

大規模災害における プロトタイプレスキューエージェントの設計と実装

Design and implementation of the prototype rescue agents for large scale disasters

*1 渡邊 日出光 *1 伊藤 暢浩 *1 本地 健一 *3 太田 正幸 *2 *4 森下 卓哉
Hidemitsu Watanabe Nobuhiro Ito Ken'ichi Honji Ohta Masayuki, Takuya Morishita
*2 *4 下羅 弘樹 *4 小藤 哲彦 *3 野田 五十樹 *4 *5 竹内 郁雄 *1 犬塚 信博
Hiroki Shimora Tetsuhiko koto Itsuki Noda Ikuo Takeuchi Nobuhiro Inuzuka
*1 和田 幸一
Koichi Wada

*1 名古屋工業大学 *2 福井大学 *3 産業技術総合研究所
Nagoya Institute of Technology University of Fukui National Institute of Advanced Industrial Science and Technology
*4 防災科学技術研究所 *5 東京大学
National Research Institute for Earth Science and Disaster Prevention The University of Tokyo

Recently, rescue projects for large-scale natural disasters that occur worldwide are brought to attention. One such a project is the special project for earthquake disaster mitigation in urban areas of Japan ministry of education, culture, sports, science and technology. We call the project Dai-Dai-Toku. In the project, a real disaster simulator, which we call it IDSS: integrated disaster simulation system, is being studied. In this paper, we presented an architecture for IDSS to process a large number of agents in an appropriate time length. And we designed and implemented prototype rescue agents based on the architecture. Further we also confirmed the efficiency of our architecture through several simulations.

1. はじめに

近年、各地で大規模な自然災害に対する取り組みに注目が集まっている。その取り組みの一つとして、文部科学省が進めている大都市大震災軽減化特別プロジェクトがある。ここでは、震災総合シミュレーションシステム (IDSS)[1] の設計と実現を行っている。本研究では、多くのエージェントを高速にシミュレーションできるしくみを提案し、そのしくみに基づいたプロトタイプエージェントの設計および実装をおこなう。また、いくつかのシミュレーションを通して、そのしくみの有効性を検討する。

2. 震災総合シミュレーションシステム

震災総合シミュレーションシステム (IDSS) とは、以下の 4 つから構成される。

時空間情報システム DyLUPAS 災害空間に存在する全てのものとその時間変化を表し、IDSS の基礎データベースの役割をはたす。

災害分析・推定シミュレータ群 震源地や地震の波形、各地の震度などの情報に基づき、建物の倒壊や道路閉塞、火災の発生、埋没者の分布など、シミュレーションの初期値となるデータを与える。

災害予測・対応シミュレータ群 災害分析・推定シミュレータ群と実世界からの情報から、実時間より早い被災地域のシミュレーションをおこなう。

連絡先: 渡邊 日出光, 名古屋工業大学, 愛知県名古屋市昭和区御器所町, TEL:052-735-5408, FAX:052-735-5408, hide@phaser.elcom.nitech.ac.jp

実世界インターフェース群 実世界からの観測データを DyLUPAS へ入力したり, DyLUPAS から実世界へ情報を提示したりする。

IDSS はこれらを統合し、実時間より早く精度の高いシミュレーションをおこなう。これにより、IDSS は緊急時には、救助活動の意思決定支援や情報の収集・提示などができ、防災に役立てることができる。また、平時には、震災のシミュレーション結果を見せることによる市民の啓蒙や、災害規模の予測をおこなうことができる。

我々は次に詳細を述べる災害予測・対応シミュレータ群の開発に携わっている。

2.1 災害予測・対応シミュレータ群

災害予測・対応シミュレータ群は、IDSS の中で意思決定支援を目的とし、被災地域のシミュレーションをおこなう役割を持つものである。そのためには、以下のような様々な要求が存在する。

- 精度の高いシミュレーションをおこなうため、被災地域を忠実に再現しなければならない
- 意思決定の支援に使うため、シミュレーションが実時間より早くなければならない
- 様々な地域で、様々な自然災害を様々な規模でおこなうことができるようにするため、システム規模の可変性、汎用性、柔軟性が保証されていなければならない

我々は災害予測・対応シミュレータ群の開発プロジェクトチームにおいて、これらの要求に対し、地図、災害・物理現象、人々といった被災地域に存在するものをモデル化した。また、災害予測・対応シミュレータ群は図 1 で示すように、様々な仕組みが導入されている。

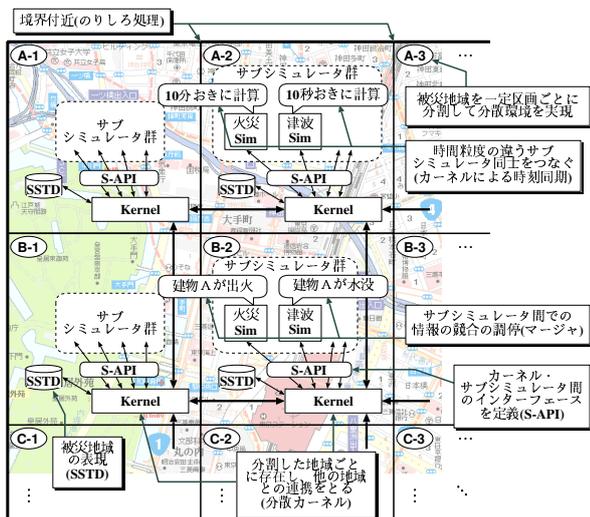


図 1: 災害予測・対応シミュレータ群の構成

様々な仕組みとして以下のものが挙げられる。

分散カーネル 各カーネルが分割された地域をそれぞれ統括・管理する。各カーネルが相互に通信することで、分割された地域をつなぐ。

SSTD 地図や人という被災地域に存在するオブジェクト群とその状態の情報を表現する。

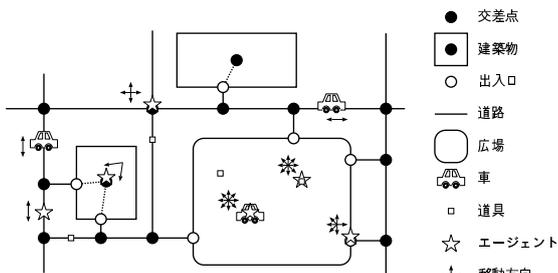
S-API 災害や人々を計算するシミュレータ (以降、サブシミュレータと呼ぶ) とカーネル間をつなぐインターフェースである。

マージャ 複数のサブシミュレータから矛盾、あるいは数値の異なる属性変更要求が来たときにそれらを調停し、もっとも正しいと考えられるイベントを反映する。

のりしろ処理 分割された地域の境界付近の整合性を保つ。

3. エージェントのモデル化

本研究では、エージェントのプロトタイプの実現のために、まず被災地域を図 2 に示すような様々なオブジェクトによって表現する。次に、この被災地域に存在する人一人をエージェント 1 体としてモデル化する。モデル化されたエージェントは、次に述べる三つのステップを繰り返しておこない、被災地域で活動する現実の人 (ロボット) の動作を表現する。



被災地域を表す地図の構造

図 2:

1. 周りのオブジェクトの存在とその状態を知覚する (Sense)
 エージェントは、自分から一定範囲内のオブジェクト (建物やエージェントなど) の状態 (建物の燃え具合やエージェントの埋まり具合など) を知ることができる。自分以外のエージェントに対しては、何を考えているかまでは知ることはできない。
2. 知覚して得られた情報を用いて行動のプランを立てる (Think)
 エージェントは、Sense によって得られた情報に加えて、自分の状態や過去のオブジェクト情報、行動履歴などの情報から行動を決定する。
3. 立てた行動プランを基に実際に行動にする (Act)
 エージェントは、Think で決定した行動を実行に移し、実際に各オブジェクトに対して作用する。これにより各災害・物理現象サブシミュレータは、エージェントの作用を受けて各オブジェクトの状態を変化させる。

4. プロトタイプエージェントの実現

プロトタイプエージェントとなるエージェントサブシミュレータに対しては、以下の四つの要求が発生する。

- 要求 1 被災地域を忠実に再現するために、人一人をエージェント 1 体としてシミュレートできなければならない。
- 要求 2 分割された地域の規模から、担当する一台のコンピュータで 1000 体程度のエージェントをシミュレートできなくてはならない。
- 要求 3 シミュレータは緊急時の意思決定支援にも使われるため、実時間よりも早いシミュレーションができなければならない。
- 要求 4 今後のエージェント開発者にとって変更が容易で利用しやすいエージェントサブシミュレータでなければならない。
- 要求 5 エージェントは分割された地域 (カーネル間) を自由に移動するため、エージェントのプロセスや内部状態などのマイグレート処理が必要となる。

次節から、それぞれの要求を満たすためのプロトタイプエージェントの設計と実装について考察する。

4.1 APX の導入

4. 章の要求 1 から、エージェントはそれぞれ一いつつカーネルと通信するのが望ましい。しかし、要求 2 からエージェントの数が多くなる。よって、個別にカーネルと通信しては通信量が増加してしまう。そのため、要求 3 の高速な実行を満たすことが出来なくなる。通信量の増加を防ぐために、エージェントサブシミュレータを図 3 のように設計した。

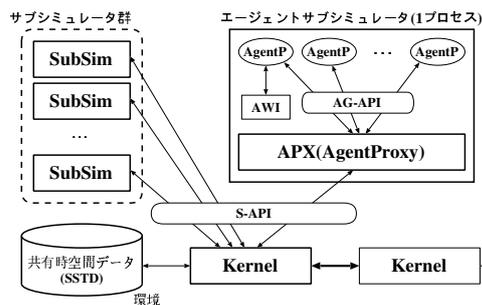


図 3: 提案するエージェントサブシミュレータ設計方法

個々のエージェントは AgentP (AgentProgram) と呼ばれるエージェントの意思決定プログラムで一つ一つ表し、エージェント全体を APX (AgentProXy) と呼ばれる統括シミュレーター一つで表される。そして、これらは全てまとめて一つのプロセス上で立ち上げられる。APX AgentP 間は通信ではなくローカルな関数呼び出しによってお互いのやりとりを実現する。

AgentP の導入によってエージェントはそれぞれ個別の意志決定ができる (要求 1)。APX によって、カーネルとの通信をまとめられるため、通信量を軽減させることができる。そのため、より多くのエージェントを接続することができる (要求 2)。ただし、シミュレータの実行速度は AgentP の実装方法に依存する。

4.2 AG-API の導入

ここでは、4. 章の要求 4 を受けて、エージェントの災害戦略の変更の容易性を実現するために導入した AG-API について述べる。4.1 節で説明した APX と AgentP その間のインターフェースとして AG-API (AGent-API) [2] を導入する。AG-API はエージェントの行動をコマンドとして APX にわたす。これにより、今後これらを利用するエージェント設計者は AG-API の仕様に従って AgentP のみを設計すれば良いことになる。そのため、エージェント設計者が S-API を使用してサーバの変更をおこなう必要が無いので、その負担が軽減される。更に、エージェントの「ある場所に移動したい」、「ある区域の消火活動をしたい」等の意図を解釈して一連の経路決定や消火プロセスを出力するような、AWI (AgentWillInterpreter) と呼ばれる基本行動セットのパッケージを作ることによって、エージェント設計者の負担を更に軽減させる仕組みも検討している。

4.3 エージェントのマイグレーション

4. 章の要求 5 は分割された地域をエージェントが自由に移動できること。のエージェントの分割された地域の移動はオブジェクトをバイト列に変換する JAVA のシリアライゼーションを使い通信により実現する。ただし、エージェントの記憶量の増加は、マイグレーション時の通信量の増加につながるため、エージェントの適切な記憶モデルについては検討が必要である。また、要求 4 の利便性を維持するために、エージェントのマイグレーション処理は AgentP ではおこなわないようにする。そうすることにより、後のエージェント開発者には地域分散を意識せず、エージェントプログラムを改変できるようになる。

4.4 プロトタイプエージェントの実装方式

我々はプロトタイプエージェントを実装するにあたり、多数のエージェントを高速に動作させるために (要求 2, 要求 3)、エージェントを逐次実行方式で実装した。ここでは、その実装方法について検討する。

実装方法として以下の 3 つの方式について検討する。

マルチプロセス方式：エージェント一体一体をプロセスとして実行し、全エージェントが並列に実行される方式。

マルチスレッド方式：エージェント一体一体をスレッドによって実現し、全エージェントが並列に実行される方式。

逐次実行方式：エージェントを一体づつ起動し、一体が行動を終了するたびに次のエージェントが起動される方式。

マルチプロセス方式について、エージェントそれぞれが JavaVM を起動することとなるので、エージェント 1000 体以上を目指す今回の場合はメモリの使用量から非現実的である。

表 1: スレッド方式と逐次実行方式の実行速度

| スレッド数 | 逐次方式 実計算時間 | スレッド方式 実計算時間 |
|-------|---------------|-----------------|
| 10 | 38m 秒 | 418m 秒 |
| 20 | 70m 秒 | 804m 秒 |
| 50 | 171m 秒 | 1797m 秒 |
| 100 | 582m 秒 | 3585m 秒 |

マルチスレッド方式と逐次実行方式について、重要な実行速度の点で比較する。

pentium3 1.0GHz 上のコンピュータ上でソートプログラムを使い、スレッド方式と逐次実行方式を比較した。表 1 にあるように簡単なソートプログラムにおいてさえ逐次実行方式の方がスレッド方式と比べ実行速度の点で大きく優位性があることがわかった。

今回求められているのは 1000 体以上のエージェントを実時間より早く実行することである。そのため、複雑なエージェントプログラムを実現するにあたり逐次実行方式での実装は合理的であるといえる。逐次実行方式はどこか一箇所問題が発生すれば全体として止まってしまう問題点があるが、一定時間応答がなければスキップするような仕組みを導入することにより、この問題点を解決できる。今回は、要求されている実行時間から逐次実行方式を採用したが、もし、使用するコンピュータの処理能力が高速で、かつ、メモリも膨大な場合は、ロバスト性の高い、マルチスレッド方式、マルチプロセス方式を採用していくことも考えられる。

5. シミュレーションとその考察

本研究で設計・実装をおこなったエージェントサブシミュレータについて、4 章の要求 1, 要求 2, 要求 3 が満たされているかシミュレータを用いた実験により確認する。また、要求 4 はエージェント設計者の利用しやすさなので、測定することが難しい。加えて、要求 5 のマイグレーションは未実装なため、この二つについてはここでは考えない。

我々は図 4 の地図 (大きさとおブジェクト数は表 2 に示す) 上でシミュレータを用いて実験をおこなった。

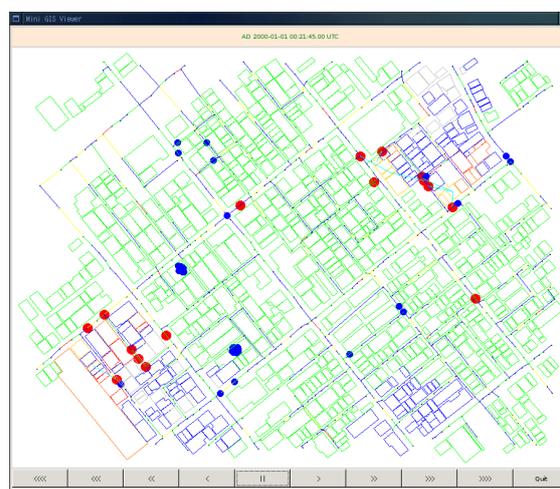


図 4: Viewer によって視覚化されたシミュレーションの様子

シミュレータの実行によってエージェントが個別に意思決定(要求1)していることが確認できた。また、エージェントの数に対するシミュレータの動作時間を測定することによって、多数のエージェントを高速に動作させるという要求(要求2, 要求3)が満たされているかどうか確認した。実験として、接続するエージェントの数を6パターンに分け、それぞれシミュレーション時間5分のシミュレーションをおこなったときの実計算時間、最大メモリ使用量を測定した(表3)。

実験環境として Pentium4 2.8GHz の CPU, 1.5GB のメモリ容量を持つコンピュータを使用した。

表2から、500体のエージェントを接続したときに、シミュレーション時間5分よりも長い実計算時間が必要となることが分かる。しかし、実験で使用した地図は300m × 400mの面積にあるオブジェクト数で比較すると表2にあるように、実測値に基づく地図よりもオブジェクト数の多い複雑な地図である。エージェントは扱うオブジェクトの数が増えると多くの計算時間を必要とする。したがって、地域分散をおこないより単純な実測値に基づく地図を使用した場合には、1000体程度のエージェントがシミュレーション時間未満の実計算時間で動作することが十分に考えられる。

表2:実測値に基づく地図と実験で使用した仮想的な地図

| 地図の種類 | | 実測値に基づく地図 | | 仮想的な地図 |
|---------|-----|----------------|------------------|------------------|
| 地図の大きさ | | 縦 4km 横 3km | 縦 400m 横 300m | 縦 300m 横 400m |
| オブジェクト数 | 建築物 | 34986 | 約 350 | 740 |
| | 交差点 | 4226 | 約 42 | 765 |
| | 道路 | 5704 | 約 57 | 820 |

表3:実験結果

| エージェント数 | 実計算時間 | 最大メモリ使用量 |
|---------|---------|----------|
| 10 | 1分 34秒 | 121MB |
| 70 | 2分 19秒 | 138MB |
| 200 | 3分 15秒 | 146MB |
| 500 | 5分 56秒 | 189MB |
| 800 | 9分 11秒 | 243MB |
| 1000 | 11分 22秒 | 281MB |

6. まとめと今後の課題

本研究では、災害予測・対応シミュレータ群におけるエージェントサブシミュレータの設計と実装をおこなった。

その動作は、平成16年11月15日におこなわれたIDSSの開発チーム全体による内部中間評価において確認された。

また、4.章のはじめに述べた使用目的からくる要求について、APX, AgentP, AG-APIを導入することによって解決した。これらによって、より現実に即したエージェント一体一体の意思決定を実現し、5.章で示したように、地域分散の処理をおこなえば、実測値に基づく地図上においても十分高速に動作する可能性を示した。

今後の課題としては、エージェントのマイグレーション処理、AWIの実現等、未実装部分の早急な実装と、エージェントの通信モデルの詳細決定がある。加えて、エージェントの戦略の最適化やエージェント間の協調による減災効果の検証などをおこなえるようにする必要がある。

参考文献

- [1] 文部科学省研究開発局, 独立行政法人防災科学技術研究所「大都市大震災軽減化特別プロジェクト 震災総合シミュレーションシステムの開発 平成15年度 成果報告書」2004.5
- [2] 名古屋工業大学 本地 健一「大規模災害におけるプロトタイプレスキューエージェントの設計と実装」名古屋工業大学修士論文 2005