

大都市大震災軽減化特別プロジェクトにおける 震災総合シミュレーションシステムアーキテクチャ

The Architecture of the Integrated Earthquake Disaster Simulation System
under Special Project for Earthquake Disaster Mitigation in Urban Areas

小藤 哲彦*1
Tetsuhiko Koto

野田 五十樹*2
Itsuki Noda

桑田 喜隆*3
Yoshitaka Kuwata

竹内 郁雄*1
Ikuo Takeuchi

*1電気通信大学
The University of
Electro-Communications

*2産業技術総合研究所
National Institute of
Advanced Industrial Science and Technology

*3NTT データ
NTT DATA CORPORATION

Special Project for Earthquake Disaster Mitigation in Urban Areas is the project aimed at decreasing the damages caused by big earthquakes. We are developing the kernel of the Integrated Earthquake Disaster Simulation System (IDSS) under the project. The system has the following features: (1) It integrates various kinds of *sub-simulators*, such as an earthquake simulator, a fire simulator, a traffic simulator, a collapse simulator, and so on. (2) It is desired to work on an economic PC cluster that consists of a number of PCs so that a disaster over large area can be simulated without being slowed down. (3) It supports rollbacks in order to enable to retry simulations with various conditions.

1. はじめに

大都市大震災軽減化特別プロジェクトとは、大地震が発生した際の人的・物的被害を大幅に軽減することを目的としたプロジェクトである。筆者らはこのプロジェクトにおいて、震災総合シミュレーションシステム (IDSS) のカーネル開発に携わっている。このカーネルは、次のような特徴を持つ (1) 地震シミュレータや火災シミュレータなどをプラグインし、それらの結合として統合シミュレーションを実現する。(2) 災害空間を地理的に分割し、それらを異なる計算機上で並行してシミュレーションすることで、空間的なスケーラビリティを持つ。(3) ロールバックをサポートする。

以下では、シミュレーションされる世界の時間・時刻を、単に時間・時刻と呼ぶ。

2. アーキテクチャ

シミュレーションシステムのアーキテクチャを、図 1 に示す。広大な災害空間を短時間でシミュレーションするために、シミュレーションを複数の計算機で行う必要がある。IDSS では、災害空間の地理的な分割を、分散の主要な単位として扱う。それぞれの分割に対して、1つのカーネルと、各種類ごとに1つずつのサブシミュレータが割り当てられる。サブシミュレータは、割り当てられた地域のシミュレーション状況をカーネルから受け取り、割り当てられた地域のシミュレーション結果をカーネルへ返す。他の地域とのインタラクションを解決するために、サブシミュレータは、割り当てられた地域に加えて、隣接する地域の一定範囲のシミュレーション状況をも、カーネルから受け取る。この一定範囲のことを、のりしろと呼ぶ。

カーネルが管理するシミュレーション経過を、共有時空間データ (SSTD) と呼ぶ。シミュレーション世界は、属性を持ったオブジェクトの集合としてモデル化され、共有時空間データはオブジェクトや属性の時系列を表す。カーネルは、サブシミュレータに対して共有時空間データを通知し、また、サブ

シミュレータから返されるシミュレーション結果を共有時空間データへ反映させる。また、カーネルは、各シミュレータのシミュレーションの進行を管理する。あるシミュレータのシミュレーションに、他のシミュレータのシミュレーション結果が必要になる場合、必要なシミュレーションが終了するまで、シミュレータを停止させる。

3. S-API

カーネルがサブシミュレータへ提供する API を、S-API と呼ぶ。S-API は、すべてのサブシミュレータに共通のインターフェイスと、サブシミュレータごとに異なるインターフェイスとで構成される。サブシミュレータの開発者は、そのサブシミュレータが、シミュレーション世界の中のどのようなオブジェクトと属性にアクセスするかを、スペース定義ファイルと呼ばれるファイルへ記述する。スペース定義ファイルは、スペースジェネレータと呼ぶプログラムによって、そのサブシミュレータ固有の S-API へ変換される。この API は、シミュレーション世界のオブジェクトを、Java のオブジェクトによって表現する。これによって、サブシミュレータは、シミュレーション世界中のオブジェクトの属性に対して、それがあたかも Java のオブジェクトの属性であるかのようにアクセスでき、Java の静的な型チェックなどの恩恵を受けることができる。分散シミュレーションのためにシミュレーション空間が分割されている場合、S-API は、サブシミュレータが担当する地域のオブジェクトと、のりしろ部分に存在するオブジェクトとを、サブシミュレータへ提供する。

サブシミュレータは、以下の (1) ~ (4) の処理を繰り返すことで、シミュレーションに参加する。サブシミュレータは、現在時刻と呼ばれる値を持っており、これはそのサブシミュレータがシミュレーションを完了して時刻を表す。現在時刻の初期値は、シミュレーションを開始する時刻である。また、サブシミュレータは、固着時刻と呼ばれる値を持っている。固着時刻は、現在時刻より未来の時刻であり、これより前の時刻はシミュレーションが終了しており結果をすべてカーネルに通知し終わっていることを表す。

連絡先: 〒 182-8585 東京都調布市調布ヶ丘 1-5-1 電気通信大学 情報工学科 小藤 哲彦 Tel. 0424-43-5321 Fax 0424-43-8074 koto@takopen.cs.uec.ac.jp

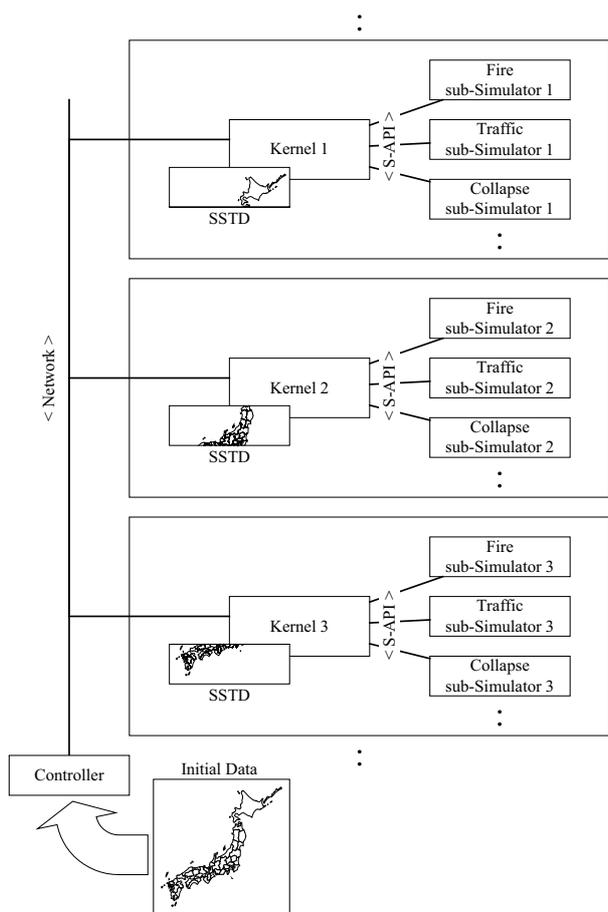


図 1: システムの概要

(1) シミュレーション経過の読み込み

現在時刻におけるシミュレーション世界の状態を読み込む。スペースジェネレータによって出力される S-API によって、シミュレーション世界に存在するオブジェクトに対応する Java のオブジェクトを取得できる。オブジェクトには、そのオブジェクトが持つ属性ごとに、その値を取得するメソッドが用意されている。このメソッドは引数を取らず、サブシミュレータの現在時刻における属性の値を返す。

(2) シミュレーション結果の通知

スペースジェネレータによって出力されたクラスは、サブシミュレータがシミュレーション結果をカーネルへ伝えるためのメソッドも提供する。オブジェクトが持つ属性ごとに、時刻と値を引数にとるメソッドが用意される。時刻はサブシミュレータの固着時刻と同じかそれより未来の値でなければならない、値はその時刻において、その属性が取るべき値である。

(3) 現在時刻と固着時刻の更新

サブシミュレータは、sync メソッドと呼ばれるメソッドを用いて、現在時刻と固着時刻を更新する。sync メソッドには、次の 3 種類がある。いずれのメソッドも、他のサブシミュレータのシミュレーションが十分進んで、新しい現在時刻におけるシミュレーション世界の状態が決定されるまで、呼び出しを終了しない。

`syncAt(time, lookahead)`

現在時刻を *time* に設定し、固着時刻を *time* に *lookahead*

の時間を加えた時刻に設定する。*time* は呼び出し時における現在時刻より大きくなければならない。

`syncByEvent(propertyList, lookahead)`

propertyList で与える属性が変化した時刻を新しい現在時刻とし、それに *lookahead* を加えた時刻を新しい固着時刻に設定する。

`syncByEventUpto(propertyList, time, lookahead)`

propertyList で与える属性が変化した時刻を新しい現在時刻とする。ただし、時刻 *time* までに属性が変化しなければ、*time* を新しい現在時刻とする。いずれも、新しい現在時刻に *lookahead* を加えた時刻を新しい固着時刻に設定する。

4. ロールバック

震災シミュレーションシステムには、ロールバックの機能が要求される。例えば、実際の災害においてを利用する場合、災害状況について新しい情報が得られた場合、より高い精度の結果を、より短時間に求めるには、その情報が得られた時刻までシミュレーションをロールバックすることが必要である。また、どのように救援物資や人員を導入すると高い効果が得られるかを試行錯誤的に調べたい場合、設定を変更したい時点までロールバックし、そこから新しい設定のシミュレーションを開始する機能が必要である。

IDSS では、ロールバックを次のようにサポートする。時刻 *t* にロールバックが行われた場合、サブシミュレータは、*t* をまたぐように現在時刻を更新した sync 関数の呼び出しが終了した時点の状態が復元される。以前のシミュレーションにおいて、サブシミュレータが、現在時刻 $t_0 (< t)$ において syncAt メソッドを呼び出し、その結果現在時刻が $t_1 (\geq t)$ になっていた場合、現在時刻 t_1 の状態が復元される。現在時刻 $t_0 (< t)$ において syncByEvent あるいは syncByEventUpto メソッドを呼び出し、その結果現在時刻が $t_1 (\geq t)$ になっていた場合、時刻 *t* 以降で属性が変化した時刻を現在時刻とする状態が復元される。

5. 関連研究

5.1 HLA

High Level Architecture (HLA) は、サブシミュレータ(フェデレートと呼ばれる)を統合するシミュレーションアーキテクチャである [1]。HLA では、のりしろのようなアプローチは行われていない。

6. まとめと今後の展開

大都市大震災軽減化特別プロジェクトにおける震災総合シミュレーションシステム (IDSS) について説明した。今後は、システムを実装し、パフォーマンスの評価を行う。

参考文献

- [1] IEEE Std 1516-2000 IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA) - Framework and Rules.