、単体マシン上での運用では支障を来すためである。

特に計算力が必要な電波伝搬シミュレーション

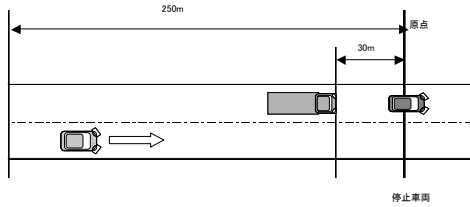


図4 比較実験のシナリオ

については、Quad コアを2つ用いたマシンを用意し、伝搬計算の高速化を図った。詳細については⁶⁾を参照されたい。

プラットフォームの実装に関しては Java を使用し、各要素シミュレータのコントローラとプラットフォームの通信に関しては CORBA を使用してメッセージをやりとりする設計とした。

シミュレータコントローラと I/F 間の通信はシミュレータ毎に別の実装とし、電波伝搬シミュレータについてはソケット通信による通信モジュールを、ネットワークシミュレータについては JNI インターフェイスをそれぞれ用いる形とした。アプリケーションシミュレータは通信部も含めて内包するように実装を行った。

それぞれのサーバの主要 SPEC と統合シミュレータ上におけるコンポーネントは表1の通りである。

4.2 実測値との比較

作成したシミュレーションシステムと実際のシミュレーションとの差異を確認するために、屋外で実際の試験車両を用いて測定した結果とほぼ同等の条件をシミュレータ上に構築したシミュレーション結果の比較を行った。

比較のシナリオは図4のように水平直線路上で車両と車両間で通信を行い(車車間通信)、送信車両の後方30mにおいて大型車両(トラック)を遮蔽車両(障害物)として配置するものとした。また比較のために実際の実験とシミュレーションのパラメータはできる限り同一となるようにしている。

図5のグラフが比較結果である。グラフを見てわかるように、遮蔽車両の後方では実験結果とシミュレーションはほぼ良好な一致をみている。一方で遮蔽車両の近傍においてはシミュレーション結果は実測値よりも通信品質を悪く見積もっていることがわかる。ただし、受信品質の傾向は一致しており、比較結果としてはまずまずで

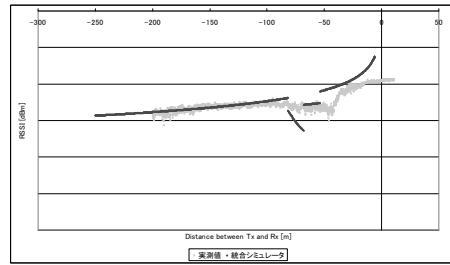


図5 実測値とシミュレーション結果

ある。また、送信点の近傍においては、アンテナの垂直方向の指向性による影響(無指向性アンテナを利用した場合でも鉛直方向には指向性を持つ)をシミュレータが評価できないため、シミュレーション結果と実測値の間に乖離がみられる。

上記の比較結果をまとめると近傍以外の点においては、シミュレーション結果は現実の結果とよい一致をみていると考えることができる。

5. 次世代統合シミュレーションシステム

ここまで述べてきた第一世代の統合シミュレータを母体に、我々は現在よりユーザにとって使いやすい第二世代の統合シミュレータを開発中である

本章では現在の統合シミュレータの課題を述べるとともに、開発中の第二世代システムについて説明する

5.1 統合シミュレーションの課題

我々は疎結合構成を特徴とした ITS 通信シミュレータである統合シミュレータを様々なシミュレーションで使用してきた。しかしながら、開発を続けるにつれ、当初想定もしていなかったような点で様々な課題や制約を抱えることとなった。

主な問題点を列挙すると以下ようになる

- I/F とコントローラの権限が不明確
- 要素シミュレータ間での依存関係
- I/F バージョンの整合性問題

5.1.1 I/F とコントローラの権限が不明確

統合シミュレータを当初設計した際には、それぞれの目的に合わせて要素シミュレータを適材適所で自由に組み合わせることを想定していた。つまり、電波伝搬やネットワークシミュレーションという各要素をシミュレーションシナリオに合わせて、自由に組み替えることが目的であった。

しかしながら、シミュレータ間でのデータの受け渡しには依存関係が存在するためこの組み替

えが自由に行えない場合あることが開発/運用の過程で判明した。

たとえば電波伝搬シミュレーションがどのようなデータを元に伝搬計算を行うのかはそれぞれの電波伝搬シミュレータによって実装が大きく異なっている。ある伝搬シミュレータは詳細な地形データや建物データを要求するのに対して、別の伝搬シミュレータは地形データは要求しないものの詳細なアンテナデータを要求するなどの違いが存在する

統合シミュレータではシミュレーションに必要なデータはすべてプラットフォームから供給することとしているので、各伝搬シミュレータ毎に必要なデータをコントローラと I/F 経由で供給する必要がある。

このため、電波伝搬シミュレータを差し替えた場合、該当シミュレータの I/F の変更のみならず、電波伝搬コントローラ部分も含めて変更せねばならないこととなった。

本来 I/F のみの変更で差し替えを考えていたのだが、すべてをプラットフォームから供給するという前提条件では、コントローラと I/F の切り分けが不明確となってしまう、I/F 部分のみを作成して別のシミュレータと簡便に差し替えるということが大変難しくなってしまう、コントローラ部 (すなわちプラットフォーム本体) ごと作り直すという作業が必要になった。

5.1.2 要素シミュレータ間での依存関係

特定の要素シミュレータ間に依存する処理があったよりも多かったのも、問題となった。たとえば電波伝搬シミュレータはネットワークシミュレータから処理を呼ばれる構造となっているが、この際にどのようなデータを電波伝搬シミュレータとやりとりするのか? はネットワークシミュレータと電波伝搬シミュレータの関係に強く依存する。

そのため、各要素シミュレータ間の通信自体を詳細に定義する必要が出てきてしまい、シミュレータ間の接続にある程度の制限ができたため、選択可能な要素シミュレータの組み合わせが限定されることとなった。

たとえば、A というネットワークシミュレータは、B という伝搬シミュレータには対応するが、C という伝搬シミュレータとはデータ交換できないということが発生した。

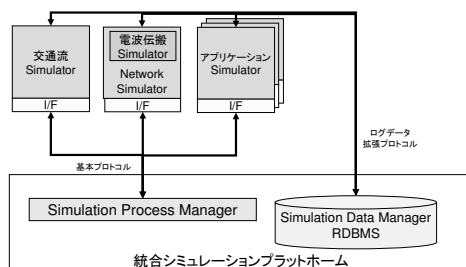


図 6 第二世代統合シミュレータのアーキテクチャ

5.1.3 I/F バージョンの整合性問題

要素シミュレータ間の通信仕様をアプリケーションシナリオごとに定義したため、の変更に伴う、改良箇所が思いもよらぬ数になると、今度は I/F のバージョン整合性ついでの問題が発生してきた。

つまり、シナリオ A で使用したコンポーネントをシナリオ B で使い回そうと考えても、プロトコルが違うため、簡単には使い回せず、プロトコルのバージョンを合わせるという作業が必要となり、なかなか要素を再利用しづらくなった。

これらの問題点をまとめると、まず要素シミュレータ単位をシミュレータ製品単位で分割したため、不要な通信がプラットフォーム経由で発生したこと。そして要素シミュレータの I/F について、本来は単純に保っておくべき通信プロトコルに複雑なアプリケーション用の通信を無理矢理押し込めたことが問題となっているとまとめられる。

5.2 設計方針

これらの反省を踏まえて、第二世代のシミュレータは必要な機能単位ごとに切り出せること、中長期的に利用できるプラットフォームとすることを目的として設計することとした。

その実現のために、以下の目標をもって設計を行うこととした

- 機能ブロックごとの切り出し
- プロトコルの分離
- I/F 部分の厳格な定義
- 設計の自由度の確保

5.2.1 機能ブロックごとの切り出し

まず各要素シミュレータについては、従来の製品または分類ごとに切り分けるのではなく、機能ブロックごとに切り出すことにした。一例を挙げると従来は電波伝搬シミュレータと通信シミュ

レータは別の要素シミュレータであったが、電波伝搬シミュレータが Layer-0、通信シミュレータが Layer-1-7 までを担当していることを鑑みて、第二世代統合シミュレータでは通信シミュレータとしてひとくくりにし、通信シミュレータが内部で電波伝搬シミュレータを呼ぶように変更した。

この変更により、電波伝搬と通信という特に依存関係が強い部分はプラットフォームを経由させず、単要素として隠蔽することが可能になった。

5.2.2 プロトコルの分離

第一世代統合シミュレータにおける通信プロトコルの混乱は、全シミュレータが必要とする通信 (たとえばノードの位置情報や通信の成否) と特定要素シミュレータ間で行われる通信 (たとえば電波伝搬に関する正確なプロファイル) の両方を一つの通信手順内に組み込んだことで発生したと考えた。

そのため第二世代統合シミュレータの設計に当たっては、全要素シミュレータがシミュレーションを行う上で必ず必要とする限定された情報のみを基本プロトコルに乗せることとし、それ以外の要素については拡張プロトコルとして分離することとした。

基本プロトコルは各要素シミュレータが連動する最低限必要な通信の手順をサポートするのみとし、基本プロトコルのみでは最低限度のシミュレーションのみが実施可能と定義した。このことは最低限度の接続ならびにシミュレーションを保証するとともにプラットフォーム部である Simulation Process Manager について、その構造を簡略化することも兼ねている。

一方で複雑な通信に関しては拡張プロトコルを利用することとし、拡張プロトコルは逆に従来よりも遙かに自由度を高める様に設計を行っている。

5.2.3 I/F 部分の厳格な定義

プロトコルの分離と同時に、I/F とコントローラ役割分担についても再考した。プラットフォームに組み込まれていたコントローラ部の大半については基本プロトコルの厳格定義を行うことで不要とし、残りの各要素シミュレータに密接に依存している部分に関しては従来の I/F とコントローラを合わせて I/F 部とした。

Simulation Process Manager と各要素シミュレータの I/F 部分のプロトコルは厳格に仕様を

定義し、基本プロトコルレベルであれば自由に差し替え変更ができるように設計を行った。

5.2.4 設計の自由度の確保

基本プロトコルを限定する一方で、各要素シミュレータ間で連携を行う部分については従来よりも自由度を確保し、自由にシナリオ作成ができるように拡張プロトコルの設計を行った。

拡張プロトコルは、シナリオごとに書き直すことを前提としており、ユーザが自由に実装を行うことが可能である。

一方で、データの受け渡しについては明示的にプロトコルで定義することを避け、各要素シミュレータが Simulation Data Manager を経由して受け渡すように設計することで、実装の負担を軽減している。

これは、拡張プロトコルの通信を通解する Simulation Data Manager 自体には高度なデータ処理機能を持たせず、あくまでデータの蓄積と提供をリクエストベースで行う様に設計することで Simulation Data Manager への改変の必要性を下げている。

5.2.5 DB の利用

拡張プロトコルを制御する Simulation Data Manager については、高速性とスケーラビリティの確保を行うため、独自のプロトコルを定義することを避け、データベースを利用して、書き込み・検索を行う設計とした。

具体的には RDBMS を利用し、それぞれの要素シミュレータが DB 内に table を作成、それを他の要素シミュレータから検索を行うことで、各要素シミュレータが必要なデータのやりとりできるようにしている。

この方法の利点は、データを書き込む側は何をデータを利用するのかについて明確な定義が不要であり、データについては基本的に利用する要素シミュレータが選択を行うことで接続部の設計を楽にしている。

また RDBMS への接続方法 (Binding) には多様な手法が提供されていることから、接続プロトコルを独自に定義したり、通信部を実装する必要が少ないこともメリットである。

Simulation Data Manager は全記録が記載されることから、それ自体が log 記録装置としても動作する。

この手法の欠点としては、シミュレータの計算結果をすべて記録するため、膨大なデータ量が

発生してしまうことがあり，対応策として何らかの専用ストレージシステムを導入することを考えている。

6. まとめと今後の課題

本発表では，ITS 通信アプリケーションのための統合シミュレータについて，アーキテクチャ設計・高速部分に着目して，統合シミュレータの開発・設計について報告を行った。また，第一世代の開発・運用経験を元にして，現在開発中である第二世代統合シミュレーションシステムの基礎設計について，その概要について説明した。

第二世代統合シミュレータは現在構築中であり，今後は完成した第二世代システムを用いて，今後導入が予想されるインフラ協調アプリケーションの性能評価などを実施していく予定である。

参 考 文 献

- 1) C. Farhat, M. Lesoinne, and P. Le Tallec. Load and motion transfer algorithms for fluid/structure interaction problems with non-matching discrete interfaces: Momentum and energy conservation, optimal discretization and application to aeroelasticity. *Computer methods in Applied Mechanics and Engineering*, Vol. 157, No. 1-2, pp. 95–114, April 1998.
- 2) Toshiro HIKITA, Toshinori KASAI, and Akira YOSHIOKA. Integrated simulator platform for evaluation of vehicular communication applications. In *Proceedings of the 2008 IEEE International Conference*, pp. 323–327. IEEE, 2008.
- 3) 吉岡顕, 小佐井潤, 本多輝彦. Its 通信アプリケーション評価用統合シミュレータの開発. *DICOMO 2007 論文集*, pp. 1762–1766. 情報処理学会, 2007.
- 4) 国土技術政策総合研究所. *Ahs 早期実現化を目指す 7つのサービス*.
- 5) 先進安全自動車推進検討会国土交通省自動車交通局. 第 3 期 asv 推進計画成果報告書, 2006.
- 6) 疋田敏朗, 笠井俊典, 吉岡顕. 統合シミュレータによる現実都市環境を模擬した車車間通信シミュレーション. *DICOMO 2008 論文集*, pp. 1600–1605. 情報処理学会, 2008.