JAIST Repository
https://dspace.jaist.ac.jp/

Title	実証実験システムにおける機器や物理現象のエージェント化に関する研究
Author(s)	押川，侑樹
Citation	
Issue Date	2018-03
Type	Thesis or Dissertation
Text version	author
URL	http://hdl.handle.net/10119/15172
Rights	
Description	Supervisor:篠田　陽一，先端科学技術研究科，修士（情報科学）

JAIST
JAPAN
ADVANCED INSTITUTE OF
SCIENCE AND TECHNOLOGY

Japan Advanced Institute of Science and Technology

実証実験システムにおける
機器や物理現象のエージェント化に関する研究

北陸先端科学技術大学院大学
先端科学技術研究科

押川 侑樹

平成 30 年 3 月

修 士 論 文

実証実験システムにおける
機器や物理現象のエージェント化に関する研究

1610036　　押川 侑樹

主指導教員　篠田陽一 教授
審査委員主査　篠田陽一 教授
審査委員　丹康雄 教授
知念賢一 特任准教授
Razvan Beuran 特任准教授

北陸先端科学技術大学院大学
先端科学技術研究科 [情報科学]

平成 30 年 2 月

Copyright © 2018 by Yuki Oshikawa

概 要

センサやネットワーク、データ解析をはじめとする技術の発展に伴い、社会のあらゆる場所に情報機器が展開され、様々な情報が交換・収集・解析されている。解析されたデータを可視化し価値のあるサービスとして提供することで、これらの情報機器は社会の生産性向上に寄与している。上記のような情報システムの社会への円滑な導入を進めるためには、導入後に得られる効果の予測が必要不可欠であり、事前に実際の利用場面を想定した動作実験による検証を行うことが重要となる。

情報システムを検証する方法として、実際に一部の地域社会に導入し実証実験を行う方法と仮想空間において実社会を模倣し検証する方法が存在する。一部の地域社会で検証する方法では、実運用と同じ現実環境であり、有用性は現実の結果として現れるため効果がを正確に測ることができる。しかし、一部の地域社会とはいえ、実際に導入して実証実験を行うためには多大なコストが必要となる。また、災害時におけるシステムの有用性を示したいような場合では、現実世界において災害時と同じ環境を再現することは難しく、また危険であるという問題がある。そのため現実世界を用いる手法は災害時に機能するシステムの実証実験には向いていない。それに比べ、仮想空間に社会を模倣し検証する方法では、実社会で再現するには危険な現象でも仮想空間では自由に実現でき、コストも抑えられるため災害や大規模環境等の検証に適している。しかし、現実環境同様に複数の要求に応じることのできる柔軟な仮想空間を提供することは難しいため、社会を構成する要素をモデル化し、各モデルを仮想空間上で有機的に繋ぐことで実社会と等価な環境を提供するような仕組みが必要となる。そのための手法として、社会への接近ができるマルチエージェントシミュレーションが有効であると考える。しかしマルチエージェントシミュレーションを支える統合プラットフォームの大半は、社会現象が起きた原因を探るための利用が主であり、情報システムの事前検証を行うための環境としては適していない。また、情報システムの開発や実験、訓練などへの応用を目的とするマルチエージェントシミュレーションも存在するが、再現できる環境や機能が限定的で特定条件下の実験しか行えないものが多い。これらの点から、柔軟な構成を行うことのできるマルチエージェントシミュレーションをより実社会に近付けるため、足りない機能を補うようにシミュレータやエミュレータとマルチエージェントシミュレータを連携させ、適切に制御する必要がある。

本研究では、マルチエージェントシミュレーションのエージェントと各シミュレーション、エミュレーションを結び付け、シミュレーションのシナリオ全体をエージェントを中心に制御を行う、エージェント化を提案する。エージェント化をするにあたり、対象の特性に合わせエージェントモデルを「個体モデル」と「空間モデル」という２種類に分類した。個体モデルには、人や携帯電話、自律的に動作するソフトウェアなど、シナリオ中で1つの個体として認識できる存在が分類される。この個体が持ち得る情報が全て、エージェントに集約され管理されることで、エージェントを中心とした連携を実現する。これに対し、空間モデルには、火災や地震のような、1つの実態として定義できず、広範囲に影響を与える存在や現象が分類される。このような個体として扱うことのできない存在や現象がもつ物理状態、物理量を空間モデルエージェントに集約し管理を行うことにより、個体モデルと同様に空間モデルにおいてもエージェントを中心とした連携を実現する。これら２種類のエージェントモデルにより、エージェントを中心としたシミュレータやエミュレータの統合的な制御を実現する。

提案手法に基づき実験を行うため、Jonathan、Inferno の二種を実装し、各モデルを実証する。Jonathan では、歩行者エージェントと通話アプリケーションの連携を行い、個体モデルのエージェント化を実証した。Inferno では、フィールドエージェントと火災シミュレータの連携を行い、空間モデルのエージェント化を実証した。また、個体モデルである避難者エージェントと空間モデルであるフィールドエージェントの協調によって、間接的に物理シミュレータが避難者エージェントに影響を与えることを示した。これにより、全てのシミュレータやエミュレータを個体モデルもしくは空間モデルのエージェントとしてエージェント化することで、エージェントを主体とする連携ができ、統合的なシミュレーションを実現できることがわかった。

提案手法であるエージェント化は、シミュレータ、エミュレータで提供される機能や情報をエージェントに集約し操作可能にできるため、統合的なシミュレーションを実現する。本研究の提案手法は、社会への利用を想定した事前検証を助け、情報システムの社会への円滑な導入に貢献するものである。

目 次

第 1 章　はじめに **1**

1.1　背景 1

1.2　目的 2

1.3　本論文の構成 2

第 2 章　統合シミュレーション **3**

2.1　用語定義 3

2.2　情報システムの普及と効果予測の重要性 3

2.3　実社会を意識した分野横断的な検証 4

2.4　仮想空間における社会の再現 4

2.4.1　大規模な実験環境の必要性 5

2.4.2　分野横断的な検証のための統合 5

2.4.3　シミュレータ連携のためのコントローラの必要性 6

2.4.4　ボトムアップなシナリオ制御の必要性 7

第 3 章　関連研究 **8**

3.1　マルチエージェントシミュレータ 8

3.1.1　RoboCup 8

3.2　シミュレータの連携基盤 9

3.2.1　RUNE(Real-time Ubiquitous Network Emulation environment) . . . 9

3.2.2　HLA(High Level Architecture) 9

第 4 章　マルチエージェントモデルによるシミュレーション統合 **10**

4.1　マルチエージェントモデルによる仮想社会 10

4.1.1　エージェントとは 10

4.1.2　マルチエージェントモデルとは 11

4.2 エージェントによるグローバルシナリオの生成 11
4.3 エージェント化の実現手法 12
4.3.1 エージェントモデルとシミュレーションのマッピング 12
4.3.2 マルチエージェントモデルにおける各シミュレータ、エミュレータ連携の利点 13
4.4 エージェントモデル適応の検討 14
4.4.1 個体モデル 14
4.4.2 空間モデル 14
4.4.3 個体モデルと空間モデルの連携における問題 15

第 5 章 システムの設計 16
5.1 システム構成 16
5.1.1 Mutil-Agent Simulatior(MAS) 16
5.1.2 External Simulator 19
5.1.3 ICT 基盤 19
5.1.4 協調システム 20
5.2 データ交換方式 20
5.3 ID のマッピング 20
5.4 データ形式 21
5.5 時間同期 21

第 6 章 Jonathan：個体モデルによる歩行者エージェントと通話エミュレータの連携 23
6.1 概要 23
6.2 MAS 23
6.2.1 artisoc による実装 23
6.2.2 実験空間全体の制御 23
6.2.3 歩行者エージェントのシナリオ 25
6.3 通話機器 25
6.4 協調システム 26
6.5 データ構造 26
6.6 実験シナリオ 26

6.7 結果 27

第 7 章 Inferno:空間モデルによる避難者エージェントと火災シミュレータの連携 30

7.1 概要 30
7.2 MAS 30
7.3 火災シミュレータ 30
7.4 協調システム 32
7.5 データ構造 32
7.6 location resolver 32
7.7 実験シナリオ 33
7.8 結果 33

第 8 章 考察 35

8.1 統合的な制御の実現 35
8.2 複数のシミュレータやエミュレータとの協調動作 36
8.2.1 データ構造の分類と適応検討 37

第 9 章 おわりに 40

9.1 まとめ 40
9.2 今後の課題と展望 40
9.2.1 複数のシミュレータやエミュレータとの連携 40
9.2.2 抽象度の高いシナリオ記述 40

第1章　はじめに

本章では、研究の背景、目的、論文の構成について述べる。

1.1　背景

情報通信、金融、航空、鉄道、電気、ガス、公共サービス、医療、水道、物流など、機能が停止または低下した場合に多大な影響を及ぼしかねないサービスは重要インフラと呼ばれ、人々の安全で快適な生活を支える必要不可欠な要素である。このような重要インフラを補助、制御し実現するため、携帯電話、PC、センサ類をはじめとする情報システムが幅広く用いられている。また近年では、少子高齢化による人手不足や技術継承問題、インフラ老朽化などの社会的な問題から、教育、医療、防災、農業などの様々な分野においてIoTやAIといった最先端技術を活用した情報システムの導入が進められている。そのため、私たちの身の回りの環境や社会生活に密接に関連する情報を適切に収集分析し、社会へフィードバックする情報システムの開発は今後ますます重要になってくるものと考えられる。しかし、そういった情報システムを構築し、運用する上では、そのシステムの振る舞いの正しさや、期待される効果の予測が必要不可欠である。また、安心安全で快適な社会を提供し、効率化をもたらすための設計・開発をするためにも、コストや社会への影響を考慮した事前の検証が必要である。

現代の社会は多種多様なシステムが有機的につながり大規模で複雑なため、社会全体の動きや変化の様子を予測することは困難である。また、それらの情報システムによる社会への影響は、利用する人間や環境との有機的につながりによって動的に変化していくと考えられるため、社会全体の動きや変化を予測することは容易ではない。

したがって、有機的な繋がりを再現して。多種多様な分野を横断的に検証することが求められている。

1.2 目的

本論文では上記で述べたような背景の中で、多種多様な分野における横断的な検証のためマルチエージェントモデルによるシミュレーション統合を目的とする。マルチエージェントモデルの特徴として、ボトムアップにシナリオを記述することができる。しかし、エージェントモデルをベースとした多種多様なシミュレータを結び付けるような枠組みは提供されていない。そこで、マルチエージェントモデルをもとに、複数のシミュレーションやエミュレーションを統合的に扱うため、エージェント化を提案する。またエージェント化によって、データの集約、交換のような機能を持つ。今回提案するマルチエージェントシミュレータは、増加し続ける情報システムを検証するための基盤となる。

1.3 本論文の構成

本論文の構成は以下の通りである。1 章では背景として、社会を取り巻く情報システム検証の重要性について述べた。2 章では統合シミュレーションについて述べる。3 章では関連研究について述べる。4 章ではマルチエージェントモデルによるシミュレーション統合について述べる。5 章ではシステム設計について述べる。6 章では Jonathan の実装について述べる。7 章では Inferno の実装について述べる。8 章では考察を行う。9 章では本論文のまとめを行い、今後の課題と展望について述べる。

第2章　統合シミュレーション

本章では、社会におけるシミュレーションを取り巻く環境とその環境における要求の変化について述べ、さらにその変化によって伴う問題について触れる。

2.1　用語定義

シミュレータとは、モデル化対象が決まっており、1つあるいは2つ、3つ程度の限られた現象やシステムの再現、分析を行う計算ソフトウェアのことである。対象とするシステムや現象を数理モデルなどに変換し、計算することで再現したり解析することを目的としている。同じシステムや現象であっても、全ての特徴を捉えモデル化することは複雑であるため、どこを注視するかによってモデルとして切り取られる内容が異なる。また、モデル化した対象を再現、解析するための計算能力の限界もあり、シミュレータ1つで全てを再現することはなく、得意とする領域の異なる様々なシミュレータが開発、利用されている。

エミュレータとは、対象とするシステムの機械的な動作や機能を再現することで、対象システムを模倣するようなソフトウェアのことである。通常のシミュレータのように対象をモデル化し再現するのではなく、実際に動作させて検証することを目的としている。予測できる現象より、予測できない現象が支配的であるときに用いられる。

2.2　情報システムの普及と効果予測の重要性

安心安全で快適な社会環境を実現するため、社会のあらゆる場所に導入された情報システムを活用し、様々なサービスが開発、提供されている。特に、今日の社会生活で利用する情報通信、金融、交通、電気、ガス、医療、水道、物流などの重要インフラは、人々の生活になくてはならないものとなっている。また、新たなサービス例として、少子高齢化による人材不足や技術の継承者不足などの社会的な問題から、IoT や AI といった最新技

術を活用したシステムが社会へと導入され始めている。具体的な例としては、物流の分野でドローンを活用した配送サービスが実験的に行われている。配送時におけるドローンの飛行では、人の侵入検知や気象計測を行うことで、安全な離着陸や荷物の受け渡しを可能にする。このように、ドローンが動作することによって、人へ与える影響や環境から受ける影響などを的確に把握し、制御できる必要がある。また、ドローンの活用は労働力不足の解決以外にも、災害時の物流インフラとしての役割も期待されている。その他の例として、自動運転車の開発、検証が進んでおり、将来的な普及によって、交通渋滞の改善や交通事故の削減などに貢献することが期待されている。このような実社会と IoT や AI を活用するモノとの結び付きは今後ますます強化され、増えていくと考えられる。今後の様々な分野における効率化のためにも、身の回りの環境や社会生活に密接に関連する情報を適切に収集・分析し、社会へフィードバックするような情報システムの開発は、今後ますます重要になると予測される。新規サービスの円滑な導入を行うためにはシステムの効果を事前に考慮する必要があるが、実社会は多種多様な要素が大規模で複雑に絡み合う社会システムであり、新規に導入するシステムの効果や影響を実験することなく正確に予測することは困難であり、実証実験による効果予測が重要となっている。

2.3 実社会を意識した分野横断的な検証

IoT 機器の普及や新規サービスの円滑な導入を行うため、システムの事前検証が必要である。情報システムに関わる人や周囲の環境がシステムに与える影響や、反対にシステムが周囲に与える影響など、実社会を意識した分野横断的な事前検証は今後ますます重要になると予測される。例えば、ロボットやモニタリングシステムは、センサを通して環境を知覚することで、周囲の状況を把握する。そのため正しく検証するためには、そういったセンサを通して交換されるような物理量、物理状態の再現も必要となる。また、効率化を目的として他システムとの連携なども考えられる。このように、システムの開発においては開発者の専門分野以外の影響を考慮した分野横断的な検証が求められる。

2.4 仮想空間における社会の再現

前節で述べたように、IoT や AI の活用シーンの高度・複雑化に伴い実証実験も複雑化、大規模化が進む中で分野横断的な検証による社会の再現が求められている。実社会におけ

る情報システムを事前に検証する手法として、現実世界で実証実験を行う手法と、仮想環境を用いる手法が存在する。現実世界で実証実験を行う手法の利点としては、実環境と同一の複雑性をもつ環境での実験を行うことができるため、効果が正しく検証されやすいといったことが挙げられる。しかしこの手法には火災や地震などの災害を実際に起こすことが難しく、また危険である点や、街一つを用いるような大規模実験が、土地や参加者の確保が困難であるなどの問題点が存在するため、実環境ではなく仮想環境を用いた実証実験が必要不可欠となる。仮想環境を実現する上で考慮するべき要件として、社会を構成する人の動作はもちろんのこと、物理的影響も考慮に入れ検証されるべきである。このような背景の中で、本節では、シミュレータ、エミュレータ等の仮想環境における検証と、そのための統合について扱う。

2.4.1 大規模な実験環境の必要性

小規模な検証を行うためにはシミュレータと実機があれば十分であるが、コスト的な問題からその二点のみでは大規模な実験を行うことが難しい。システムによっては、複数の機器が動作することによって初めて見つかる運用上の課題や不具合などが存在するため、大規模な実験を行うことは必要不可欠である。このような大規模実験が困難という課題を解決し実際と同等のアプリケーションを動かして動作確認したいという欲求を充足するために、汎用的な計算機の上で実機の機械的な振る舞いを模倣するようなエミュレータによる仮想機器の利用が有効である。エミュレータを利用することで、実際に作成したソフトウェアを動作させることができ、計算機に依存するが実機を製作しなくとも大規模な実験を行うことができる。

2.4.2 分野横断的な検証のための統合

2.1 で述べたように、人の動作や物理現象など開発するシステム以外の分野を意識した検証が重要になる。また、検証の方法としてシミュレータやエミュレータの利用が有効であると述べた。しかし、シミュレータは対象とするシステムや現象を数理モデルなどに変換し、計算することで再現したり解析することを目的としている。そのため、同じシステムや現象であってもどこを注視するかによってモデルとして切り取られる内容が異なり、また、モデル化した対象を再現、解析するための計算能力の限界もあり、あるシミュ

レータ単体で全てを再現することはなく、得意とする領域の異なる様々なシミュレータが開発、利用されている。したがって、再現したい現象に合わせ多様なシミュレータを連携させる必要がある。また、エミュレータに関しても、様々な機器間の連携なども考えられるため、多数の機器を柔軟に連携できるようにする必要がある。複雑に連携するシステムや物理現象、人間の振る舞いを考慮したシミュレーションを行うためには、シナリオに合わせて各シミュレータが動作するように設定を行うのではなく、現実に即した概念モデルに合わせてシナリオを記述し、シミュレーション環境を制御できることが望ましい。本シミュレーションは、大規模マルチエージェントシミュレータを基盤とする、多様なシミュレータ、エミュレータが統合されたシミュレーションとして実現される。しかし、その環境を再現するためにもまた、多様なシミュレータやエミュレータを利用し連携させる必要がある。そのため、

- 多種多様なシミュレータの連携
- 多数の検証機器の柔軟な連携

の二点が必要となる．また、主にシミュレータは、負荷の分散や開発の簡単化のため、１つの現象やシステムの再現、解析を目的として開発される。そのため、複雑な現象やシステムの検証を目的とする場合、目的の異なるシミュレータ同士を組み合わせることで実現することが一般的である。しかし、シミュレーションを連携させるためには、目的とする検証内容に合わせた適切なシナリオの作成とそれに基づく制御が必要となる。主に必要となる制御として、以下の２つが挙げられる。

- シミュレーション間の時間同期
- 適切なタイミングでのデータ交換

2.4.3　シミュレータ連携のためのコントローラの必要性

社会の多岐に渡る分野において分野横断的に応用され、相互に関係する情報システムを検証する際、現状ではシミュレータやエミュレータには相互に連動し補完しあうような枠組みは存在しない。実験環境に分散するシミュレータやエミュレータの設定を、実験のたびにアドホックに設定することは困難であり、現実的とは言えないため、全体を集中的に制御するコントローラが必要とされる。

集中制御を行うコントローラがシナリオに合わせたイベントとデータのみを各シミュレータ、エミュレータに公開することで、インタラクションを簡潔にすることが可能となる。また、異なる分野、領域のシミュレータ、エミュレータを統合する場合、それぞれの専門分野の専門家は特定のシミュレータ、エミュレータのみに精通しており、連携先の各シミュレータ、エミュレータの細部を知らずに利用することが多い。そのような中で、開発者、利用者が他のシミュレータ、エミュレータについて知ることなく連携する基盤は必要不可欠である。

2.4.4　ボトムアップなシナリオ制御の必要性

背景より、有機的な繋がりを再現し、多種多様な分野を横断的に検証できることが重要である。分野横断的な検証に対しては、ボトムアップな手法によるシミュレータを利用した仮想社会の再現が有効となる。検証においては社会的、物理的または機能的に影響を観測する必要があるため、多数のシミュレータ、エミュレータ、実機がシナリオの元に連携・統合し、検証できる枠組みが必要である。他のエージェントモデルが再利用可能とし、ボトムアップなアプローチを取ることで、利用者が実験したい内容に合わせてシステム全体のシナリオ設定や各シミュレータの環境設定を行うことなく、エージェントのシナリオ作成に専念し、検証に取り組むことができる。

第3章　関連研究

3.1　マルチエージェントシミュレータ

社会にはいくつもの要素が有機的に結びつきあい、影響し合っている。そのため人間一人一人での振る舞いは理解できても、全体として動作した際に想定できないような現象が生じることがある。マルチエージェントシミュレータでは、このような現象の解析をサポートするために設計、開発されている。主に、人間の思考や行動が反映されるような経済、交通、災害時の避難などの現象の解析や人間の行動を制約を加えることで全体を望んだ形に導くための方法 (効率の良い誘導モデルや避難モデルなど) として用いられている。

3.1.1　RoboCup

「西暦 2050 年までに、人間のサッカーの世界チャンピオンチームに勝てる、自律型の人型ロボットチームを作る」ことを目的とし、毎年開催される国際的ロボット競技大会である。大きく 4 つの分野に分けられ、さらに細かく分類される。その中でも特に、シミュレーションを用いた RoboCup Soccer と、RoboCup Rescu について述べる。

- RoboCup Soccer Simulation
  人と一緒に協力して競技を行える制御プログラムの実現を目的としている。11 体分の仮想選手を制御するためのプログラムを作成し、シミュレーション環境で競技を行う大会である。11 名の仮想の選手と競技を行うためのフィールドが、シミュレーション環境として提供される。大会参加者は、11 体分の選手を制御するためのエージェントモデルのプログラムを作成し競技を行う。[めも] ここに画像の引用をする

- RoboCup Rescue Simulation
  都市直下型の大地震における災害救助活動をシミュレーション環境で行うマルチエージェントシミュレーションである。救助隊、消防隊、土木隊、被災者の 4 つの役割

に分けられたエージェントが存在する。地図や火災の発生頻度などのパラメータを事前に変更することができ、多様な災害状況を再現することができる。

3.2 シミュレータの連携基盤

3.2.1 RUNE(Real-time Ubiquitous Network Emulation environment)

Rune はユビキタスネットワークのシミュレートを目的とし、大規模ネットワークのシミュレーションに対応可能なスケーラビリティの確保や様々なネットワーク機器およびプロトコルをエミュレートできる機構などの要求を満たす実験環境として提案された。エミュレーション対象、シミュレーション対象は全て Space という抽象概念で扱われ、これら Space 間の連携は Conduit と呼ばれる仮想の通信経路を通して行われる。

また、ユビキタスネットワークのシミュレーションでは、数理モデルによる機能的な検証から、開発が進むにつれソフトウェアの検証、実装した試作機の検証へと段階的に検証を行いたいという要求がある。そのため Rune では、マルチレベルエミュレーションレイヤと呼ばれる段階的な実験を支援する枠組みが提供されている。

3.2.2 HLA(High Level Architecture)

米国防省の主導により標準化された分散シミュレーションの規格で、RTI(Run-Time Infrastructure) と呼ばれるミドルウェアを通してデータ交換や時間同期などのサービスを提供する。HLA に準拠した RTI は共通のサービスや API を備えているが、実装方法や通信プロトコルの詳細は特に規定されていない。そのため、プロトコルが異なる場合があり、複数種類の RTI を混在して利用することはできない。

第4章　マルチエージェントモデルによるシミュレーション統合

本章では、あらゆる対象をエージェントとして表現したマルチエージェントモデルによるシナリオ制御を提案し、その実現のためにどのような機能が必要であるかを議論する。

4.1　マルチエージェントモデルによる仮想社会

本研究で提案するシナリオ制御のためのマルチエージェントモデルでは、現実の端末もシミュレーションモジュールもエミュレータも全てマルチエージェントシステムの世界の中にマッピングされたエージェントとして表現する。検証のため現実社会を忠実に再現しようとすればするほど、ある1面のみを切り取るような形でシナリオを表現することは困難となる。特に、実社会におけるシステムや現象の1つ1つをある程度理解していたとしても、実際にはそれらが有機的につながり、社会全体として複雑な振る舞いを見せるため、トップダウンなシナリオの記述では今まで想像もしていなかった結果が得られるかもしれず正確な結果を得ることができない。そのため、創発という形で複雑な現象を解き明かすことを試みるような手法であるマルチエージェントを利用することは、複雑かつ大規模な現実世界を、シナリオとして制御、検証するために有効であると考える。

4.1.1　エージェントとは

人や生き物など、周囲の状況を認識し、一定のルールのもとで自律的に行動する主体のこと [1] をエージェントと呼ぶ。

4.1.2　マルチエージェントモデルとは

個々の人間や生物の個人的な好みや振る舞いから理解できることが、全体の結果 (現象) に反映されないことを複雑系という。このような、複雑系に関する現象を、個々のエージェントが相互に作用して行った結果、積み重なってできたものであると捉えて解析する [1] モデルのことをさす。マルチエージェントに構成される対象世界では、エージェントと呼ばれる要素で構成される。対象世界で起きる物事の全ては、エージェントの振る舞いの結果引き起こされるものである。そのため、エージェントが起こす変化は、たとえエージェント自身への変化が小さいものであっても、対象世界へ少なからず影響を与える。

4.2　エージェントによるグローバルシナリオの生成

2 章の通り、社会に様々なサービスや技術が適応された世界を模写するためには、複雑系に対応したシナリオの生成が重要である。ここでの複雑系とは、複数の構成要素をもち、要素が密接に結び付いて互いに影響し合うようなシステムのことである。複雑系における要素は、自身がもつ規則に基づいて他の要素の影響を受けつつ、状態や行動を変化するため、システム全体の状態、挙動が動的に変化する。そのため、自律的な規則に基づく変化による全体の動的な変化を模倣できる環境が重要である。このような複雑系モデルに対するアプローチとして、社会科学の分野ではマルチエージェントモデルが利用されている。マルチエージェントモデルでは、エージェント単位の行動規則の多様性とその多様性 (ランダム性、あいまい性) を考慮した相互作用の記述を前提とする。これにより、多様なエージェントの局所行動の変化がシステム全体へ影響する創発現象を観測することができる。

この特徴を利用し、エージェントの状態遷移とエージェント間のインタラクションによって、全体のシナリオを生成していくボトムアップなアプローチが行える。

ボトムアップなアプローチを取ることで、実験環境全体のシナリオ設定に置いて各シミュレータの環境設定を行うことなく、用者が実験したい内容に合わせ、(他のエージェントモデルが再利用可能であるという前提であるが) 対象としているエージェントのシナリオ作成に専念できる。

4.3 エージェント化の実現手法

本論文で提案するエージェント化とは、様々なチャネルをもつシミュレータや ICT 機器を MAS(Multi-Agent Simulator) に接続・統合するために必要とされるインターフェースの追加や交換されるデータをエージェントの属性としてマッピングすることである。

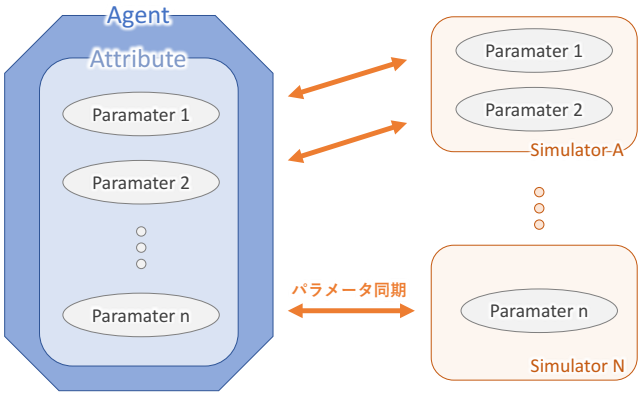

図 4.1: エージェント化の提案

4.3.1 エージェントモデルとシミュレーションのマッピング

以下に具体的な概念図を示す。

- 物理量は空間の属性として扱う
- イベントやシナリオ進行における協調動作はエージェント間のインタラクションとして表現する

各シミュレータでデータの更新があれば、エージェント自身の状態の変化として反映される。この時の状態変化があらかじめ与えられていた条件に合えば、自身の他の状態を変化させたり、他エージェントへ働きかける。これはつまり、エージェントモデルからシミュレータへのシナリオ制御として反映される。このとき、シミュレータ間で交換するデータは、何らかのエージェントモデルの属性としてマッピングされる。

4.3.2 マルチエージェントモデルにおける各シミュレータ、エミュレータ連携の利点

マルチエージェントモデル内のエージェントとしてシナリオの構成要素全てを定義することによって、主に以下の二つの利点が挙げられる。

- ボトムアップなシナリオによる複雑系の記述
- 各シミュレータ、エミュレータに分散するパラメータをマルチエージェントシミュレータ内に集約

一つ目の利点であるボトムアップなシナリオによる複雑系の記述はすでに述べたとおりである。現実世界は複雑系であり、要素同士が絡み合うためある側面だけを切り出したトップダウンなシナリオ記述が困難である。エージェント化を行うことにより、エージェント単位でのボトムアップなシナリオを直感的に記すことができ、それぞれの協調動作により現実世界を模倣できる。

二つ目の利点であるエージェント化によるパラメータの集約を以下に示す。

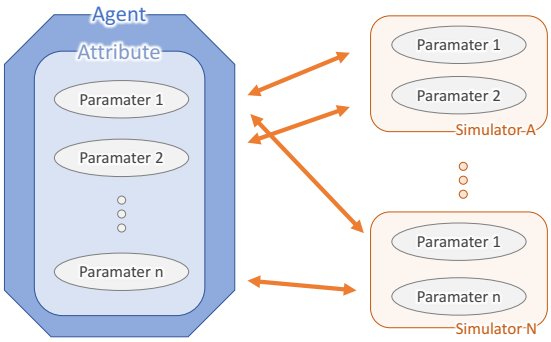

図 4.2: エージェント化によるパラメータの集約

図 4.2 の通り、マルチエージェントシステムを用いなければ各エミュレータ、シミュレータに分散し存在していたシナリオを構成するパラメータを、マルチエージェントシミュレータ内に全て集約することが可能となる。これにより、異なるエミュレータ、シミュレータ間でのパラメータのやりとりや制御を容易にすることができると同時に、新たなシミュレータ等を追加する度に他の全てのシミュレータ、エミュレータとの連携機構を書くことなく、マルチエージェントシミュレータとの連携のみでパラメータの共有を行うことが可能となる。

4.4 エージェントモデル適応の検討

エージェント化するにあたり、その対象によって特徴が異なる。そこで、エージェント化する対象を「個体モデル」と「空間モデル」という 2 種類のモデルに分類した。それぞれの特徴と、それらを連携する際の問題を以下に記述する。

4.4.1 個体モデル

人や車、携帯電話、自律的に動作するソフトウェアなど、1 つの個体として認識できる存在が個体モデルに分類される。個体モデルには、対処とする個体の持ち得る情報が統合される。例えば、携帯電話を例にすると、

- 通話やメールなどのサービス機能に関する状態
- バッテリー残量や発熱などの物理量、物理状態に関する情報

などが挙げられる。そのため、個体モデルではこのような対象に関連する様々な情報をエージェントに集約し操作できるように結びつける。これによって、個体モデルエージェントをマスターとし、対象とする個体の状態に関与するシミュレータ、エミュレータとの協調動作を実現する。

4.4.2 空間モデル

火災や地震のような、広範囲に影響を与え 1 つの実体として定義できない存在や現象が空間モデルに分類される。

空間モデルには、初期値として与えられた範囲の空間に関する物理量や物理状態が統合される。空間モデルとして統合する理由として、

- 連携する複数の物理シミュレータに同じ物理量や物理状態の情報が存在する可能性があるため

が挙げられる。物理シミュレータでは、同じ現象を扱う場合でもシミュレータ内で利用する物理量が異なることや、異なる現象を扱う場合でもシミュレータ内で利用する物理量が同じであることが考えられる。そのため、空間モデルでは同じ空間に関する物理量や

物理状態をエージェントに集約し操作できるように結びつける。これによって、空間モデルエージェントをマスターとし、空間の状態に関与するシミュレータとの協調動作を実現する。

4.4.3 個体モデルと空間モデルの連携における問題

空間に遍在する物理状態や物理量は、空間モデルエージェント (空間エージェント) の属性として同期される。そのため、個体モデルのエージェント (個体エージェント) が特定の状態や物理量を得るためには、空間エージェントへ問い合わせる必要がある。しかし、空間エージェントの領域が初期の段階で固定されるのに対し、個体エージェントは移動することがある。そのため、個体エージェントが現存する位置と位置に対応する空間エージェントの組み合わせが状況に応じて動的に変化する。その結果、人エージェントが問い合わせをする際に、どの空間エージェントに問い合わせれば良いかわからないといった問題が生じる。そこで、位置情報から空間の ID へ変換する機能が必要となる。

第5章　システムの設計

本章では、提案手法であるエージェント化に基づくシステムの設計を述べる。

5.1　システム構成

システムの概要図を図 5.1 に示す。また、図 5.2 の流れに沿って、実験全体やシナリオの制御が行われる。

5.1.1　Mutil-Agent Simulatior(MAS)

本システムにおける MAS は、実験者が定義したエージェントモデルに従い協調動作するための環境を提供する。また実験環境全体の制御コントローラとしての役割も持つ。以下、MAS 内部の構成要素について述べる。

- Experiment Environment Controller(EEC)
  実験環境全体の初期化処理を行うための制御メッセージや共有するべきパラメータをイベントメッセージとして提供する。また、MAS におけるエージェントの生成、管理も行う。

- Agent
  実験における最小構成要素であり、実験者が振る舞いを定義したエージェントモデルをもとに動作する。エージェントが、MAS の外部に存在するシミュレータやエミュレータからイベントやデータを受け取ったり、外部に向けてインベントやデータを生成することで、エージェントモデルをベースとしたシナリオ制御を実現する。エージェント自身は、あらかじめ定義している規則にしたがい、イベントやデータに応じた状態の遷移や遷移した状態に応じたイベントの生成を行う。

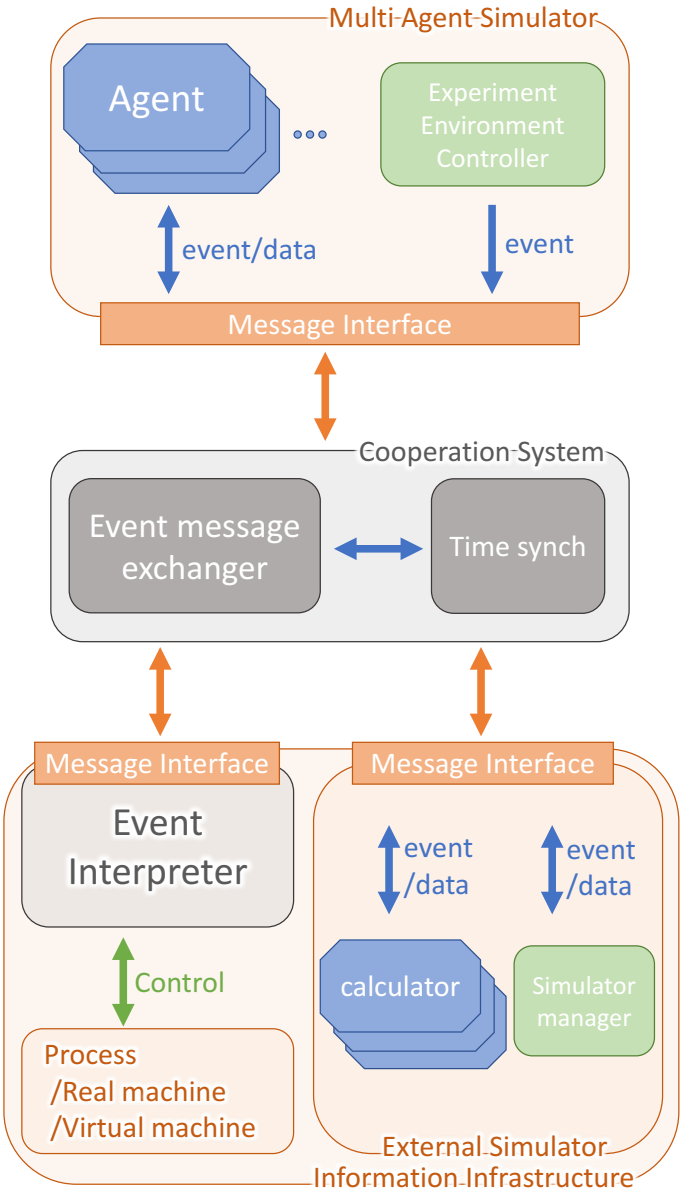
Multi-Agent-Simulator
Agent
Experiment
Environment
Controller
event/data
event
Message Interface
Cooperation-System
Event message
exchanger
Time synch
Message Interface
Message Interface
Event
Interpreter
event
/data
event
/data
Control
calculator
Simulator
manager
Process
/Real machine
/Virtual machine
External-Simulator
Information-Infrastructure

図 5.1: システム概要図

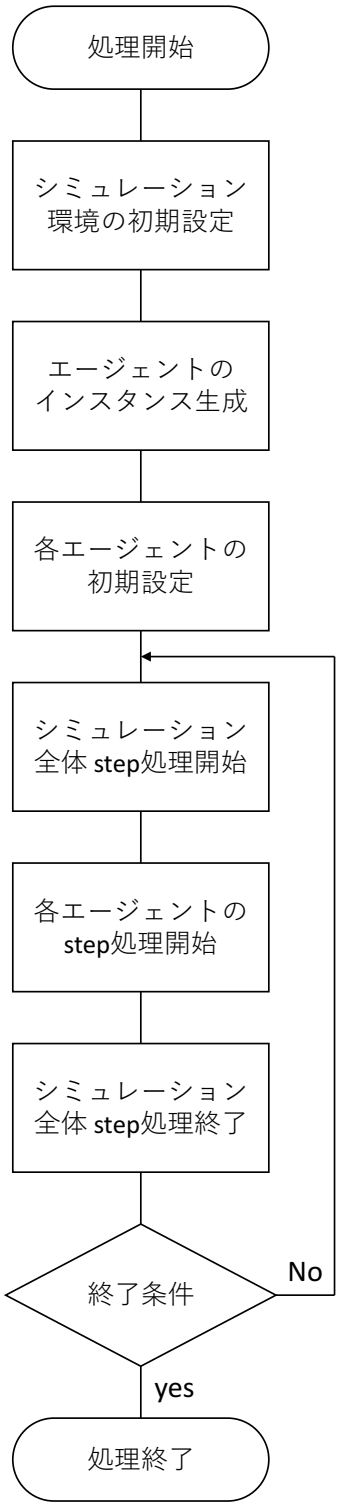
処理開始
シミュレーション
環境の初期設定
エージェントの
インスタンス生成
各エージェントの
初期設定
シミュレーション
全体 step処理開始
各エージェントの
step処理開始
シミュレーション
全体 step処理終了
終了条件
No
yes
処理終了

図 5.2: 実験の流れ

5.1.2 External Simulator

実世界のモノや現象を演算可能な形にモデル化し、そのモデルにしたがって再現するための環境を提供する。 MAS に対し、従属するような形で連携する。実世界を再現するモデルだけでなく、仮想環境のシナリオに沿ってモデル化が可能であれば実世界では試すことの出ない災害時の検証や現実にはありえない状況を想定した分析も可能となる。以下、外部シミュレータ内部の構成要素について述べる。

- Simulator Manger
  外部シミュレータ内の環境構築と管理を行う。MAS の EEC から送信されるイベントメッセージに応じて、エージェント数に対応した計算ノードの作成、編集、削除を行う。

- Calculator
  数理モデルにしたがい計算を行う。計算に必要なデータは、計算ノードが生成された後やサイクル処理の終了後に MAS に存在する Agent とデータ交換することで得る。

5.1.3 ICT 基盤

実機や仮想機器、またその上で動作するアプリケーションなどが実動作する環境を提供する基盤。演算可能な数理モデルによって再現するのではなく、実際に動かすことで分析を行ことを目的としている。外部シミュレータと同様、MAS に対して従属するような形で連携する。

- Event Interpreter
  連携する対象が実動作するソフトウェアやハードウェアである。また、実験したい内容によって、利用するプロトコルやソフトウェアの動作が異なる。しかし、システムで利用されるプロトコルは統一されており、イベントメッセージを介して連携する。また、連携するソフトウェアやハードウェアによって制御する内容が異なるため、交換するイベントメッセージの内容も異なる。したがって、対象に合わせ制御する解釈器が必要となる。イベント解釈器では、エージェントから送信されるイベントを実動作するソフトウェアやハードウェアに合わせた制御プロトコルに変換し、適切な制御を実現する。

- Process・Real Machine・Virtual Machine

  実際に動作する実機、仮想機器、アプリケーションを示す。実動作する機器 (例えば、通話ソフトウェア、Beacon のようなセンサ、ロボット制御に使われる組込み機器など) が、正しく動作するか検証するために連携させる。しかし、検証対象となる機器は、連携のためのインターフェースを持っておらず、また拡張することも現実的ではないため、Event Interpreter を介して連携する。多彩な連携を行うには、Event Interpreter を拡張することで実現する。

5.1.4 協調システム

MAS 内部に存在するエージェントと実機や仮想機器、シミュレータを実験シナリオに合わせて、柔軟に接続、連携するためのシステムである。

5.2 データ交換方式

連携における依存度を低減させるため、メッセージの送信者が特定の受信者を想定せずにメッセージを送信する非同期メッセージングモデルである Pub/Sub モデル (出版/購読型モデル) を利用する。Pub/Sub モデルでは、メッセージのアドレスに当たるトピックと交換内容のコンテンツの組み合わせを交換する。交換する際に送信者と受信者は互い相手を知っている必要はなく、受信する場合にはトピックを指定して事前に購読の登録をするだけで良い。また、メッセージを送信する場合はトピックを指定し公開すればよく、購読登録された相手全てに配信される。

5.3 ID のマッピング

MAS 側で生成されたエージェントには、AID(Agent ID) と呼ばれる識別子が割り当てられる。これは、MAS の中でエージェント同士が連携する際に、連携先となるエージェントを識別するために利用している。このように、エージェントには互いを識別するための情報を持っている。しかし、エージェントに結び付けられる外部シミュレータや ICT 機器は、自身がどのエージェントに関する状態を扱っているのか判別がつかない。したがって、協調システムを通して連携しても、エージェントの元へ情報を届けることができな

い。そのため、外部シミュレータ、ICT 機器がどのエージェントに結び付けられているかを認識させるための情報が必要となる。

5.4 データ形式

json 形式で交換する。これにより、複数のシミュレータや ICT 基盤のシステムと連携を行うことになっても、データを識別するための情報が付加されるため、シミュレータやシステムごとに解釈が異なることを防ぐ。

5.5 時間同期

ICT 基盤で動作する実機や仮想機器は、実時間で動作する。しかし、シミュレーションの実行に必要な時間は、数理モデルの計算量やコンピュータの性能によって変化する。そのため、シミュレーション内で扱うシミュレーション時間と実時間は必ずしも一致しない。実機や仮想機器とシミュレーションを連携させるためには、それぞれの時間軸を同期させる必要がある。この問題を解決するため、図 5.3 に示すように、同期させるタイミングをあらかじめ決めておき、タイミングに合わせて同期をとる方式であるバリア同期方式を適応する。

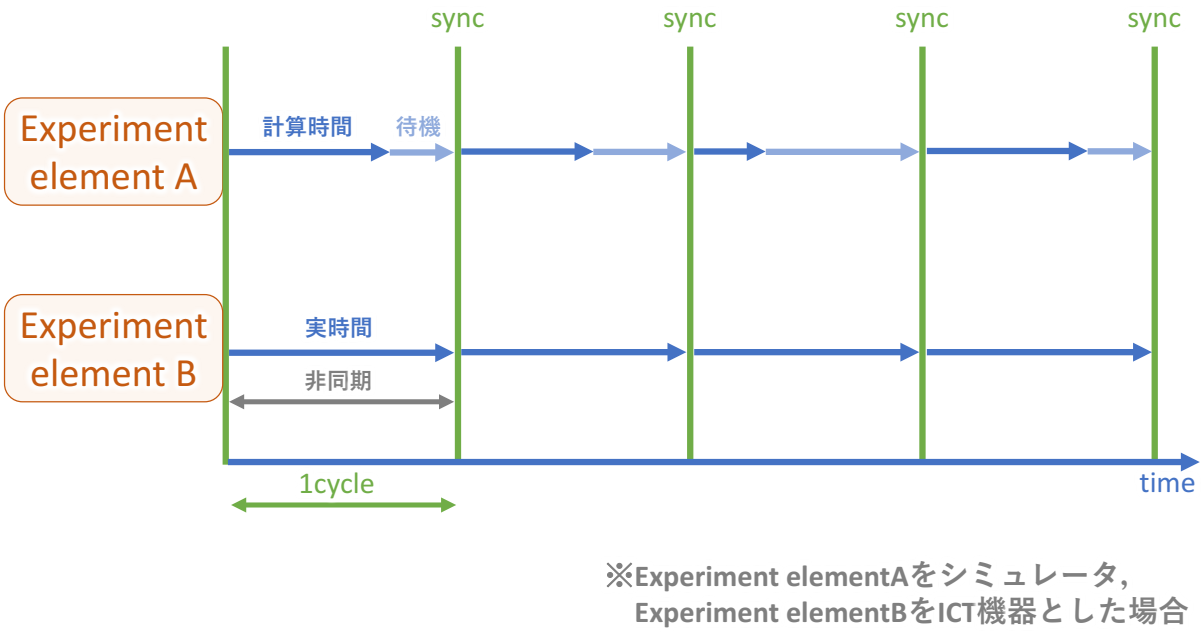
sync
sync
sync
sync
Experiment element A
計算時間
待機
Experiment element B
実時間
非同期
1cycle
time
※Experiment elementAをシミュレータ,
Experiment elementBをICT機器とした場合

図 5.3: 時間同期

第6章　Jonathan：個体モデルによる歩行者エージェントと通話エミュレータの連携

6.1　概要

実験空間全体の制御やエージェントによるシナリオ進行により、MAS を中心とした制御を行う。Jonathan では、個体モデルの携帯電話をもつ歩行者エージェントと SIP(Session Initiation Protocol) クライアントアプリケーションを連携させる。ICT 基盤で実動作する SIP のクライアントと連携することで、通話を介した歩行者エージェントの振る舞いの動的な変化を実現する。

6.2　MAS

実験空間全体の制御やエージェントによるシナリオ進行により、MAS を中心とした制御を行った。

6.2.1　artisoc による実装

MAS には構造計画研究所社製のパッケージソフトウェアである artisoc を使用した。artisoc では、各種の処理を簡便にするための機能が付いている。

6.2.2　実験空間全体の制御

MAS では、最初に一度だけ実行される初期化処理と、周期的に実行されるサイクル処理を行う。

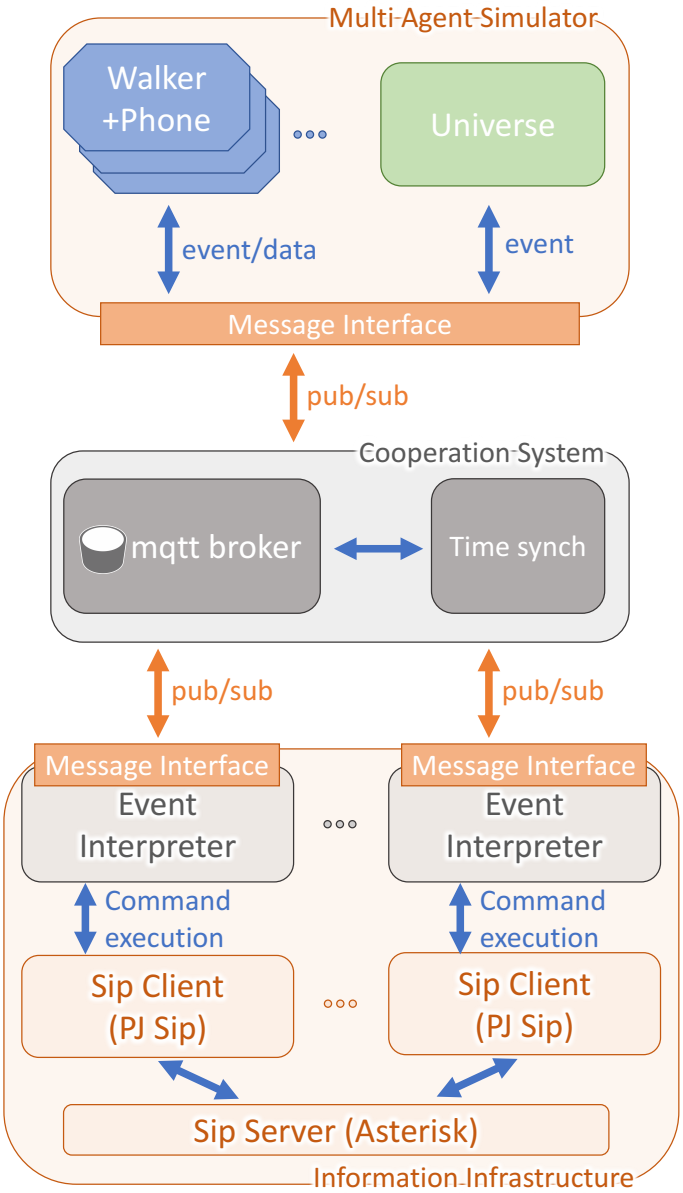

図 6.1: jonathan の概要図

- 初期化処理
  あらかじめ実験者が設定したパラメータにしたがい、歩行者エージェントの生成や各エージェントの初期化が行われる。最初に歩行者エージェントを生成し、初期化情報を持つ Universe 初期化メッセージを送る。

- サイクル処理
  ここでは、全エージェントの通話成功数、失敗数、現在の通話数などの情報を集め送信したり、各歩行者エージェントの状態に合わせたイベントが生成される。

6.2.3 歩行者エージェントのシナリオ

携帯電話を持つ歩行者をエージェントとして扱い、各歩行者エージェントの位置や速度の算出、通話発呼・着呼のイベント生成を行う。その後、各エージェントは、移動速度、方向、位置などのパラメータをランダムに決定し、エージェント初期化トピックを送信する。これにより、通話機器との連携および初期化の設定が完了し、サイクル処理に移行する。各エージェントが移動しながら、通話着呼、発呼、終了の順に現在の状態 (接続中、着呼中、発呼中、通話終了中) を参照して停止た処理 (状態の変更、メッセージの送信など) を行うため、その時々でどのようなイベントを生成するかは異なる。

6.3 通話機器

ICT 基盤で実際に動作させた SIP(Session Initiation Protocol) のクライアントと連携することで実現した。通話コントローラ、SIP クライアント、SIP サーバで構成される。SIP サーバは、IP-PBX ソフトウェアの Asterisk、SIP クライアントは PJSIP を利用し、通話コントローラは ruby によって構築されたプログラムを利用した。(注: 通話コントローラは共同研究の一環で NICT が作成したソフトウェアである)

通話コントローラは、1 つにつき 1 台の SIP クライアントを管理するように展開され、エージェントが発行するイベントに応じて SIP クライアントの操作を行う。例えば、表 **??**の発呼メッセージである call/server/<SID> を受け取ると、接続先 AID を確認し、SIP クライアントを制御して実際に接続先クライアントに発呼する。その結果、通話の成立、不成立に応じて、call/result/success もしくは call/result/busy のイベントを生成し、エージェント宛てに返信する。

このように、イベントに応じて処理を加え、その結果をエージェントに対してフィードバックするような機構になっている。

6.4 協調システム

協調システムは、時刻同期機構と MQTT ブローカで構成される。MQTT ブローカにはオープンソフトウェアの Mosquitto を使用した。連携に使用されるメッセージには、実験空間に関するもの (Universe)、エージェントの状態に関するもの (Agent)、通話に関するもの (call、ringoff) がある。

6.5 データ構造

実装したインターフェースを表 6.1 に載せる。

表 6.1: インターフェース

トピック	コンテンツ
Universe/init	エージェント数、幅、高さ、ブロック数
call/server/<SID>	接続元 AID、接続先 AID
ringoff/server/<SID>	接続元 AID、接続先 AID
call/result/success	接続元 AID、接続先 AID
call/result/busy	接続元 AID、接続先 AID
ringoff/result/success	接続元 AID、接続先 AID

(注：トピックにプレフィックスが付随するものがある)

6.6 実験シナリオ

個体モデルとエミュレータを連動させ、エミュレータ側の変化を通して振る舞いが動的に変化することを確認するため、携帯電話をもつ歩行者エージェントと通話アプリケーションを連動させるシナリオを構築し行った。具体的なシナリオとして、携帯電話を持つ歩行者エージェントが 100 人、それと連動するように sip クライアントを 100 機立ち上げた。通常の常態では、空間上を常に移動するようにし、通話が開始されると移動をやめその場に止まるようにしている。このシナリオは、各歩行者エージェントに他の歩行者がラ

ンダムに登録された電話帳を与え、その電話帳に登録されている相手全員にかけ終えることで終了する。

6.7 結果

図 7.2 は実行中の様子を、図 6.3 は実行の結果を示している。左上にあるグラフは、通話状況を表したものである。青い線が累計接続成立数、灰色の線が累計接続失敗数、橙色の線が現在の通話数を示す。通話アプリケーションを介して、他の歩行者エージェントとの通話が成立すると、累計接続成立数と現在の通話数にカウントされる。また、右上の図は、歩行者エージェント間の通話接続状況を表している。丸で表しているのが歩行者エージェントである。灰色は電話未接続、つまり電話をしていない状態を示し、黄色は通話接続待ちを示す。そして、緑色が通話接続成功の状態を示し、実際に通話アプリケーション側で通話が始まる。また、歩行者エージェントをつなぐ青色の線は、実験のシナリオ全体を通して電話をかける候補者を示しており、この線で繋がる相手からランダムに選び通話をかける。右下の図では、右上で解りづらい歩行者エージェントの位置を示している。青色の丸で示しているのが、歩行者エージェントであり、四角い緑色で示しているのが、障害物エージェントである。歩行者エージェントは、障害物エージェントが設置されている領域に入ることができないようになっており、図からもその様子が読み取れる。

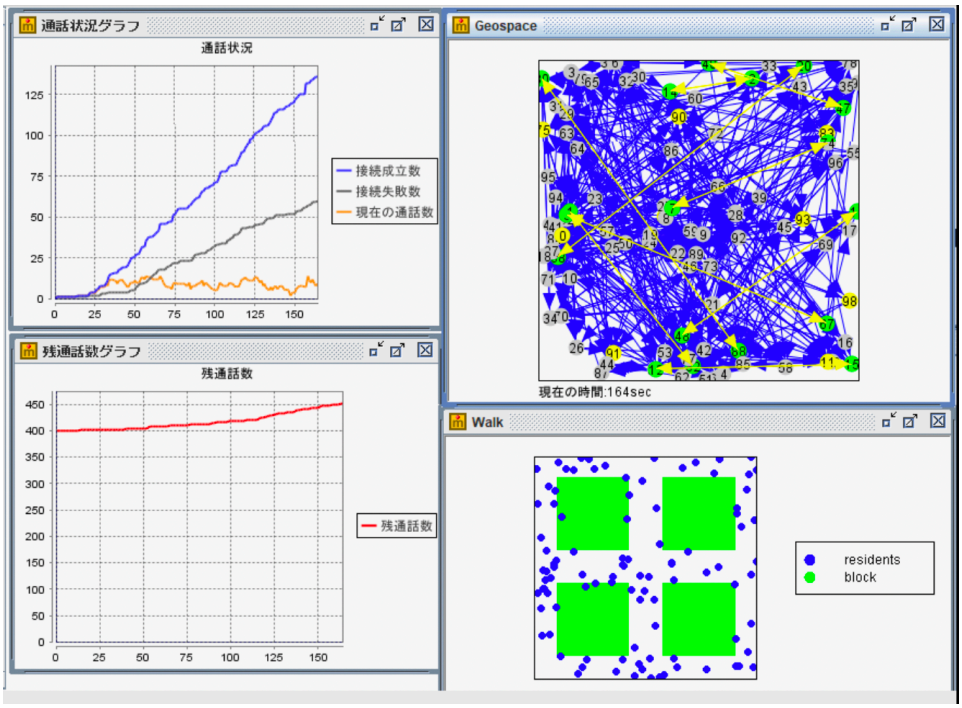

図 6.2: Jonathan:シミュレーション実行中の様子

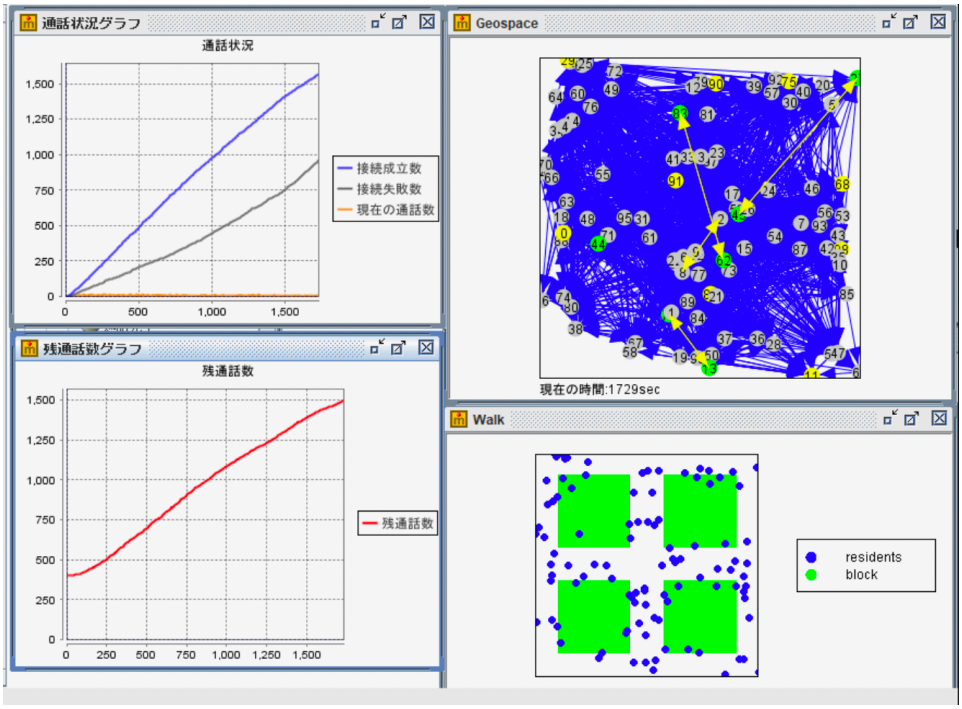

図 6.3: Jonathan:シミュレーション実行結果

第7章　Inferno:空間モデルによる避難者エージェントと火災シミュレータの連携

7.1　概要

実験空間全体の制御やエージェントによるシナリオ進行により、MAS を中心とした制御を行う。Inferno では、空間モデルのフィールドエージェントと火災シミュレータを連携させる。連携のためフィールドエージェントと火災シミュレータの状態を同期させ、同期したフィールドエージェントの状態から避難者エージェントの振る舞いが動的に変化することを実現する。

7.2　MAS

MAS には構造計画研究所社製のパッケージソフトウェアである artisoc を使用した。初期化処理では、避難者エージェントとフィールドエージェントの生成および各エージェントの初期化を行う。最初に歩行者エージェントを生成し、Universe が初期化メッセージを送る。その後、各エージェントは、温度、空間状態、延焼確率などのパラメータを情報をもつ初期化トピックを送信する。

7.3　火災シミュレータ

図 7.1 で示すように、火災シミュレータはシミュレーションマネージャと分割された空間状態を計算する処理機能で構成される。

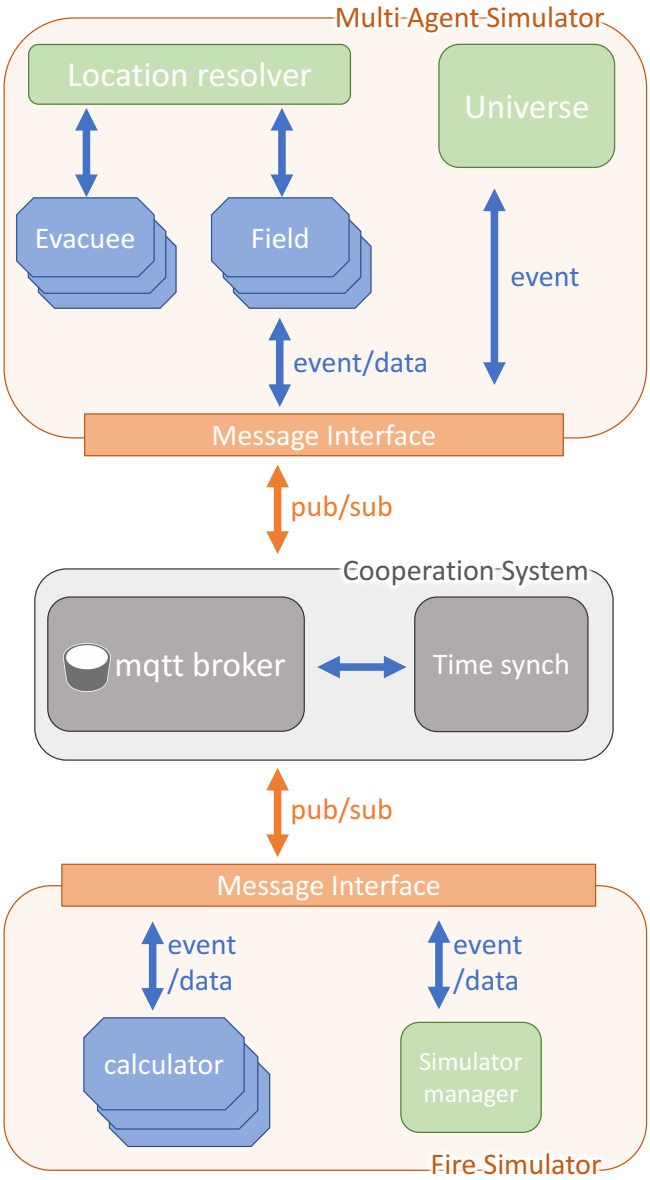

図 7.1: Inferno の概要図

- シミュレーションマネージャ

  シミュレーションマネージャでは、マルチエージェントシミュレータの Universe から初期化イベントである、Universe/init のメッセージを受け取り、メッセージをもとにシミュレーション環境の初期化を行う。

  表 7.1 に示すように、避難者エージェント数、仮想空間全体の横と縦の長さ、メッシュサイズを送信する。フィールドエージェントの個数と同じだけの状態計算処理を立ち上げ各計算処理の初期化を行う。

- 空間状態計算機能

  マルチエージェントシミュレータに存在するフィールドエージェントと連動し、状態に関するデータを同期する。ここで生成された状態計算処理は、フィールドエージェントと 1 対 1 に結び付けられ、ステップ毎に状態の変化が生じればデータ交換によって状態の同期を行う。

7.4 協調システム

協調システムの構成は、Jonathan で使用したものと同じである。連携に使用されるメッセージには、実験空間に関するもの (Universe)、エージェントの状態に関するもの (Field) がある。

7.5 データ構造

実装したインターフェースを表 7.1 に示す。

7.6 location resolver

マルチエージェントシミュレータ内で複数のフィールドエージェントとして展開される。そのため、避難者エージェントが物理量や物理状態を必要とする場合、対応する空間領域を持つフィールドエージェントに直接問い合わせる必要がある。しかし、歩行者エージェントは、空間上を常に移動するため、どのフィールドエージェントに対して問い合わ

表 7.1: インターフェース

トピック	コンテンツ
Universe/init	避難者エージェント数、幅、高さ、メッシュサイズ
Field/<AID>/init	AID、温度、空間状態、延焼確率、延焼時間
Field/<AID>/cycle	AID、温度、空間状態、延焼確率、延焼時間

(注：トピックにプレフィックスが付随するものがある)

せれば良いのかわからないという問題がある。そこで、location resolver では歩行者エージェントの位置情報から対応するフィールドエージェントの AID を取得できるようにする機能を提供する。

7.7 実験シナリオ

上で述べたようなシステムの動作確認のため、歩行者エージェントとフィールドエージェントが連携し振る舞いが変化するようなシナリオを構築し行った。具体的なシナリオとして、マップ上に存在する避難者エージェントが 100 人、空間をメッシュ状に分割したフィールドエージェントを 400 個構築した。避難者は、延焼しているフィールドエージェントを避けるように移動する。もし逃げきれず、延焼中のフィールドエージェントの上に存在した場合は、炎に巻き込まれたと判断し避難をやめる。シナリオは、延焼したフィールドが全て燃え尽きることで終了する。

7.8 結果

図 7.2 は、シナリオ実行中の様子を表している。青色や黄色の丸が避難者エージェントを示す。左下の図は避難者のみを可視化した図で、フィールドエージェントと分離することで、移動の様子を見やすくしている。また、右下の図はフィールドエージェントの延焼の様子と避難者エージェントの状態を示している。避難者エージェントが延焼しているフィールド上に存在する時、火災に巻き込まれたと判断し自身の状態を変化させる。この時、避難者エージェントの色を黄色に変える。これにより、避難者エージェントがフィー

ルドエージェントと情報を交換し、状態を変化させることで、シナリオに変化が生じたことがわかる。

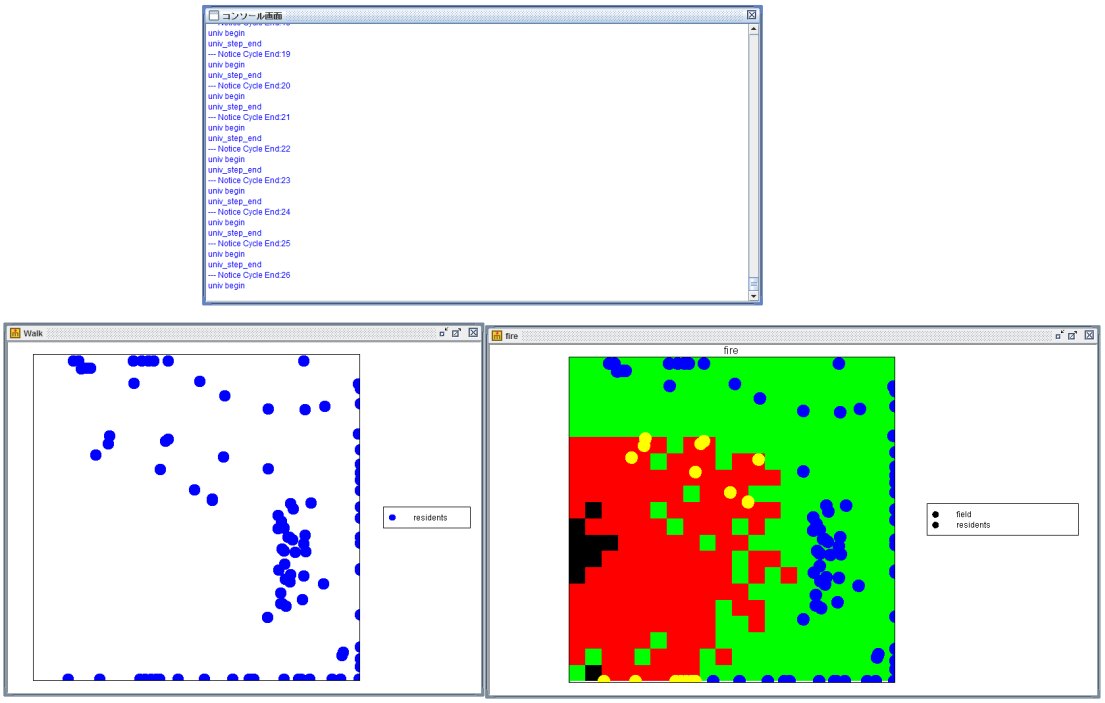

図 7.2: inferno:シナリオ実行中の様子

第8章　考察

本章では、提案手法の考察を行う。

8.1　統合的な制御の実現

エージェントと実験構成要素との協調動作

シミュレータやエミュレータを統合的に制御するため、個体モデルと空間モデルを定義し、モデルにしたがってエージェントにデータを集約し操作ができるエージェント化を行った。それぞれのモデルに合わせ、実際にエージェント化できるかを実証するため、Jonathan と Inferno を実装した。

Jonathan では、個体モデルにしたがい携帯電話を持つ歩行者エージェントと実動作する SIP アプリケーションが結び付けられ、SIP アプリケーションが持つ、通話に関する状態データが歩行者エージェントに集約される。

また、集約されたデータをもとに通話に関する状態に変更があれば、歩行者エージェントの状態が通話中に遷移する。逆に、歩行者エージェントが自発的に発呼し接続中の状態に遷移した場合は、実際に発呼するように SIP アプリケーションにイベントメッセージを送信する。これにより、個体モデルによるエージェントを主体とする協調動作を実現している。

Inferno では、空間モデルにしたがいフィールドエージェントと火災シミュレータが結び付けられ、空間の状態や物理量に関するデータがフィールドエージェントに集約される。

集約されたデータに変更を加えると、変更された状態データが火災シミュレータと共有され同期される。その結果、火災シミュレータ内で他空間に影響を及ぼし、燃焼中であればその状態が広がっていき、空間に対応するフィールドエージェントへ同期される。図 7.2 では、他空間へ延焼する様子が見て取れる。これにより、空間モデルによるエージェントを主体とする協調動作を実現している。

個体モデルエージェントと空間モデルエージェントの連携

Inferno では、空間モデルのエージェントと物理シミュレータとの協調動作だけではなく、個体モデルである避難者エージェントと空間モデルであるフィールドエージェントの連携も実現している。図 7.2 では、避難者エージェントがフィールドエージェントと連携して、燃焼している空間を避けるように移動していることがわかる。

避難者エージェントは、物理現象を知覚するために物理シミュレータと直接連携するのではなく、フィールドエージェントへ問い合わせることで空間に関する情報を得ている。これは、提案手法で述べたとおり、複数の物理量や物理状態を空間に結びつけるためである。これによって、個体モデルのエージェントは連携するべき物理シミュレータが増えても、空間モデルのエージェントに問い合わせるだけで済むため、個体モデルにおける記述の負担を軽減できる。

全てのシミュレータやエミュレータのエージェント化

上記より、個体モデルのエージェント化、空間モデルのエージェント化を実現し、互いに協調動作できることを示した。したがって、全てのシミュレータやエミュレータを個体モデルもしくは空間モデルのエージェントとしてエージェント化することが出来れば、エージェントを主体とした各シミュレータ、エミュレータとの連携ができ、エージェントが協調動作するマルチエージェントモデルにより統合的な制御を実現できる。

8.2 複数のシミュレータやエミュレータとの協調動作

各々のシミュレータやエミュレータをエージェント化することで、マルチエージェントシミュレータから統合的に制御することができる。しかし、Jonathan、Inferno において、同じ実験内で 1 種類の個体モデルもしくは空間モデルのエージェント化が実現可能であること示したが、複数のシミュレータやエミュレータが混在するような実験環境で統合的な制御が実現可能であるかは示していない。そのため、議論の余地がある。以下、考慮するべき項目について検討する。

8.2.1 データ構造の分類と適応検討

Jonathan では、携帯電話を持つ歩行者が電話をかけるというシナリオで実験を行った。しかし、日常の利用では電話だけなくメールを送信したり、メッセンジャーアプリケーションを利用して簡単なメッセージ交換をしたりするような利用も考えられる。したがって、通話機能だけではなく、テキストメッセージ交換機能も同時に利用するような、複数の動作を組み合わせた実験シナリオがあると想定し、そのために必要となる連携のためのデータ構造を検討する。

データの構造として、状態共有型と操作通知型が存在する。ここで、表 8.1 のように動作を示す情報がトピックに埋め込められている方法を操作通知型と定義し、トピックの中にエージェントを識別する ID のみを入れ、動作や状態に関する情報はコンテンツの中に全て取り込んで連携する方法を状態共有型と定義する。

表 8.1: データ構造の分類の例

データ構造の型	トピック	コンテンツ
状態共有型	Agent/<ID>	接続元 ID、役割、x 座標、y 座標、移動速度、方向、通話フラグ、接続先 ID
操作通知型	Call	接続元 AID、接続先 AID

状態共有型

状態共有型では、エージェントが保持する全ての状態がそのまま共有されるため、共有するべきエージェントの ID さえ把握しておけば全ての情報を取得できる。しかし、何に関連する状態の共有なのかをトピックによって明示的に示していないため、変更された状態から各々のシミュレータやイベント解析器が変更されたかを判断し、内容を反映する必要があるという問題が存在する。さらに、必要ではない情報も一緒に転送されるため、何か変更があるたびに転送されたデータ内容を検査する必要があり、明らかに無駄なデータ処理が多く発生するようになる。これを解決するため、コンテンツ内でフィルタリングに必要なメタ情報を付加する方法が考えられる。実験における連携では、基本的に通話する、メールを送信するなど動作に変更が加わった際に行われる。また、動作ごとに必要なパラメータが異なると考えられる。そのため、図 8.1 に示すように動作に関するメタ情報

の入れ子になるように関連情報を置き、fileter の中に格納されている情報の有無で、変更があったかを判断できる。

```
{
  myID: 1,
  role: human,
  filter: [call]
  move: {
    x: 12,
    y: 14,
    speed: 3,
    direction: 45
  },
  call: {
    calling: true,
    talking: false,
    targetID: 3
  }
}
```

図 8.1: 動作に関するコンテンツフィルタの例

操作通知型

操作通知型では、動作を示す情報がトピックとして利用され、動作に必要な情報がコンテンツ情報として転送される。そのため、トピックの段階でどの動作に関する変更かを判別することができ、かつ動作に関連するシミュレータやイベント解析器のみにメッセージ転送を行えばよくなるので、トピックによるフィルタリングを行えるという利点がある。しかし、動作ごとに通知するためのメッセージを発行するので、複数のシミュレータやエミュレータと連携し動作が複雑になるほど、メッセージが増加すると考えられる。

適応検討

状態共有型は、マルチキャストするような利用に適している。状態共有型のコンテンツ内でフィルタリングを行う場合と比べ、操作通知型はトピックによるフィルタリングを行

うことになるため、pub/sub モデルの特徴からメッセージの転送そのものを防ぐことができ、メッセージの削減に貢献できる。

共有データの整合性の確保

複数のシミュレータやエミュレータが連携をする場合、共有するデータへの操作が重複する可能性がある。そのため、一度エージェントにデータ集約し処理することで、整合性を確保する。

第9章　おわりに

本章では、本論文におけるまとめと今後の課題と展望について述べる。

9.1　まとめ

本論文では、社会での利用を想定した事前検証を実現するため、マルチエージェントシミュレーションのエージェントと外部シミュレーション、エミュレーションを結び付け、エージェント中心にシミュレーションシナリオの制御を行う エージェント化を提案した。エージェント化をするにあたり、対象の特性に合わせエージェントモデルを「個体モデル」と「空間モデル」という 2 種類に分類した。これら 2 種類のエージェントモデルにより、エージェントを中心としたシミュレータやエミュレータの統合的な制御が実現できた。

9.2　今後の課題と展望

9.2.1　複数のシミュレータやエミュレータとの連携

本論文で、考察のみだった複数のシミュレータやエミュレータの連携を実現するための実装が課題として残っている。また、連携する対象が増えるほど、交換するメッセージも増えるため、環境に適したデータ構造の選択が重要である。

9.2.2　抽象度の高いシナリオ記述

本論文では、エージェントのシナリオ記述 (振る舞い定義) を、状態変数を管理することで実現した。しかし、人間の直感的な動きなどを記述するためには、どのような状況で、どんな状態が変化するかなどを考え実装する必要がある。そこで、柔軟性を向上させるために、より抽象度の高いシナリオ記述で実験できることが重要である。

謝辞

本研究を行うにあたり、多くの方から多大なご助言やご助力を頂きました。それらの方々のご協力がなければ、本研究は成り立ちませんでした。心から厚くお礼申し上げます。本研究を進めるにあたり、指導教員である篠田陽一教授には様々な助言、適切な御指導を賜りました。心から深く感謝します。また、助言を頂いた主テーマ審査員である丹康雄教授、副テーマ指導教員である田中清史教授に感謝致します。本研究室の知念賢一特任准教授、宇多仁助教には、研究に関して活発な議論や多大なご指導を賜りました。心から感謝致します。情報通信機構北陸リサーチセンターの湯村翼氏、安田真吾氏、井上朋哉氏、榎本真俊氏、明石邦夫氏、三浦良介氏、宮地利幸氏には研究に関して様々な助言、ご指導を賜りました。心から感謝致します。本研究室の博士後期課程の太田悟史氏、阿部博氏には、研究に関して活発な議論、ご指導を賜りました。心から感謝致します。本研究室修了生の廣瀬真人氏、三木晶司氏には、研究に関して活発な議論、ご指導を賜りました。また、研究生活を送る上で様々なご助力を頂きました。心から感謝致します。本研究室の博士前期課程の可児友邦氏、大川昌寛氏、橋本光世氏、村上正樹氏、阿波史和氏、砂川真範氏、淺葉祥吾氏、広瀬太志氏、三島航氏、宮崎駿氏、山口礼央氏、小松源氏には活発な議論や、研究生活を送る上で様々なご助力を頂きました。心から感謝致します。最後に研究や生活で支えてくれた家族へ心から感謝致します。

参考文献

[1] マルチエージェント・シミュレーションとは, http://mas.kke.co.jp/modules/tinyd4/index.php?id=16

[2] Asterisk, https://www.asterisk.org

[3] PJSIP, http://www.pjsip.org/

[4] 石田 亨, 福本 理人, インタラクション設計言語 Q の提案, 人工知能学会論文誌, Vol.17, No.2, pp.166-169, 2002

[5] 中島 悠, 服部 宏充, マルチエージェントシミュレーション統合のためのシミュレータアーキテクチャ, 情報処理, Vol.53, No.11, pp.2477-2484, Nov.2012

[6] 古市 昌一, 和泉 秀幸, 分散シミュレーションのための統合基盤アーキテクチャHLAの紹介, 情報処理, Vol.41, No.12, Dec.2000.

[7] 鳥居 大祐, 石田 亨, Q と CORMAS の結合：多層マルチエージェントシミュレーションの挑戦, 人工知能学会全国大会 (第 18 回) 論文集, 2H1-01, 2004

[8] 谷塚 俊輔, 中西 英之, 石田 亨, 阿部 伸之, 山田 常圭, 建物火災を対象とした疑似体験型マルチエージェントシミュレータの開発, 3B1-04, 2005

[9] 飯田 進史, 五十嵐 孝浩, 高木 茂知, 分散シミュレーション統合技術を適応した水害危険管理行動支援システムに関する一考察, 河川技術論文集, 第 9 巻, 2003

[10] 宇都 正哲, 植村 哲士, 神尾 文彦, 小林 庸至, 北崎 朋希, 人口減少時代におけるインフラ整備の問題と対応策, 知的資産創造, pp.78-95, 2009

[11] 宇都 正哲, 次世代社会インフラの動向, OKI テクニカルレビュー, Vol.82, No.2, 2015

[12] 中島 悠, 椎名 宏徳, 山根 昇平, 八槇 博史, 大規模マルチエージェントシミュレーションにおけるプロトコル記述と実行基盤, 電気情報通信学会論文集, Vol.J89-D, No.10, pp.2229-2236, 2006

[13] 宮地 利幸, 中田 潤也, 知念 賢一, Razvan Beuran, 三輪 信介, 岡田 崇, 三角 真, 宇多 仁, 芳炭 将, 丹 康雄, 中川 晋一, 篠田 陽一, StarBED:大規模ネットワーク実証環境, 情報処理, Vol.49, No.1, pp.57-70, Jan.2008

[14] 山本 学, 田井 秀樹, 水田 秀行, 1 億エージェントを用いたエージェントベースシミュレーションの実現への考察, 電気情報通信学会論文誌, Vol.J90-D, No.9, pp.2423-2431, 2007

[15] 服部 宏充, 十見 俊輔, 中島 悠, 大規模マルチエージェントシミュレーションに基づく社会システムデザインの可能性, 電子情報通信学会論文誌, Vol.J100-D, No.2, pp.180-193, 2017