

エージェントベースモデリングのための 対話型モデリング手法の提案とその実践

三浦 政司

鳥取大学大学院工学研究科 機械宇宙工学専攻

要旨

独立している個々の要素の振る舞いが簡単に理解できたとしても、多数の要素が互いに影響を及ぼしあって相互作用を持つと、その全体的な振る舞いを説明したり理解したりすることが難しくなり、複雑さが生まれる。インターネットや交通の発達によってヒト・モノ・情報がとてつもない規模で接続され、グローバルなレベルから個人のレベルまで様々な相互作用を持つようになった現代社会は極めて複雑であり、その振る舞いを予測することが困難となった。このような複雑性を扱う新しい分野の科学は「複雑系科学」と呼ばれる。従来の科学が対象を要素に分析し、個々の要素の振る舞いについて詳しく知ろうとする要素還元的な考え方をベースにしていたのに対して、複雑系科学は要素に還元できない全体的な性質があるという創発性に着目した世界観を持った新しいパラダイムであると言える。この新しいパラダイムは、ますます複雑化する現代社会において、モノゴトを理解したり予測したりして諸問題を解決するにあたって極めて重要な考え方となっている。

複雑系科学が扱う諸手法のうち、個々の要素間の相互作用がボトムアップ的に全体の性質や振る舞いを生み出すという創発性を最も自然な形で表現するアプローチの一つとして、エージェントベースモデリングやそれを用いたマルチエージェントシミュレーションがある。エージェントベースモデリングやマルチエージェントシミュレーションでは、社会を構成する要素である人や組織を意思決定の主体（エージェント）として扱い、社会現象をエージェントおよびエージェントの振る舞いの集合として記述する。そこでは、個々のエージェントの振る舞いとエージェント間およびエージェントと環境間の相互作用がモデルとして記述され、現象全体についてはモデルとしては記述しない。個々のエージェントの振る舞いと相互作用がモデル化された上で計算され、どのような全体的振る舞いが生じるのかをシミュレーションする。このようにエージェントベースアプローチによる社会シミュレーションは、複雑系科学の基盤的な考え方である創発性を自然な形で取り入れており、複雑な現象を理解したり予測したりするにあたって極めて有用な手法である。このような考え方や手法は、複雑さに起因する社会の諸問題を解決したり、**Society5.0** と呼ばれる近未来のテクノロジー埋め込み型の社会をデザインしたりするにあたって、重要な役割を果たすことが期待されている。本研究では、これからの社会において果たす役割の大きいエージェントベースなアプローチのモデリングやシミュレーションに着目し、その社会応用を拡大、促進す

ることを目的としたアイデアの提案とその実践に取り組む。

ここまで述べてきたように、社会的な意義と期待の大きいエージェントベースモデリングであるが、現在のところ実際に社会応用が実現しているのは、交通・物流・避難などの一部の分野にとどまっており、今後はより幅広い分野での社会応用や様々なケースへの適用が期待されている。また、特定の地域・組織・環境下における現象のモデリングやシミュレーションなど、具体化・詳細化されたニーズも多くある。例えば、特定の地域や組織における人々の行動特性や慣習などを考慮したモデリングや、現実にある特定の建物の利用・運用に関するシミュレーションなどが相当する。本研究では、上記のような社会応用に際する課題に対して、エージェントベースモデリングに関する専門性を持たないが、着目する現象に関する知見を持つ人々との協働による対話型のモデリングプロセスを提案する。エージェントに相当する人自身や関連する業務に取り組む実務家など、対象となる社会現象に深く関連する人々は、エージェントの振る舞いに関する詳細な経験値・暗黙知を有している。このようなステークホルダらと協働でモデリングに取り組み、その経験値・暗黙知を抽出してモデルに組み込むことができれば、これまで一般的なモデルが提案されていない現象や十分なデータの取得・蓄積が難しい現象などを対象としたエージェントベースモデリングが可能となり、行動要因やパラメータの抽出漏れや妥当性に対して精度の高いフィードバックを得ることができる。

このようなアプローチのアイデアを実現するための具体的なプロセスとして、本研究では「ゲームをつくることを通して対話しながら協働でモデリングを行う」という手法を考案し、GBSP (Game-Based Situation Prototyping : ゲームに基づく状況プロトタイプング) と名付けてツールやメソッドを整理した。また、そのような手法が実際に有効であることを実践的に確認するために、組織におけるチーム形成に関する研究、企業間のポジショニングに関する空間的競争の研究、電力ネットワークにおける需給制御に関する研究など、社会シミュレーションを軸とする様々な研究活動に対して GBSP を適用し、協働モデリングを実践した。その結果、それぞれの研究活動において効果的で有用性の高いモデリングを実現することができ、優れた研究成果を生み出すことができた。また、研究活動だけでなく教育活動の場などにも適用し、同様に優れた結果を生み出した。このような実践的な取り組みを通して GBSP が高い有効性を持つことを確かめることができた。さらに、GBSP の有効性を定量的に評価するために、既存手法との比較による評価実験にも取り組み、状況共有に関する評価手法を構築するとともに、ある側面における GBSP の有効性を示すことができた。

目次

要旨.....	3
第1章 緒論.....	9
1.1. 複雑化する現代社会と社会シミュレーション.....	9
1.2. エージェントベースモデリング.....	11
1.3. 社会応用における課題.....	14
1.4. 本研究のアプローチ.....	16
1.5. 本研究の目的.....	17
1.6. 本論文の構成.....	18
第2章 対話型モデリング手法「GBSP」の提案.....	22
2.1. アプローチ.....	22
2.2. ゲームとエージェントモデル.....	22
2.3. Game-Based Situation Prototyping.....	23
2.4. 既存手法との比較.....	25
2.5. GBSP の具体例.....	26
第3章 学際研究チームの形成モデル.....	35
3.1. GBSP の実践.....	35
3.1.1. 協働モデリングへの参加者.....	35
3.1.2. GBSP の実践結果.....	36
3.1.3. GBSP の効果.....	37
3.2. 研究の背景・目的.....	37
3.3. エージェントモデル.....	38
3.3.1. 組織とエージェント.....	38
3.3.2. 研究テーマ.....	38
3.3.3. チームの成立.....	39
3.4. シミュレータの構築.....	40
3.4.1. コントロールパネル.....	40
3.4.2. 空間(研究組織).....	41
3.4.3. 研究テーマ空間.....	43

3.4.4. 時系列グラフ	43
3.5. シミュレーション結果	44
第4章 空間的競争モデル	47
4.1. GBSP の実践	47
4.1.1. 協働モデリングへの参加者	47
4.1.2. GBSP の実践結果	47
4.1.3. GBSP の効果	48
4.2. 研究の背景・目的	49
4.3. 2次元上の空間的競争モデル	50
4.4. エージェントベースモデル	51
4.5. シミュレーション結果	55
4.6. まとめ	57
第5章 分散型電力需給制御モデル	58
5.1. GBSP の実践	58
5.1.1. 協働モデリングへの参加者	58
5.1.2. GBSP の実践結果	59
5.1.3. GBSP の効果	59
5.2. 研究の背景・目的	60
5.3. 問題設定	61
5.3.1. ネットワークモデル	61
5.3.2. 最適化アルゴリズムによる価格調整	63
5.3.3. 従来の分散型価格調整法	66
5.3.4. 従来法の問題点	67
5.4. 提案手法	68
5.5. シミュレーション	72
5.5.1. シミュレータの開発	72
5.5.2. シミュレーション結果	73
5.6. まとめ	76
第6章 初学者によるモデリングへの適応	77
6.1. GBSP の実践	77

6.1.1. 協働モデリングへの参加者.....	77
6.1.2. GBSP の実践結果	77
6.1.3. GBSP の効果.....	78
6.2. 背景・目的.....	79
6.3. 授業設計	79
6.3.1. 達成目標	80
6.3.2. 授業の流れ.....	80
6.4. マルチエージェントシミュレーションのチュートリアル	81
6.4.1. 開発プラットフォーム.....	81
6.4.2. サメー小魚モデル.....	82
6.5. 行動ルール.....	82
6.6. シミュレーション結果.....	83
6.7. モデリングプロセス	84
6.7.1. 協働型モデリング手法 GBSP の導入	84
6.8. クラウド型システムモデリングツールの活用	85
6.9. 実践結果	86
6.10. 授業の様子	86
6.11. アンケート評価.....	88
6.12. まとめ	89
第7章 有効性評価.....	90
7.1. 目的	90
7.2. 状況共有に関する評価実験.....	90
7.2.1. 問題状況	90
7.2.2. 比較手法	91
7.2.3. 実験の流れ.....	92
7.2.4. 共有された情報の評価.....	94
7.3. 状況共有の定量的評価.....	96
7.3.1. プレテスト／ポストテスト結果の点数化	96
7.3.2. ポストテストの結果	97
7.3.3. プレテスト／ポストテストの比較.....	98

7.3.4. 共有度による評価.....	99
7.3.5. ヒアリング結果と共通認識度.....	101
7.3.6. 共通認識度による評価.....	103
7.3.7. 定量的評価のまとめ.....	104
7.4. 形式概念分析による質的評価.....	104
7.4.1. 形式概念分析の方法.....	105
7.4.2. 形式概念分析の適用.....	108
7.4.3. GBSP 群に対する適用.....	109
7.4.4. リッチピクチャ群に対する適用.....	110
7.4.5. GBSP 群の分析結果.....	111
7.4.6. リッチピクチャ群の分析結果.....	113
7.4.7. 形式概念分析による評価のまとめ.....	115
第8章 まとめと今後の展望.....	116
8.1. 研究内容.....	116
8.2. 研究成果.....	117
8.2.1. 対話型協働モデリング手法の構築.....	117
8.2.2. 提案手法の有用性確認.....	118
8.2.3. 提案手法を活用した研究の成果.....	119
8.2.4. 提案手法を活用した教育の成果.....	120
8.2.5. 協働モデリング手法に関する評価手法の構築.....	120
8.2.6. 成果のまとめ.....	121
8.3. 今後の展望.....	121
8.3.1. ゲームからシミュレーションへの変換.....	121
8.3.2. ステークホルダとの協働による社会デザイン.....	123
参考文献.....	125
謝辞.....	134
付録.....	136

第 1 章 緒論

1. 1. 複雑化する現代社会と社会シミュレーション

近代における交通網・物流網の発達により，人間社会はグローバルに接続され，遠く離れた地域や文化の間が互いに影響し合うようになった[1][2]．さらに 20 世紀末に登場したインターネットによってグローバル化は著しく加速され，情報を介した相互作用が社会に大きな影響を与えるようになった[3]．また，グローバルな単位だけでなく，個人の単位でもスマートフォンや SNS などによって情報が接続され，ヒト・モノ・情報の相互作用ネットワークが短い期間で大きく姿を変えている[4][5][6]．その結果，現代社会は極めて複雑になり，その振る舞いを予測することが困難となった．

「複雑」という言葉は，モノゴトの関係性が入り組んでいて理解や説明が困難な状態を指す．独立している個々の要素の振る舞いが簡単に理解できたとしても，多数の要素が互いに影響を及ぼしあって相互作用を持つと，その全体的な振る舞いを説明したり理解したりすることが難しくなり，複雑さが生まれる[7]．前述のように現代社会はグローバルなレベルから個人のレベルまでが巨大な相互作用ネットワークで結びついており，ますます複雑になっている．そのため，様々な場面で理解や予測が困難な現象が生じ，しばしば社会的な問題を生み出している．

多数の要素の間に相互作用が生じることで，個々の要素の性質や振る舞いからは直接説明することのできない全体的な性質や振る舞いが生まれることは創発性[8][9]と呼ばれ，創発性について扱う科学研究は総じて「複雑系科学」と呼ばれる[10][11][12]．従来の科学が対象を要素に分析し，個々の要素の振る舞いについて詳しく知ろうとする要素還元的な考え方をベースにしていたのに対して，複雑系科学は要素に還元できない全体的な性質があるという創発性に着目した世界観を持った新しいパラダイムであると言える．この新しいパラダイムは，ますます複雑化する現代社会において，モノゴトを理解したり予測したりして諸問題を解決するにあたって極めて重要な考え方となっている．

複雑系科学はネットワーク理論[13]，カオス理論[14]，進化計算[15]など広範な理論や手法を含むが，個々の要素間の相互作用がボトムアップ的に全体の性質や振る舞いを生み出すという創発性を最も自然な形で表現するアプローチの一つとして，エージェントベースモデリング[16][17]やそれを用いたマルチエージェントシミュレーションなども含まれる．

本論文では、現代社会を理解するための重要な考え方である複雑系科学のうち、特にこれらのエージェントベースなアプローチに着目する。

エージェントベースモデリングやマルチエージェントシミュレーションについての詳細は次節以降で述べるが、これらはしばしば社会シミュレーションの手法として用いられる。社会シミュレーションとは、社会現象をモデル化してコンピュータ上に実装し、その振る舞いを計算によってシミュレートする手法である[18][19]。社会シミュレーションの目的は大きく2つに分けられる。ひとつは分析型と呼ばれ、社会現象を理解したり説明したりすることを目的としている。もうひとつは創造型と呼ばれ、様々な条件下での社会現象を予測することで、新しい社会システムをデザインしたり、最適化したり、事前検証や訓練に用いたりする[20]。初期の社会シミュレーションとして有名なものには、人種や文化の違いによる住み分け現象を説明した Shelling の分居モデル[21]や、曖昧な状況下における意思決定を説明するゴミ箱モデル[22]などがあり、分析型の目的に沿って研究が進められてきた。社会シミュレーションの手法には微分方程式を用いた解析的なアプローチを用いたものなど様々なものがあるが、エージェントベースアプローチによる社会シミュレーションでは、社会を構成する要素である人や組織を意思決定の主体（エージェント）として扱い、社会現象をエージェントおよびエージェントの振る舞いの集合として記述する。そこでは、個々のエージェントの振る舞いとエージェント間およびエージェントと環境間の相互作用がモデルとして記述され、現象全体についてはモデルとしては記述しない。個々のエージェントの振る舞いと相互作用が計算され、結果としてどのような全体的振る舞いが生じるのかをシミュレーションする。このようにエージェントベースアプローチによる社会シミュレーションは、複雑系科学の基盤的な考え方である創発性を自然な形で取り入れており、複雑な現象を理解したり予測したりするにあたって有用な手法の一つとなっている。人や組織、それらと相互作用を持つ機能システムなど、個々の要素の振る舞いをモデル化することができればシミュレーションが実行可能であり、創発的な振る舞いを予めモデルに取り入れる必要はない。一般的に複雑な現象において創発的な振る舞いを直接モデル化することは困難である。一方で、個々の要素の振る舞いをモデルとして記述することは可能なことが多く、エージェントベースアプローチな考え方は、複雑な社会現象をモデル化したりシミュレーションしたりするための実行可能な手段を与えてくれる[23][24]。

エージェントベースアプローチの社会シミュレーションが対象とする現象は交通・人流・物流・エネルギー・防災／減災・組織・ビジネスなど多岐に渡り[25][26]、人や組織など個々

の意思決定主体（エージェント）の行動をモデル化さえできれば、あらゆる社会現象を対象とすることができる。一方で、妥当性の高いエージェントのモデルを構築することはそれほど容易ではなく、様々な課題点がある。本論文ではそのような課題に焦点をあて、有用性の高いモデル化手法を提案する。

エージェントベースな社会シミュレーションにおいて意思決定主体（エージェント）としてモデル化できるのは、人や組織だけでなく、それらと相互作用して自律的に機能を発揮する各種機能システムを含む。例えば、信号機や券売機、IoT システムの各種端末、生産工場内における各種自動機械、自動運転が普及した社会を想定した場合の自動運転車などがこれに相当する。政府が策定した第 5 期科学技術基本計画[29]には IoT システムやロボットシステムが社会の中に埋め込まれ、サイバー空間と現実社会が高度に融合した「超スマート社会 (Society5.0) [30][31][32]」の概念が目指すべき方向として明示されており、人と機能システムの相互作用を含む複雑な系についての理解・説明・予測を可能とするエージェントベースアプローチの社会シミュレーションは、今後の社会システムを設計する上で欠かせない手法の一つである。

ここまで述べてきたように、エージェントベースモデリングやマルチエージェントシミュレーションなどのエージェントベースアプローチな社会をシミュレーションは、ますます複雑になっていく現代社会や、テクノロジーとさらに深く融合していくこれからの未来社会において、様々な現象を理解したり社会システムをデザインしたりするための極めて重要な手法である。本研究は、それらの手法におけるモデル化のプロセスと社会応用に関する課題に着目してユニークな観点からの提案を提示するものであり、社会的な意義の高い研究となっている。

1.2. エージェントベースモデリング

エージェントベースモデリングは、与えられたルールに基づいて自律的に行動・意思決定・学習などを行う「エージェント」を基本的な単位とし、多数のエージェントの集合として現象を記述する手法である[33]。また、そのように記述されたモデルはエージェントベースモデルと呼ばれ、エージェントベースモデルを計算機上にプログラミングしてその振る舞いをシミュレートとしたものはマルチエージェントシミュレーションと呼ばれる[34]。

エージェントベースモデルやマルチエージェントシミュレーションにおいて各エージェントは、他のエージェントの状態や環境（エージェントが配置され行動する世界およびその

世界と関連付けられたパラメータ)を観測し、その結果に基づいて自身の行動を決定する。このときの観測結果と選択される行動の対応関係は「行動ルール」と呼ばれる。あるエージェントが行動ルールに基づいて決定された行動をとると、自身の状態が変化したり環境に影響を与えたりする。するとそれらの状態変化を観測した他のエージェントの行動が変化したり、影響を受けた環境の状態が変化したりする。すなわち、各々のエージェントは他のエージェントや環境と相互作用を持つと言える。

エージェントベースモデリングの対象となる現象においては一般的に、エージェントは他の全てのエージェントと相互作用することではなく、自身と隣接関係や接続関係などにある一部のエージェントとしか相互作用しない。同様に、環境との相互作用についても局所的なものであることが多い。そのような局所的な相互作用しか持たない場合でも、多数のエージェントの行動の結果が大域的な現象を創発することがあり、そのような現象の理解や予測にこそエージェントベースモデリングやマルチエージェントシミュレーションが有効である。さらに、大域的な(マクロな)現象はエージェントの行動ルールや状態、環境などのマイクロな現象にフィードバックを及ぼし、マイクロ・マクロループ[35][36]を形成する。

ここまでに述べたエージェントベースモデルの特徴を概念図としてまとめたものを図1-1として示す。

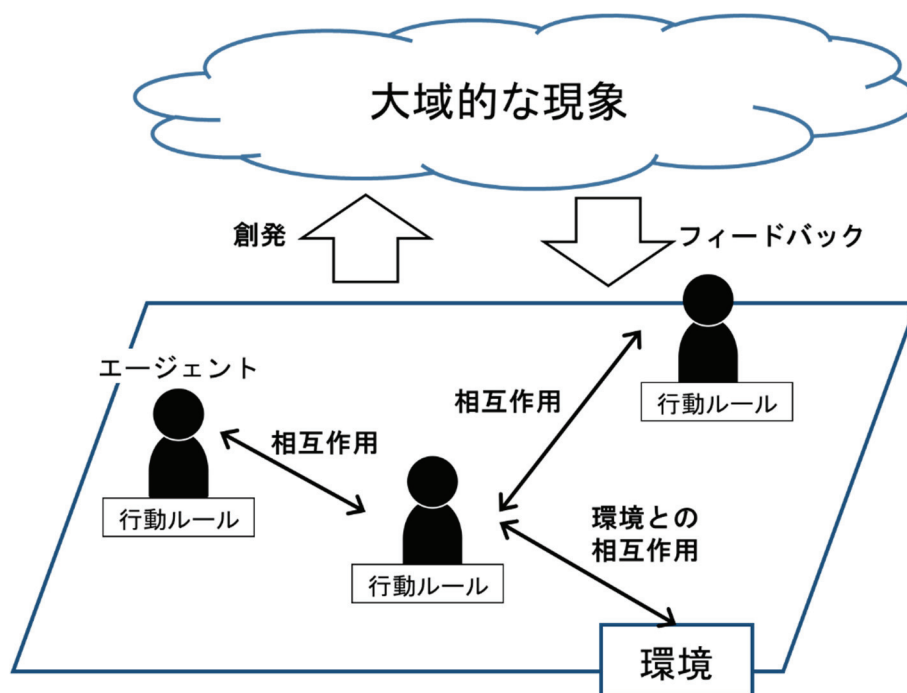


図 1-1 エージェントベースモデルの概念図

ここまで述べたことをまとめると、エージェントベースモデルは下記のような特徴を持つ。

- エージェントが局所的な相互作用を持ちながら行動する
- 局所的な相互作用の集積が大域的な現象を生み出す
- 大域的な現象は、個々のエージェントの行動ルールから直接は導き出せない
- 大域的な現象が個々のエージェントや環境にフィードバックを及ぼし、マイクロ・マクロループを形成する

これらは複雑系と呼ばれる現象[39]を特徴づけるものでもあり、エージェントベースモデリングのようなボトムアップな方法でないと記述することができない。これとは逆にトップダウンにシステムを記述する例の一つとしてはシステムダイナミクス[37]があり、システムを複数の変数間の因果関係や時間遅れを伴うフィードバックモデルとして記述して現象を理解しようとする[38]。しかし、システムダイナミクスのような記述は、変数群の因果関係を陽に記述できる必要があり、複雑系のように大域的な振る舞いを個々の構成要素の振る舞いに還元できない現象を記述するには適していない。これに対して、エージェントベースモデリングではエージェントの行動ルールや相互作用を記述すればよく、複雑系の理解や予測を目的とした記述に適している。

最も代表的な複雑系としては人間社会が挙げられ、エージェントベースモデリングやマルチエージェントシミュレーションは、社会モデリングや社会シミュレーションに適用されることが多い。応用例の多いものとしては、交通などが挙げられる。個々の自動車（ドライバー）の行動ルールがモデルとして記述され、車間距離やスピードを通じた相互作用を持つ。この場合の大域的な現象としては交通渋滞などがあり、個々の自動車の振る舞いからは説明できない複雑な現象である。エージェントベースモデリングにより自動車をエージェントとしたシミュレータを構築し、渋滞の予測や道路網の設計などに応用する研究が数多く行われている（例えば[42]）。交通の他にも、避難・物流・感染症対策・組織運営など様々な社会現象を対象としてエージェントベースモデリングを用いた研究や応用が行われている。これらの分野においてエージェントベースモデリングを行う場合のエージェントや相互作用の例を表 1-1 に示す。このような複雑な現象を記述し、理解したり予測したりすることがエージェントベースモデリングの目的とするところである。

表 1-1 代表的な応用分野におけるエージェントと相互作用

分野	エージェント	相互作用
交通	自動車など	車間距離による加減速など
避難	避難者など	歩行干渉, 情報伝達など
物流	事業者, 倉庫など	注文, 資材の受け渡しなど
感染症対策	感染者, 乗り物など	感染など
組織運営	組織のメンバーなど	情報伝達など

1.3. 社会応用における課題

エージェントベースモデリングは複雑な現象をボトムアップ的に記述・理解するための優れた方法であり、社会現象や生物現象など、様々な現象の理解や予測に活用されている。特に近年の計算機の発達により、規模の大きい計算または短時間での計算が可能となり、社会における実問題への応用が現実的となってきている。実際に、交通や避難など一部の分野では実社会での応用も実現している[43][44]。しかし、社会応用が実現している範囲はまだまだ広いとは言えず、事例も決して多くはない。今後はより幅広い分野での社会応用や様々なケースへの適用が期待されている。また、特定の地域・組織・環境下における現象のモデリングやシミュレーションなど、具体化・詳細化されたニーズも多くある。例えば、特定の地域や組織における人々の行動特性や慣習などを考慮したモデリングや、現実にある特定の建物の利用・運用に関するシミュレーションなどが相当する。実社会の問題を解決したり、実際の社会システムのデザインに応用したりするためには、そのような個別具体の特性や事情を組み込んだモデルが必要となる[40]。さらに、1.1節で触れたように、今後は社会システムのデザインに「超スマートシティ (Society5.0)」に代表されるような、自律分散型のテクノロジーを埋め込む方向性が求められており、エージェントベースモデリングの応用もそれらに対応できることが望まれる。このような場合は、モデル化の対象として人や組織の他に、IoT や AI などを含む機能システムを加えなければならない、社会学的な知見だけではなく、工学的な知見も必要となってくる。これは、すでに一部で産業応用が進んでおり、今後も応用の拡大とデザイン手法の重要性が高まると考えられる製造業におけるスマート化 (工場 IoT など) を考えた場合も同様である[49][50]。

以上をまとめると、エージェントベースモデリングの有用性を高め、社会応用をさらに拡

大していくためには、モデル化のプロセスに対して下記の①～③の事柄が求められている。

- ① より幅広い分野での応用や様々なケースへの適用が可能となること
- ② 個別具体のモデルを素早く効果的に構築することができること
- ③ 人や組織に加え、IoTなどの機能システムの振る舞いや相互作用を適切な形でモデル化できること

実社会の現象を対象としたエージェントベースモデリングの典型的なプロセスは次の通りである。交通、歩行者、感染症などいくつかの対象については数々のモデルが提案されて議論されており、それらを扱う場合は既存のモデルをベースにして行動モデルを記述することができる。既存のモデルがない場合は、まずはエージェントの行動を規定する要因を抽出する。大勢の人々の行動や意思決定結果のデータを分析して、その中から優位性の高い行動要因や行動パターンを抽出し、各行動要因をエージェントの行動モデルに反映する。設定するパラメータもデータから推定する必要がある。前述の社会応用の拡大に向けて必要なこと（適用範囲の拡大、個別具体のモデリング）を考えた場合、データ分析に基づく従来型のプロセスには次のような課題がある。

- (1) 行動要因の抽出に必要なデータを取得・蓄積することのできない現象については適用できない。
- (2) 個別具体を対象にする場合には、データが十分に取得できないことが考えられる。
- (3) データから推定された行動要因が妥当であるかどうかについて確認できない。
- (4) 抽出された行動要因が十分であるかどうかを確認できない。見落としていた行動要因やデータには現れない行動要因があるかもしれない。

また、データ分析に基づくプロセスに限らず、社会モデリングのプロセスに一般的な課題として次のようなものがある。

- (5) 社会現象は複雑で分野横断的であり、その理解や記述には様々な角度や立場からの視点が必要である。

特に前述の③の要求を考慮すると、(5)の課題は社会的な知見や視点に加え、工学的な視点なども含める必要があることを示しており、分野を横断した専門性のもとからのモデリングが必要となる。

本研究では上記に挙げたような諸課題に対応する、新しい形のエージェントベースモデリングのためのモデリングプロセスを提案する。

1.4. 本研究のアプローチ

前節で挙げたモデリングプロセスに求められる事項①～③や課題(1)～(5)に対して、本研究ではエージェントベースモデリングに関する専門性を持たないが、着目する現象に関する知見を持つ人々との協働による対話型のモデリングプロセスを提案する。このような人々には、着目する現象に関する当事者や実務家、該当分野の研究者などが含まれるが、以降ではそれらをまとめて「ステークホルダ」と称する。これに対して、エージェントベースアプローチによるモデリングと社会シミュレーションの構築を企図し、研究活動の中心を担う人物はエージェントベースモデリングや社会シミュレーションに関する専門性を有していることが一般的であるため、以降では「専門家」と記載する。図 1-2 に、本研究が提案する協働型かつ対話型のモデリングプロセスに参加する人物の概要と、本論文における記載についてまとめたものを示す。

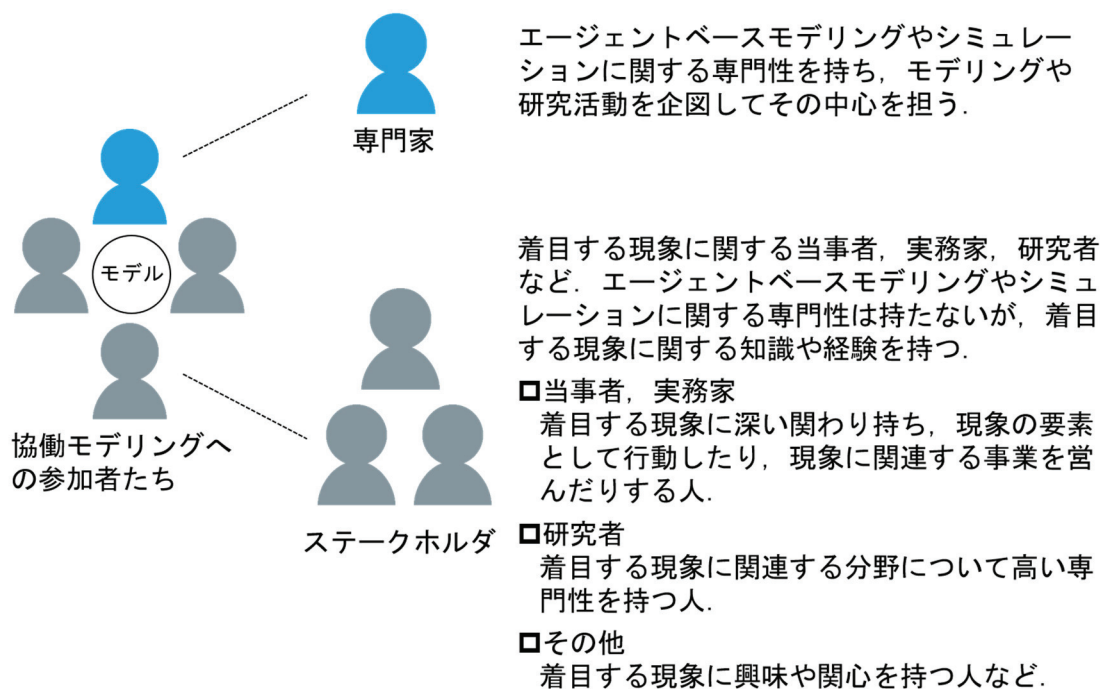


図 1-2 協働モデリングに参加する人物の概要

エージェントに相当する人自身や関連する業務に取り組む実務家など、対象となる社会現象に深く関連する人々は、エージェントの振る舞いに関する詳細な経験値・暗黙知を有している。このようなステークホルダらと協働でモデリングに取り組み、その経験値・暗黙知を抽出してモデルに組み込むことができれば、これまで一般的なモデルが提案されていない現象や十分なデータの取得・蓄積が難しい現象などを対象としたエージェントベースモデリングが可能となり、行動要因やパラメータの抽出漏れや妥当性に対して精度の高いフィードバックを得ることができる。これにより、1.3節で挙げた①、②および(1)～(4)の課題に対応できることが期待できる。また、複雑な社会現象を扱うためには専門分野をまたいだ多角的なアプローチが必要であり、しばしばエージェントベースモデリングや社会シミュレーションに関する専門性を持っていない研究者との連携が必要となる。このような研究者と効果的な協働モデリングを行うことができれば、1.3節で挙げた③および(5)の課題に対して有効であることが期待できる。

上記のような期待に対して、本研究では、着目する現象についての理解・説明・予測を得るためにモデリングを企図する専門家が、エージェントベースモデリングやシミュレーションに関する専門的な知識・手法を持っていないステークホルダらと対話しながら、その暗黙知・経験知・専門知を共有し、協働でモデリングを行うことのできる手法を構築することを目指す。そのような手法の具体的な実装としては、第2章で述べる通り、ゲームとエージェントベースモデルの類似性に着目し、協働でゲームをつくるという行為を通してモデリングを行うプロセスを提案する。

1.5. 本研究の目的

1.3節で述べた社会応用拡大に向けた課題点および1.4節で述べたアプローチに基づいて、本研究の目的として下記に示す3つの目的を設定する。

【目的 I】

専門家^{*1}とステークホルダ^{*2}が対話しながら協働でエージェントベースモデリングを行い、ステークホルダらが持つ暗黙知・経験知・専門知を活用しながらモデルを構築することのできる手法を確立する。これにより、幅広い分野や個別具体の事象、人と機能システムが相互作用を持つ状況などを対象とした場合において有用性・妥当性の高いモデルやシミュレーションを効果的に構築することを可能とし、エージェントベースモデリングの社会応用をより拡大することを目指す。

※1 専門家とはエージェントベースモデリングとそれを用いたシミュレーションを企図し、研究活動の中心を担う人物のことを指す。

※2 ステークホルダとは、着目する現象に関する当事者や実務家、該当分野の研究者などを指し、エージェントベースモデリングや社会シミュレーションに関する専門性を有しない人物を指す。

【目的Ⅱ】

提案する対話型モデリング手法を、様々な分野における実際のエージェントベースモデリングに適用し、マルチエージェントシミュレーションの構築を含む活動を実践することで、提案内容を洗練させるとともに、その有用性を示す。

【目的Ⅲ】

ステークホルダの知見を必要とする分野横断型の研究活動や教育活動において、提案する対話型モデリング手法を活用し、それぞれの活動における問題点を解決し、優れた成果を得る。

1.6. 本論文の構成

本論文の次章以降の構成は次の通りである。まず第2章では、提案する対話型モデリングの手法である GBSP (Game-Based Situation Prototyping) について、コンセプトと具体的な手順を示す。第3章では GBSP を教育活動に適用した事例について紹介する。これにより、具体的に GBSP がどのようなものなのかを示す。続く第4章から第6章では、GBSP を適用して実践したエージェントベースモデリングを伴う研究活動とその成果について紹介する。GBSP を用いることでこれらの研究活動において効果的なモデル構築を実現することができたことを持って、GBSP の有効性が高いことを示す。さらに第7章では、他手法との比較実験により、GBSP の有用性を定量的に評価することを試みた。そして第8章において、本研究における提案や実践結果を改めて総括し、成果と今後の展望についてまとめる。以下に、各章の概要をそれぞれ簡単に示す。

第1章

研究全体の緒言として、研究の背景と目的をまとめた。研究背景としては、複雑化する現代社会において重要なパラダイムである複雑系科学の視点について取り上げ、その基盤的な考え方であるエージェントベースなアプローチによる社会シミュレーションについて取り上げた。そして、社会シミュレーションを構築するための具体的な手法としてのエージェントベースモデリングについて述べ、社会応用を進めていくにあたっての課題点と本研究によるアプローチについて論じた。

第2章

第2章では、本研究で提案する協働モデリング手法である「GBSP (Game-Based Situation Prototyping)」について、そのコンセプトと具体的な手順、既存研究における位置づけなどについてまとめた。GBSP はゲームとエージェントベースモデルの類似性に着目することで交換した対話型の協働モデリング手法であり、参加者が対話しながら着目する現象に関するゲームを構築する活動に取り組み、その結果として参加者の知見や経験を踏まえたエージェントベースモデルを得るという手法である。第2章の後半では、簡単な仮想状況を用いて、GBSP においてどのような手順でゲームが構築されるかの具体例も示した。

第3章

第3章では、GBSP を活用して進めた研究活動のうち、大学などの研究機関におけるチーム形成を扱った研究について述べた。研究者やコーディネータをエージェントとして扱い、コーディネータの情報発信によってどのように学際的な研究チームが形成されるのかをモデル化した。エージェントベースモデルの構築にあたっては GBSP を活用し、コーディネータ業務の担当者や、学際的な研究チームに参加したことのある研究者などのステークホルダと協働でモデリングを行った。その結果、コーディネータの情報発信の方法と研究者チームの成立のしやすさの関係性など、詳細な行動モデルを含めたモデルとシミュレーションを構築することができた。

第4章

第4章では、GBSP を活用して進めた研究活動のうち、企業による立地競争や製品差別化による競争のモデルである「空間的競争モデル」に関する研究について述べた。空間的競争モデルに関する既存研究では、ゲーム的状况における最適化問題を解析的に解くというア

アプローチにおいて様々な研究が行われてきたが、本研究では既存研究における解析的なモデルをエージェントベースなモデルへと自然な形で拡張することに成功した。これにより、従来のアプローチでは扱うことのできなかつた複雑な状況（例えば、消費者の分布が一様ではなくランダムになっている場合など）における空間的競争の帰結を計算することが可能となった。これは、GBSPを用いて関連分野の研究者とエージェントベースモデリングの専門家が協働でモデリングを行うことで、はじめて実現することのできた成果であり、GBSPの有効性を示す事例の一つである。

第5章

第5章では、電力ネットワークにおける需要と供給の制御に関するエージェントベースなアプローチの研究について論じた。供給量が不安定な自然エネルギー利用において有用であると考えられているリアルタイムプライシング（電力価格を変動させることで電力需要を制御する手法）を分散型のアルゴリズムで実行する方法を提案し、その定式化に取り組んだ。そして、提案手法を評価するためのマルチエージェントシミュレーションを構築した。この研究は、提案内容そのものがエージェントベースなアプローチによる工学的なものであり、エージェントベースモデリングについて専門性を持つ研究者らが取り組んだものであるが、提案手法の最初期におけるディスカッションや構想の整理、研究に参加する学生らとのコミュニケーションなどにおいてGBSPを活用した。そのような議論や整理にGBSPを活用することで、提案手法の構築をスムーズに行うことができた。

第6章

第6章では、教育活動においてGBSPを導入し、活用した事例について述べた。モデリングやシミュレーションに関する専門性を持たない参加者との協働モデリングを可能とするGBSPは、初学者同士や初学者と教員との間での対話にも有用である。GBSPを導入することで初学者を対象とした複雑系科学に関する効果的な入門プログラムを構築することができた。この教育プログラムへの参加者は複雑系科学やプログラミングに関する基礎知識を持たない初学者ばかりであったが、GBSPによる協働モデリングを実践することで、限られた時間の中で初学者が主体的にマルチエージェントシミュレーションの構築に取り組むことができ、効果的な学びの機会を創出することができた。これも、GBSPの有効性を示す事例の一つとして取り上げる。

第7章

第3章から第6章では、GBSPを研究や教育などの場において実践し、その成果をもって有効性を定性的に示したが、第7章では実験による有効性の定量的評価を試みた。協働モデリングにおけるGBSPの重要な機能である「状況の共有」に焦点を当て、その有効性を評価するために、既存の状況共有手法との比較対照実験を行った。「状況の共有」に対する有効性を定量的に評価する一般的な既存手法は存在しないため、本研究ではそのような評価を行う方法と指標を開発することも含めて取り組んだ。その結果、状況の共有を評価する指標として「共有度」や「共通認識度」などの独自指標を定義するとともに、形式概念分析を用いた質的な評価手法を構築することができた。また、構築した評価手法を適用することで、GBSPが状況共有に関する特定の側面で高い有効性を持つことを示すことができた。

第8章

第8章では、第1章から第7章の内容を総括し、本研究の取り組み内容とその成果についてまとめた。さらに、残された課題としてGBSPの成果物からシミュレーションを構築するプロセスの一般化について触れた。また、本研究の取り組みの延長として得られると期待できるステークホルダとの協働による社会デザインの方法について述べ、その社会的な意義が大きいことを確認した。

第2章 対話型モデリング手法「GBSP」の提案

2.1. アプローチ

本研究は1.5節の【目的I】において示したように、本研究では専門家とステークホルダが協働でエージェントベースモデリングに取り組み、有用性の高いモデルとシミュレーションの構築を可能とする手法を提案することを目的としている。そのためのアプローチとして、ゲームとエージェントベースモデルの類似性に着目し、ゲームをつくるという行為を通してエージェントベースモデリングを行うという方法について考える。次節で説明するように、ゲームとエージェントモデルの構造は良く似ており、ゲームをデザインすることを通してエージェントモデルを構築することが可能である。そのようなことが可能になると、着目する現象に関するゲームを協働でつくるということを、モデリングに関する専門性を持たないステークホルダが、モデリングという専門的な事柄を意識することなく、自然な形で協働モデリングに参加することができる。なぜなら、ゲームは一般的に良く知られた概念であり、一定のプロトコルを与えることで誰でもそのデザインに参加することができる。一方で、エージェントモデルについては専門的な概念であると捉えられており、専門性を持たない人から見たときに、そのデザインに参加するハードルは高い。しかし、対象となる現象がゲームとして構築された場合、ゲームとエージェントモデルの類似性から、専門家がそのゲームをエージェントモデルやマルチエージェントシミュレーションに変換することは容易である。本研究ではこのような性質を利用し、ゲームをつくることを通してエージェントモデルを構築するという新しい形の協働モデリングの手法を提案する。提案する手法は「Game-Based Situation Prototyping: ゲームに基づく状況プロトタイピング」と名付け、独自の手法として発表している[45][46][47]。

2.2. ゲームとエージェントモデル

ゲームは、プレイヤーが従うべきルール、プレイヤー間またはプレイヤーとゲームシステム間の相互作用、そしてプレイヤーやゲームシステムが保持する様々なパラメータ（リソース）の移動によって特徴付けられる[48]。一方、エージェントモデルは、エージェントが従う行動ルール、エージェント間またはエージェントと環境間の相互作用、そしてエージェントや環境が保持する様々なパラメータの変化によって特徴付けられる[51]。このように、ゲームとエージェントモデルの構造は良く似ており、ゲームをデザインすることとエージェントベー

モデリングを行うことは本質的に近いプロセスであると考えられる。表 2-1 にゲームとエージェントモデルの類似性についてまとめたものを示す。

この類似性のため、ゲームをエージェントモデルとして記述することは容易である。ゲームでは、プレイヤーが従うべきルールや、特定の状況下におけるプレイヤーの行動の選択肢、各行動を選択した場合の結果の導き方などが決められており、それらはそのままエージェントの行動ルールとして扱うことができる。また、ゲームではスコアをはじめ、プレイヤーやゲームシステムが保持する様々な値（リソース）が定義されており、それがどのように移動したり湧き出したり消滅したりするのかについてのルールやメカニクスが定められている。これらはエージェントモデルにおけるパラメータの移動や変化に対応する。

表 2-1 ゲームとエージェントモデルの類似性

ゲーム	エージェントモデル
プレイヤーはゲームのルールに従って行動する	エージェントは与えられたルールに従って行動する
プレイヤーは状況に応じて意思決定する	エージェントは人や組織など、意思決定の主体を表している
プレイヤー同士が相互作用する	エージェント同士が相互作用する
プレイヤーとゲームシステムが相互作用する	エージェントと環境が相互作用する
プレイヤーやゲームシステムが持つリソース（パラメータ）が変化する	エージェントや環境が持つパラメータが変化する

2.3. Game-Based Situation Prototyping

本研究では、前節で示したようなゲームとエージェントモデルの類似性に着目し、ゲームデザインのプロセスを応用することで、ステークホルダらが協働でエージェントベースモデルやマルチエージェントシミュレーションを構築するための手法として **Game-Based Situation Prototyping (GBSP)** を提案する。GBSP はゲーム設計におけるペーパープロトタイピングの手法[52]に基づいており、図 2-1 に示すような紙やスチレンボードなどの

簡易材料、トークン、カード、チップなどの各種プロトタイピングツールを用いて、エージェント、環境、相互作用、パラメータの移動などを机上に素早くプロトタイプとして可視化する。



図 2-1 GBSP に用いるプロトタイピングツール

参加者らはこのように展開されたプロトタイプに対して、リアルタイムに変更を加えながら議論を進める。また、ゲームをつくるつもりでエージェントの行動ルールや相互作用の手続きを議論し、規定していく。これにより、対象としている状況や現象に関して、参加者の意見を拾いながら、そして共通認識を得ながらエージェントベースモデルの基礎となるゲーム形式のプロトタイプを構築していくことができる。GBSP の基本的な手順は以下①～④の通りであり、対応するプロトタイピングの様子を図 2-2 に示す。

<GBSP の基本的な手順 (基本)>

- ① 着目する現象の舞台となる環境と、登場するエージェントを紙、スチレンボード、ポーントークン、チップトークンなどをつかって机上に配置する
- ② エージェントや環境が持つパラメータを、チップトークンやブロックトークンなどを使って配置し、相互作用やパラメータ変化についてカード等を用いて記述する
- ③ ボードゲームを想定して、行動や処理の順序、意思決定が必要な箇所とタイミングなどを確認し、ルール化・手順化する
- ④ 組み立てたルールと手順に従って簡易的なゲームプレイを行い、細部を確認・修正していく

上記に記載したのは GBSP の最も基本的な流れである。より詳細な手順や方法については、2.5 節において具体的な事例と合わせて示す。

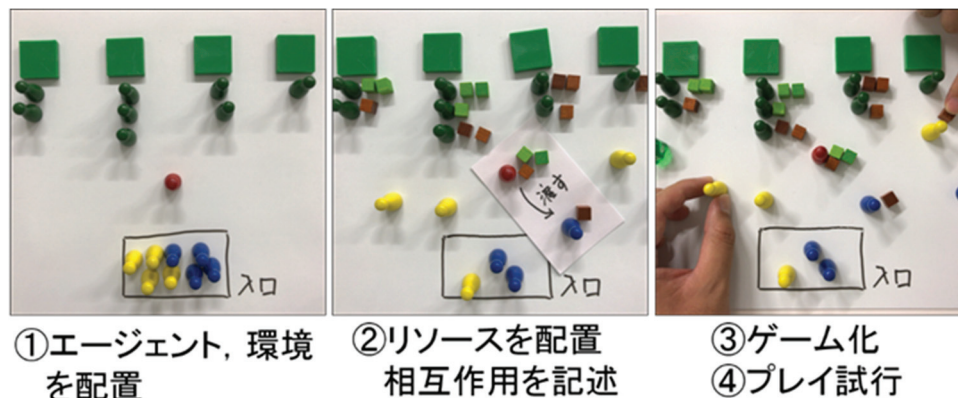


図 2-2 GBSP の基本的な手順

2.4. 既存手法との比較

ゲームを利用したモデリングの手法としては、これまでに参加型のゲーミングシミュレーションが提案、実践されており、例えば参考文献[53][54][55]に挙げるような先行研究がある。

本研究で提案する GBSP ではステークホルダがゲームの作り手としてモデリングに参加するのに対して、ゲーミングシミュレーションの場合はステークホルダがゲームのプレイヤーとして参加する。そのため、ゲーミングシミュレーションにおいてはステークホルダが参加する前の段階で、モデルの骨子に相当するゲームのルールやメカニクスがすでに構築されている状態となる。このようなモデリングにおいてステークホルダの知見や行動が反映されるのは、行動を決める際の確率的重みや相互作用の強さ（ゲイン）など、パラメータとして記述される部分だけとなる。すなわち、ゲーミングシミュレーションはステークホルダと協働でモデリングを行っているとは言えず、その暗黙知や経験値の活用は限定的である。

一方、GBSP ではステークホルダがゲームの作り手として参加し、ルールやメカニクスの構築も協働で行う。これは目的設定や課題抽出の段階から協働で取り組むことに相当し、モデルの骨子からパラメータまで、かなり広範にステークホルダの知見を取り入れてモデリングを行うことに相当する。状況を素早く可視化し、議論しながら随時変更を加えていくことのできるペーパープロトタイピングの手法を導入することで、はじめてそのようなプロセスが可能となる。参加型ゲーミングシミュレーションと GBSP の比較をした結果を表 2-2 にまとめる。

表 2-2 ゲーミングシミュレーションと GBSP の比較

ゲーミングシミュレーション	Game-Based Situation Prototyping
ゲームのロールプレイ性に着目	ゲームの手続き的な側面に着目
非専門家はゲームプレイヤーとして参加	非専門家はゲームの作り手として参加
非専門家が参加する前の段階で、ある程度のルールやメカニクスが構築されている	非専門家らと協働でルールやメカニクスを構築していく

また、ステークホルダと対話しながらモデリングを進めるという観点では、類似の研究としてグループ・モデル・ビルディング[56][57]やコンパニオンモデリング[58][59]などの提案がある。グループ・モデル・ビルディングは参加者となるステークホルダの間のディスカッションをベースにしてモデルを構築する手法であるが、成果物はエージェントベースモデルとなるわけではなく、ディスカッションや対話の進め方についても特定のプロトコルが定まっていない。ステークホルダと協働でモデルを構築するという概念は GBSP と共通しており、GBSP はエージェントベースモデルの構築を目的とした場合のグループ・モデル・ビルディングを実践するための具体的な手法の一つと考えることもできる。また、コンパニオンモデリングではステークホルダによるロールプレイングゲームへの参加とインタビューへの回答を繰り返す形でモデル化に必要な情報を集めていく手法である。ゲームを利用しているという点で GBSP と共通するが、コンパニオンモデリングで用いられるロールプレイングゲームは事前に専門家が設計したものであり、ゲーミングシミュレーションと同様に協働性という点が限定的となる。これらの手法に対して、GBSP では対話のプロトコルとしてゲームをつくるプロセスを導入し、エージェントベースモデリングに最適化された形で協働性を実現するという点が特徴となる。

2.5. GBSP の具体例

0 節において GBSP の基本的なプロセスを述べたが、これだけでは実際の GBSP の進行がどのようなものかイメージすることができない。そこで本節では、仮想的に設定した状況（林業と環境を題材としたもの）を GBSP でモデリングする場合の過程について、構築されるゲームの画像を用いて順を追って見ていく。ここで挙げる例は、第3章で扱う教育活動への適用や第7章で扱う評価実験において、GBSP に取り組む学生たちに対する事前の説

明として使用したものである。

まず、下記のような状況があるとする。これはあくまでも GBSP の説明のために用意した仮想的な状況であり、実際の林業や下流域の環境に及ぼす影響、行政からの補助金について事実と異なる部分が多分にあることに注意されたい。また、実際の研究活動や議論の場において、着目する状況が予めこのような形で整理されていることがないことにも注意が必要である。実際の GBSP では、モデル化されていない対象となる現象や状況があり、それを GBSP を通してモデルとして記述していくので、整理された状況の説明は GBSP の結果として得られるものであって前提となるものではない。

<状況設定>

- 林業従事者は植林して木を育て、伐採して木材を売ることによって利益を得る。
- 植林の量は多すぎても少なすぎてもよくない。土砂滑りなどの災害が起きやすくなる。
- 下流域の住民に対する影響（災害、景観、水質など）に配慮しながら林業を行わなければならない。
- 下流域の住民は環境税を払っており、そこからの支出が補助金として林業従事者に入っている。
- 木材を売るときの市場価格は変動する。
- 木を植えるのにも管理するのにも伐採するのにもコストがかかる。林業は儲からないかもしれない。
- 伐採した木は乾燥してから売らなければならないし、あまりにも長期間保存しておくこともできない

上記のような状況がある場合に、ステークホルダとして林業従事者や行政担当者、下流域の住民などが集めて GBSP を行うことを想定する。この場合に、ゲームとしてのモデルがどのように構築されていくかを以下に示す。実際にはステークホルダとの対話に基づいて、試行錯誤しながらゲームが構築されていくが、ここでは対話の部分の記述は省き、アウトプットとしてのゲームのプロトタイプがどのように変化してくかに着目してその様子を追う。

① 環境を設定する

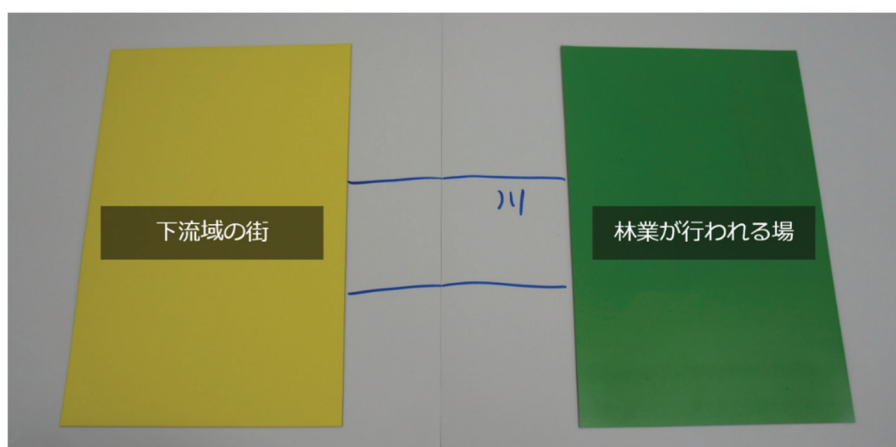


図 2-3 環境の設定

まず、着目する現象の舞台となる環境を、ホワイトボードシートなどを用いて表現する。今回の例では、図 2-3 のように林業が行われる山(上流域)を緑色のシート、下流域の街を黄色いシートで表現している。

② 人物や組織（エージェント）を配置する

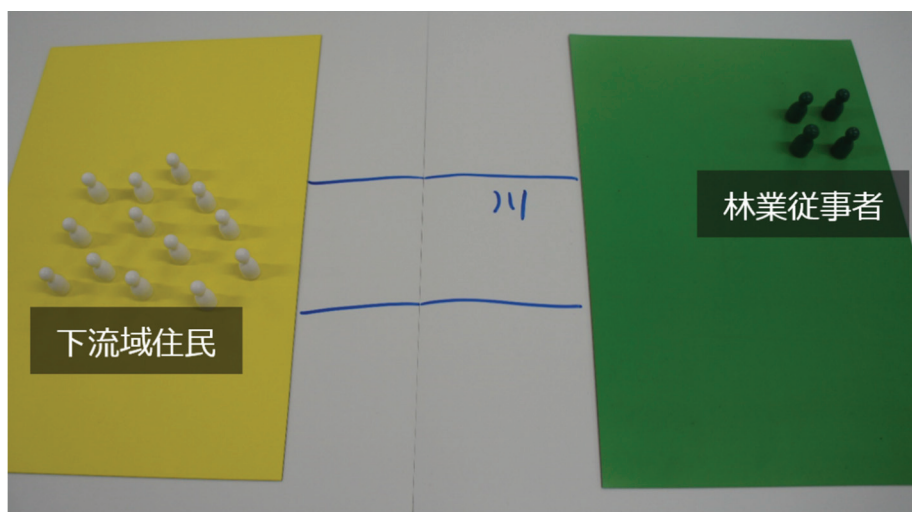


図 2-4 人物や組織の配置

着目する現象に関連する人物や組織などをエージェントとして考え、ゲームの中に配置する。

今回の例では図 2-4 のようにポイントトークンをつかって、林業従事者と下流域住民を表現しており、トークンの色を変えることで属性が異なることを表している。1つのポイント

一トークンが1人の人間を表現する必要はなく、例えば1万人の人間を1つのポイントトークンで代表して表すと考えることもできる。そのような場合、表現のルールについて参加者の間で認識を合わせておく必要がある。

③ 代表的なパラメータを設定する

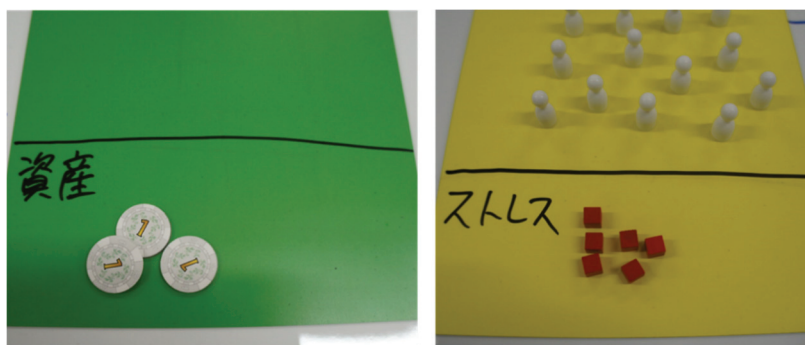


図 2-5 資産パラメータとストレスパラメータの設定

②で設定したエージェントたちが持つ様々なパラメータや状態のうち、その行動に影響を与えるものは何かを考え、ゲームにおけるリソース（スコアを含む様々な状態量）として設定する。設定するパラメータはチップトークンやキューブトークンなどを用いて分かりやすく表示する。

今回の例では図 2-5 のように林業従事者と下流域住民が持つ代表的なパラメータとして資産および環境から与えられるストレスをパラメータとして設定した。このうち、ストレスパラメータについては、環境が悪化することで下流域住民が被る不利益（景観や水環境が悪くなったり、災害被害を受けやすくなったりすること）を抽象的に表すパラメータである。

④ ルールを設定する：木の成長

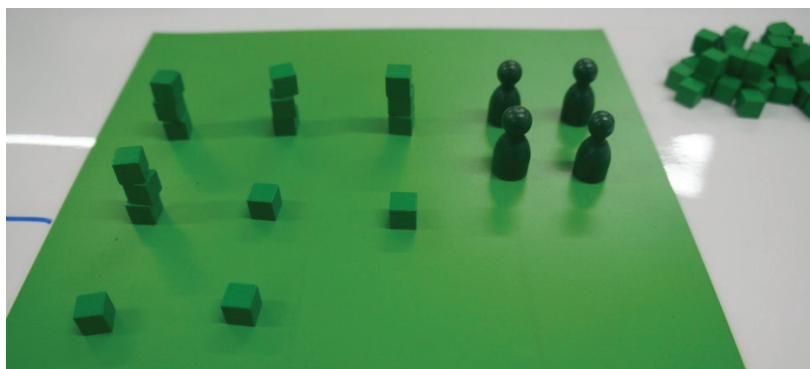


図 2-6 林業で扱う樹木の振る舞いの設定

環境、代表的なエージェント、代表的なパラメータの設定の後は、各エージェントの行動について考えていながら、ゲームのルールを設定していく。

今回の例ではまず、林業で扱う樹木について考え、樹木の成長に関するルールを取り入れた。図 2-6 では緑のキューブトークンが樹木を表し、ゲームの進行によって樹木が成長していくことを、キューブトークンを積み上げていくことで表現している。

⑤ ルールを設定する：木の伐採

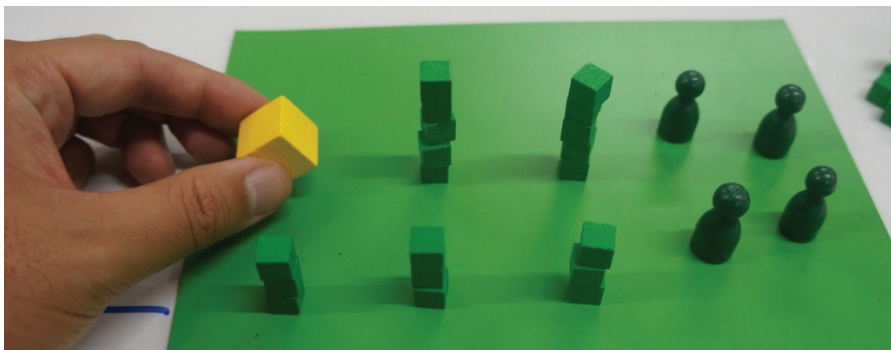


図 2-7 樹木を木材に変換（伐採）

次に、林業従事者の代表的な活動である木の伐採と木材の獲得をルールに取り入れた。樹木が伐採できるようになるためには、一定の成長を待たなければならないとし、その閾値を超えた樹木は伐採行動によって木材に変換することができるとした。この行動は図 2-7 のように緑のキューブトークンで表される木材を、黄色いキューブトークンで表される木材に変換することで表現する。また、伐採行動には一定のコストがかかるとし、資産パラメータを消費しなければならないルールとした。

⑥ ルールを設定する：木材の乾燥と保管

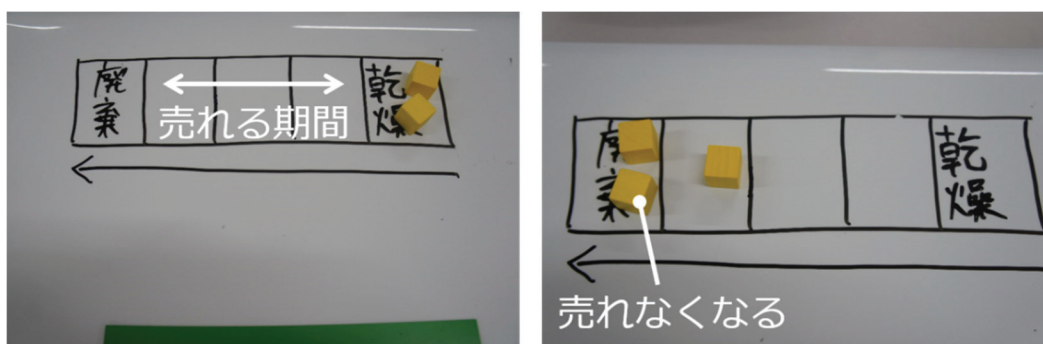


図 2-8 木材保管庫の表現

次に、伐採した木材は乾燥のためにすぐ売れることはできないことと、あまりに長くは保管できないということに対応するルールを設定した。図 2-8 のような木材保管庫に相当する場を設定して伐採直後の木材を保管庫の右端に置き、ターンを経るごとに左のマスへ移動していくことで時間経過を表している。これにより、乾燥に必要な期間と保管可能な期間があることをルールとして取り入れることができる。

⑦ ルールを設定する：価格変動



図 2-9 価格を決定するダイス

不確実性のある現象や、確率的に生じる現象はダイスなどを使ったルールを取り入れることで表現する。今回の例では、変動するパラメータの一つとして木材の市場価格がある。各ターンにダイスを振って出た目を価格とし、価格に応じて林業事業者が木材を売るかどうか意思決定するというルールを設定する。ゲームの場には、図 2-9 のような価格を決定するためのダイスを設置しておく。

⑧ ルールを設定する：樹木の密度と環境への影響

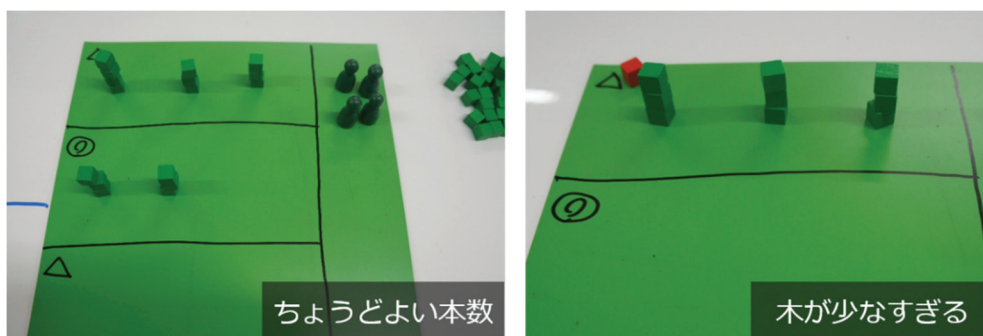


図 2-10 樹木の密度に関するルール

今回の例では、植林した樹木の密度には適切な範囲があり、多すぎても少なすぎても環境に悪い影響を与えてしまうという状況がある。このような状況をルール化するには、ある

パラメータの値に応じて、別のパラメータへの影響が変わってくるようなルールを設定すればよい。図 2-10 では樹木の数が3以下や7以上のときは木の数が適切ではないとしており、その状態にある限り、災害が起こりやすいというルールにしている。

⑨ ルールを設定する：樹木の密度と災害発生への影響

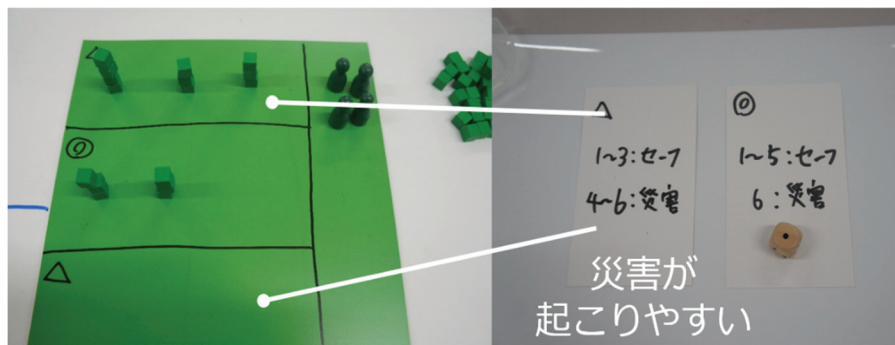


図 2-11 災害発生に関するルール

⑧で触れたように、樹木の数が適切でないときは災害が起こりやすいという状況をルールする。災害が起こるかどうかは不確実性を持つので、ダイスを振って判定とする。このとき、災害が起こりやすい状態すなわち樹木の数が適切でない状態の場合は災害が起こりやすくなるように判定の基準を設定しておく。図 2-11 右側のようなカードにそれぞれの場合の判定基準を記載しておき、樹木の数の適切さによって判定基準として使うカードを変えるようにすればよい。一定期間ごとに災害ダイスを振って災害が起こるかどうかを判定し、災害が起こった場合は下流域住民のストレスパラメータが大きく増えるというルールにしておく。

⑩ ルールを設定する：管理コスト

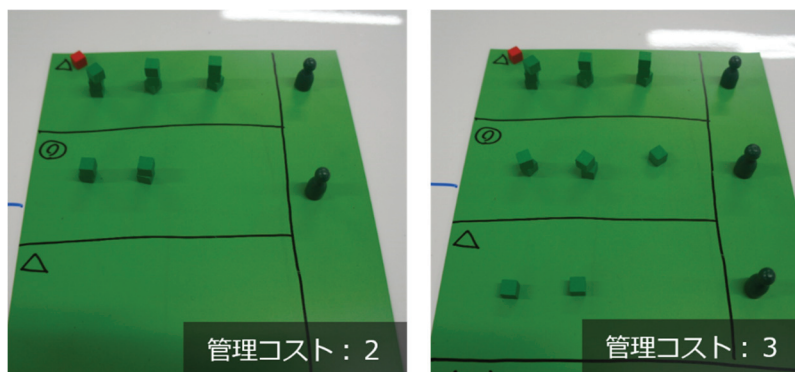


図 2-12 樹木の数と管理コスト

今回の例のような事業やビジネスを含む社会現象に着目する場合、コストをルールに取り込んでおくことは重要である。今回の場合は、樹木の数が多いと、それを管理する人件費が大きくなるので、樹木の数によって毎ターンに支払うコストが高くなるとする。図 2-13 では樹木の数に応じて横(図では右側)に設置するポイントトークンの数を変え、コスト(人件費)が増減することを表す。

⑪ ルールを設定する：環境税

⑩で考えたコストとは逆に、収入や利益についても考える必要がある。今回の例では林業従事者は木材を売却して得た利益の他に、行政からの補助金などを得ることができる。その財源は環境税のような目的税であることが多いので、ここでは税を集めてその一部を補助金として目的別に用いるという部分を簡略化して、環境税が林業従事者の資産に流入するというルールを設定しておく。そのことが簡単に見て取れるように、図 2-13 のようにパラメータの移動を図示しておくとう分かりやすい。

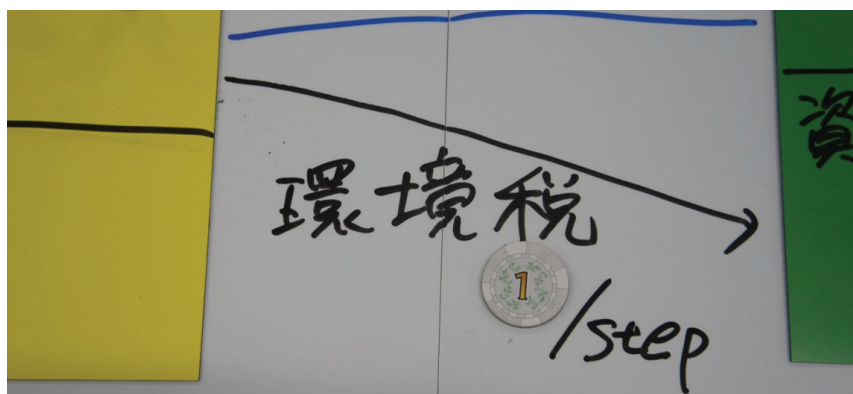


図 2-13 環境税の設定

⑫ テストプレイ

ここまでで、一通りの状況や振る舞いをルール化してゲームを構築することができた。図 2-14 に構築したゲームの全体像を示す。文面ではルールの構築とその表現の過程だけを示しているが、実際の GBSP では、林業従事者や下流域住民、行政担当者などのステークホルダと対話しながら少しずつ状況に関する知見を引き出し、試行錯誤しながらトークンを配置したりルールを設定したりする。ルールの設定が一通り完了した後は、テストプレイという形で、これまでに構築したゲームを動かしてみる。ゲームを動かすことはモデリングの結果を総合的に見ることに相当し、改めてステークホルダらからの意見を収集したり、見落としていた選択肢や状況を発見できたりする。

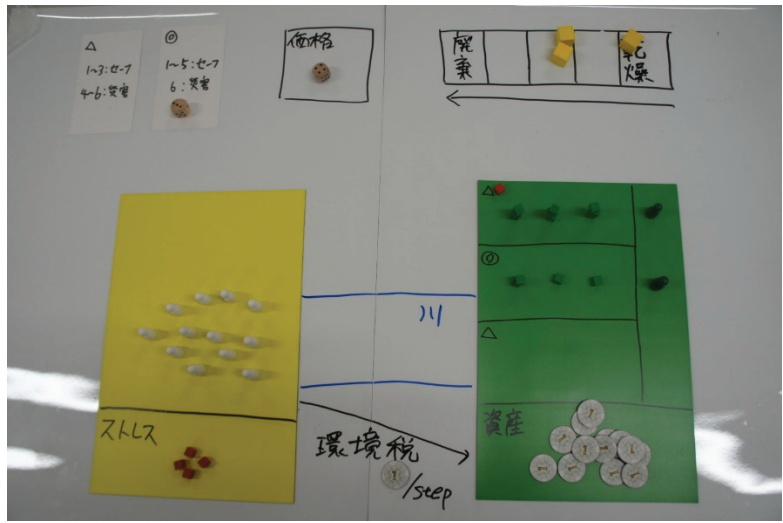


図 2-14 構築したゲームの全体像

⑬ 改善のプロセス

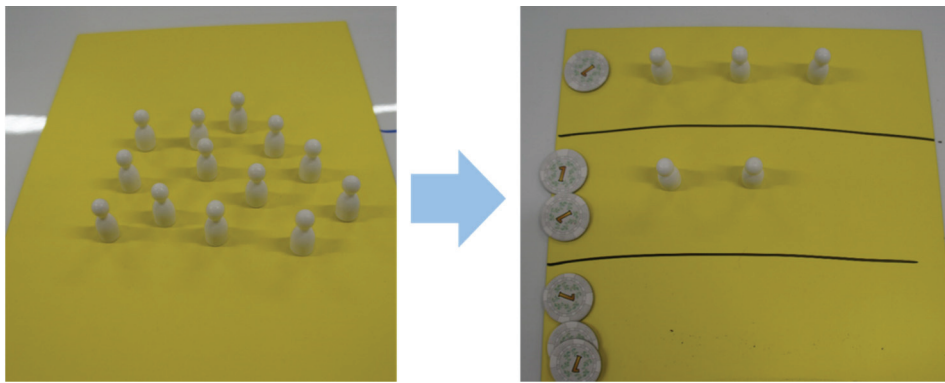


図 2-15 テストプレイの結果を反映

テストプレイを実行すると、その時点までに構築したゲームが、実際の現象のモデルとして妥当でない部分があることにステークホルダが気づくことが頻繁にある。また、見落とししていた状況、条件、選択肢などを洗い出すこともでき、ゲームとしてのモデルをより妥当性の高いものへと改善することができる。

例えば今回の例では、テストプレイで一通りの人物の振る舞いや資産などのパラメータの移動を確認したとき、環境税の金額が住民の数によって変化するという状況がゲームに取り入れられていなかったことに気づくということが考えられる。そこで、図 2-15 のようにその場で住民の数をパラメータとして扱い、その値によって環境税というパラメータが変化することをルールとして定める。このようなテストプレイと改善のサイクルを何回も繰り返すことで、モデルを洗練させていくことができる。

第3章 学際研究チームの形成モデル

大学などの研究機関において学際的な研究チームが構築されるプロセスについてのエージェントベースモデリングにおいて GBSP を実践し、研究者やコーディネータをエージェントとしたモデルを記述した。また、記述したモデルに基づくシミュレーションを行うことでコーディネータの情報発信戦略について考察した。本章ではまず3.1節にてこの研究における GBSP の実践について述べた上で、3.2節以降であらためて研究内容や研究成果についてまとめる。

3.1. GBSP の実践

本章で述べる研究活動においては、研究機関における研究者への支援業務とその結果として実現する学際的な研究チームの形成について着目し、そのプロセスをエージェントベースモデルで記述することを試みた。そこで GBSP を導入し、着目する現象のステークホルダとの対話による協働モデリングを実践した。

3.1.1. 協働モデリングへの参加者

本節で記述する研究においては、ステークホルダとして下記のようなメンバーが GBSP を用いた協働モデリングに参加した。

- 専門家

研究を企図するエージェントベースモデリングの専門家という立場で、本論文の著者が協働モデリングに参加した。

- ステークホルダ（実務家）

今回の研究では、大学などの研究機関において研究活動や研究チームの形成を支援する業務をモデリングの対象とした。そこで、そのような業務の実務を担うステークホルダとして、産学連携を担当する部署の教職員が協働モデリングに参加した。

- ステークホルダ（当事者）

今回の研究では、研究機関において研究活動に取り組む研究者が意思決定者（エージェント）として対象の現象やモデルの中に登場する。そこで、そのような立場の人物に相当するステークホルダとして、他分野や他組織との共同研究に取り組んだ経験を持つ工学系の研究者が協働モデリングに参加した。

3.1.2. GBSP の実践結果

GBSP による協働モデリングを実践した結果, 図 3-1 の右側に示すようなゲームとしてのモデルを得ることができた. 図 3-1 の左側の写真は GBSP を実践している様子である. 協働モデリングに参加したステークホルダ自身が, モデルを構成する要素としての環境やエージェントを表現するために各種トークンを用いて机の上に配置し, 自身の経験に基づいてそれら振る舞いについて説明してゲームのルールを構築していった. そのようにして得られたゲームに沿って, 3.3 節で説明するようなエージェントベースモデルを記述した.

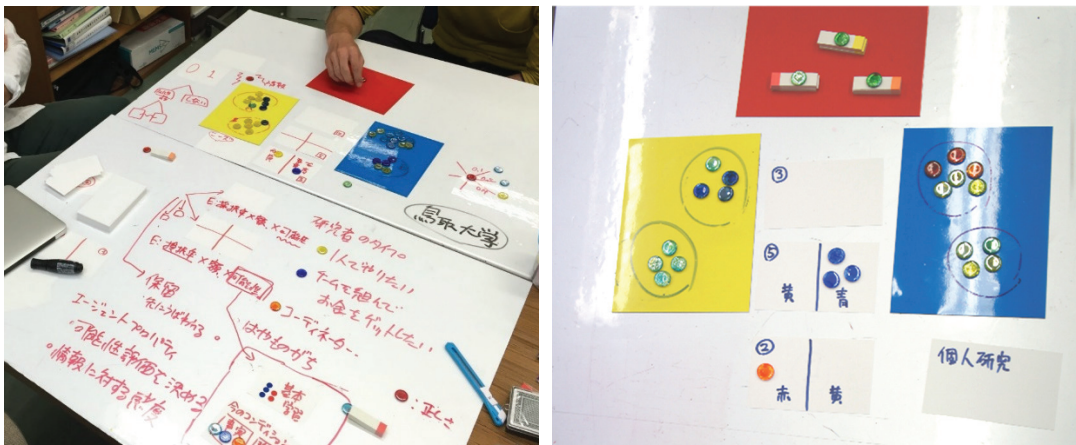


図 3-1 GBSP の様子 (左) と結果として得られたゲーム (右)

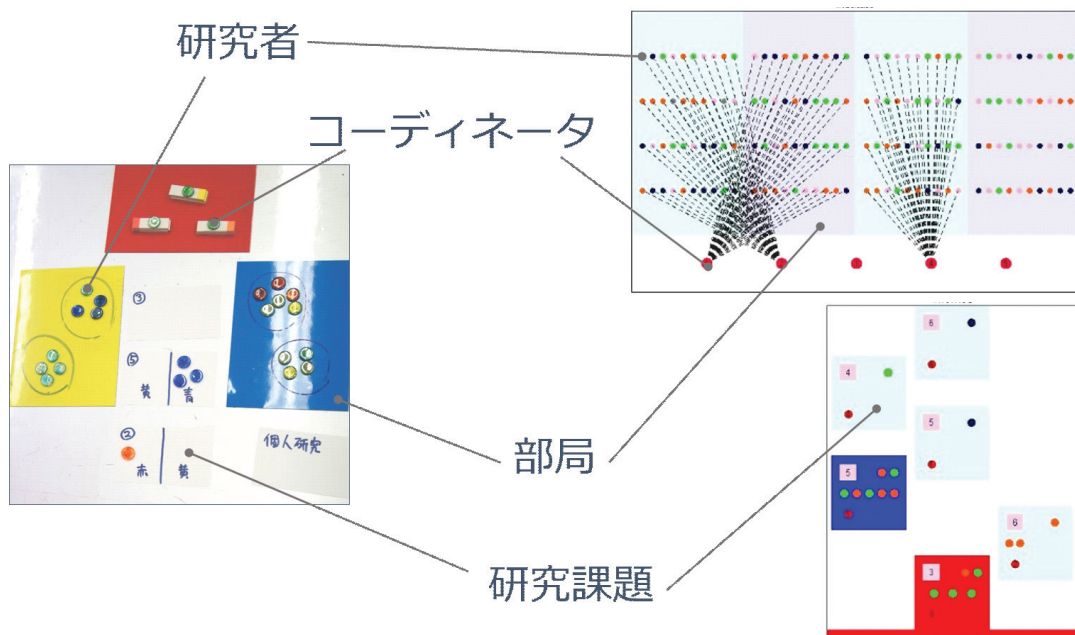


図 3-2 GBSP から得られたゲームとシミュレータとの対応

さらに、そのようにして得られたエージェントベースモデルに基づいて、3.4節で示すようなシミュレータを構築した。図 3-2 で示すように、GBSP の結果として得られたゲームは、シミュレータの中におけるエージェントや環境の表現と対応しており、構築したゲームのルールをそのままエージェントベースモデルおよびシミュレーションに変換することができた。

3.1.3. GBSP の効果

今回の研究に GBSP を導入することで、研究支援を行う実務担当者やモデルにおける主要なエージェントに相当する研究者などのステークホルダと協働でモデリングを行うことができ、ステークホルダの経験や業務知識をエージェントの行動モデルに反映することができた。特に、コーディネータの情報発信の方法によって個々の研究者とコーディネータのコミュニケーションの強度が異なり、その違いが研究者の取得する情報や意思決定に影響を及ぼすことなど、細かい部分ではあるがモデルにとって重要な事実をステークホルダから引き出すことができた。その結果、GBSP がなければ実現することのできなかつた高い妥当性を持つモデル構築を実践することができた。

3.2. 研究の背景・目的

現代社会において、科学研究はその発展に伴う必然として専門分野の細分化がきわめて推し進められている。一方で、宇宙開発・環境問題・平和研究など組織横断・分野横断で取り組まなければならない事柄が多い。つまり、科学と社会の両面の発展から、異分野間での協同を求められることが多くなっている。そのことから、研究者への公的資金の配分条件として学際性が求められる傾向が強まってきた[60]。すなわち、ある課題がそれまで繋がりなかつた多数の研究者の協力を必要としているとき、的確に参加を募ってプロジェクトを成立させるためのシステムが必要とされている。本研究はこの課題に対して、マルチエージェントモデリングやマルチエージェントシミュレーションを用いて研究支援活動を分析することで、組織の規模や構造に応じた適切なコーディネータ配置といった、システムをより効率的に運用するための知見を求めるということを目的としている。

既存の同様の研究支援手法の研究は、ケーススタディ[61]やパネルデータを基にした統計分析[62][63]がほとんどである。ケーススタディは個別のケースについて説明することは可能だが一般化は難しく、パネルデータを使った統計分析ではそれぞれのコーディネータ組織や研究機関といった個々の事例については適用できないと、一長一短があった。これらに

対して本研究ではマルチエージェントモデルを用いる。マルチエージェントモデルを用いた研究活動に関する既存研究としては[65][66][67]などがある。これらはいずれも研究者や技術者などの個人をエージェントとして扱うことで組織やプロジェクト内部の相互作用を記述しようとする試みであり、本研究にも示唆を与えるものである。マルチエージェントモデルやそれを用いたマルチエージェントシミュレーションでは研究者やコーディネータをエージェントとし、振る舞いや配置を変えて相互作用させることで状況にどのような変化が生じるかを分析しやすく、組織構造やコーディネータの行動を変えたときに状況がどう変わるかを分析するのに適している。

3.3. エージェントモデル

本研究では、研究支援コーディネータが研究者に働きかけて研究者チームビルディングを支援するプロセスについて、GBSP を用いて実務担当者の意見を取り入れつつモデル化を行った。本節では構築したモデルについて説明する。

3.3.1. 組織とエージェント

シミュレーションする研究組織は、内部に N_D 個の部局を持ち、その中に N_R 人の研究者エージェントを持つ N_G 個のグループが存在する。加えて N_C 人のコーディネータエージェントが存在するとする。研究者エージェントはそれぞれ専門とする分野、自身の活動状態、所属する部局・グループ、過去のチーム研究の成否の経験値 L といったパラメータを持つ。

コーディネータエージェントは下記のテーマの複数を均等に担当し、そのチーム研究テーマが必要としている専門性を持った研究者集団に対して情報を発信する。情報発信対象となるのは組織全体、部局、グループのいずれかである。これはシミュレーションごとにあらかじめ設定され、すべてのコーディネータについて同じであり、変化することはない。

3.3.2. 研究テーマ

研究テーマについては、各研究者が個人でとりくむ個人研究テーマと、複数の研究者が共同で取り組む研究テーマが存在する。チームで取り組む研究テーマは、研究助成の応募や地域や企業からのニーズなどに相当し、1ステップごとに一定確率で生成される。それぞれ研究チーム成立に必要な研究者の分野、必要な人数 n_{Req} 、募集期間が設定されている。募集期間中に必要なだけの研究者が参加すればチームが成立し、そうでなければ不成立となる。

3.3.3. チームの成立

募集期間中のチーム研究テーマはそれぞれいずれかのコーディネータエージェントが担当する。コーディネータエージェントは担当するテーマそれぞれについて必要な専門性を判断し、設定された情報発信対象が組織全体である時はすべての研究者に、部局かグループであるときは必要な専門性に応じて、その専門性を持つ研究者がより多い部局・グループをより高い確率で選択し、それらに属する研究者に対して情報を発信する。研究者エージェントは、まず、チーム研究テーマのうち未成立のものから自分の専門性に合致するものを探し、以下の式に基づいてテーマごとのスコア S_T を算出する。

$$s_1 = B \quad (1)$$

ここで B は各テーマが持つ利得である。一方で、個々の研究者が持つ経験から来るチーム研究への期待 s_2 は次のようにあわらすことができる。

$$s_2 = I + C_1 \times L \quad (2)$$

ここで、 I は各研究者エージェントが共通で持つ志向を示す値、 L はそれまでに参加したチーム研究テーマの成立・不成立の回数の累積値であり、正負の値をとる。 C_1 は定数である。

$$s_3 = \frac{C_2}{n_{Req} - n_{app}} \quad (3)$$

s_3 は必要なだけの研究者が参加してテーマが成立することへの期待度を示す。定数 C_2 と必要研究者数 n_{Req} 、参加している研究者数 n_{app} から算出される。

$$s_4 = 1 + C_3 \times \frac{S_{CD}}{N_{cast}} \quad (4)$$

s_4 はテーマについて受け取った情報の影響を示し、定数 C_3 とコーディネータエージェントの情報発信力 S_{CD} と発信対象者数 N_{cast} からなる。つまり、発信対象者数 N_{cast} が増えるほど一人あたりの影響度が小さくなることを意味する。これは現実の事象において、たとえば数百人に対してメールを一通送る際の影響度と、ごく少ない人数に絞って足を運んで研究プロジェクトについて説明する際の影響度の違いに相当する。

(1)~(4)で算出した評価値 s_1, s_2, s_3, s_4 から、次の(5)式によってテーマごとのスコア S_T を算出することができる..

$$S_T = s_1 \times s_2 \times s_3 \times s_4$$

$$= B \times (I + C_1 \times L) \times \frac{C_2}{n_{Req} - n_{app}} \times \left(1 + C_3 \times \frac{S_{CD}}{N_{cast}}\right) \quad (5)$$

テーマごとのスコア S_T と、参加せず個人研究を行うスコア S_p それぞれに比例する確率で、いずれかの行動に着手する。

各チーム研究テーマに必要なだけの研究者が揃ったとき、チーム研究テーマが成立したものと見なし、参加する研究者エージェントは参加テーマの成立の累積値である L を一定値加算、一定期間、チーム研究に着手している状態となる。また、自身が参加したにも関わらずそのチーム研究テーマが募集期間中に必要なだけの研究者を集められず不成立に終わったとき、 L を一定値減算する。このとき L は負の値もとりチーム研究プロジェクトが不成立に終わるばかりであれば、研究者は次第に手を出さなくなるといった事象に相当する。

3.4. シミュレータの構築

前節で示したモデルに従ってシミュレーションを実行し、結果を可視化するシミュレータを `artisoc`[87][91]によって構築した。図 3-3はシミュレータ全体のキャプチャ画面である。

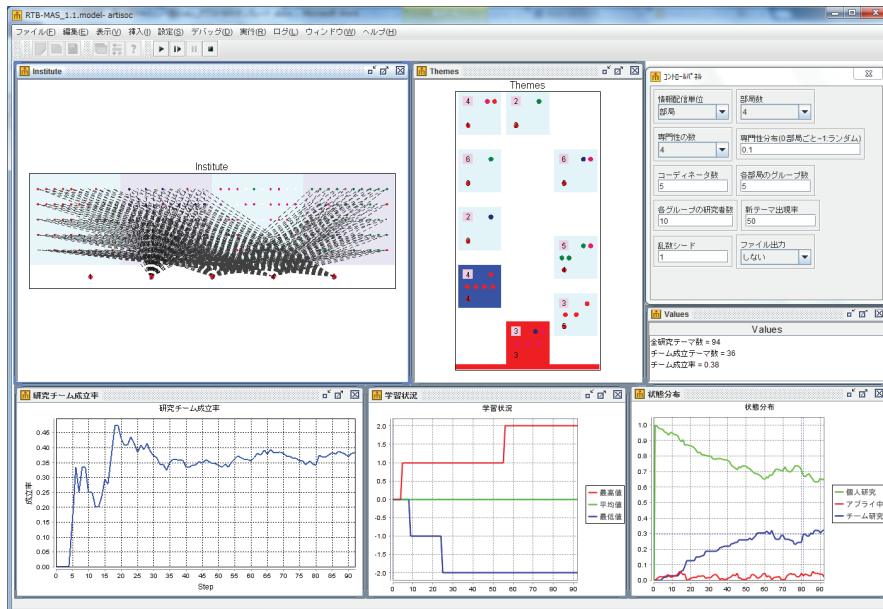


図 3-3 シミュレータの画面

3.4.1. コントロールパネル

図 3-4に示したコントロールパネルからはコーディネータの情報配信単位、研究組織の部局数、部局ごとのグループ数、グループごとの研究者数、研究者の持つ専門性の種類、部

局に対する専門性の分布の偏り，コーディネータ数，チーム研究テーマの出現率，乱数のシード値，ファイル出力の有無を設定できる．

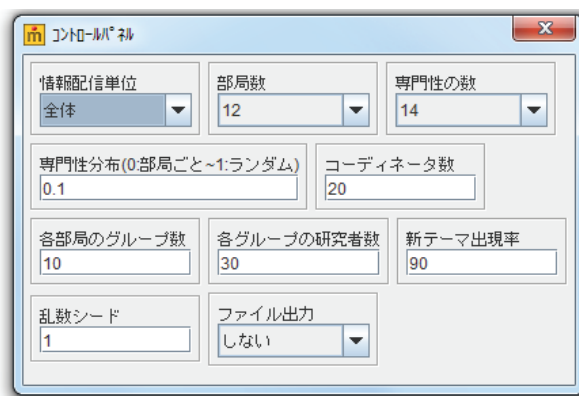


図 3-4 コントロールパネル

3.4.2. 空間(研究組織)

図 3-5 は $N_D=4$, $N_R=10$, $N_G=8$, $N_C=5$, すなわち所属人数 $N = N_D \times N_R \times N_G + N_C = 325$ 名の研究組織の表示である．現実には当てはめれば上部の小円のひとつが一人の研究者エージェントであり，背景の色分けが部局，部局内における横の並びが同一のグループであることを示す．個々の研究者の色は専門性を示す．下部の赤い円がコーディネータエージェントである．

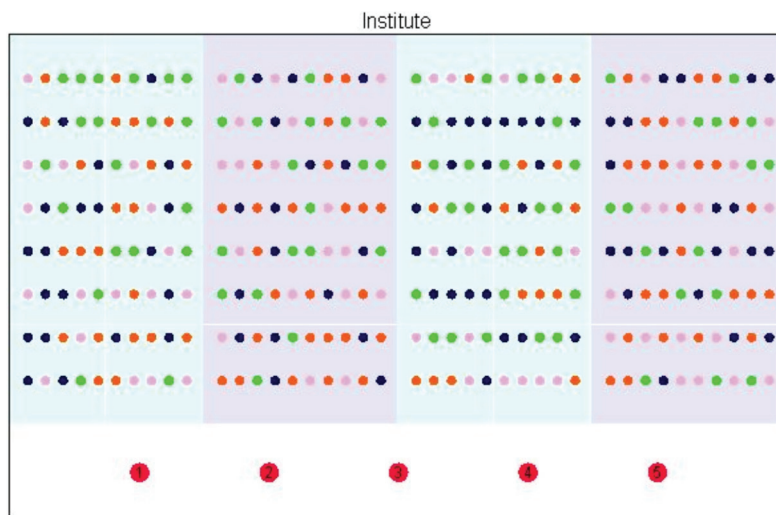


図 3-5 組織空間

コーディネータエージェントが情報を発信すると，コーディネータエージェントと情報発信対象が点線で結ばれるよう表示される．図 3-6，図 3-7，図 3-8 はそれぞれ全体，部局ごと，グループごとを対象にして情報を発信する場合の表示である．

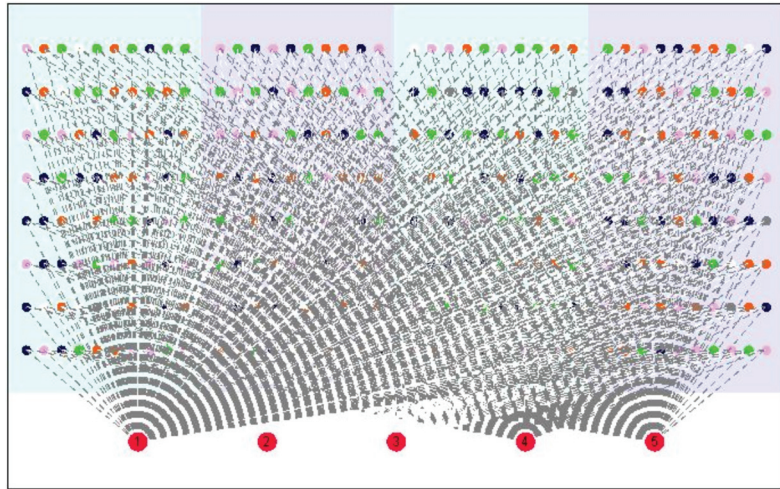


図 3-6 全体へ情報発信する場合

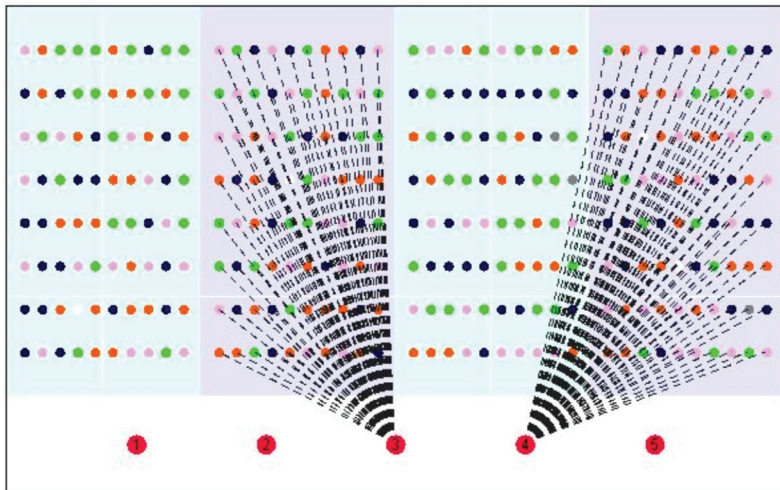


図 3-7 部局へ情報発信する設定の場合

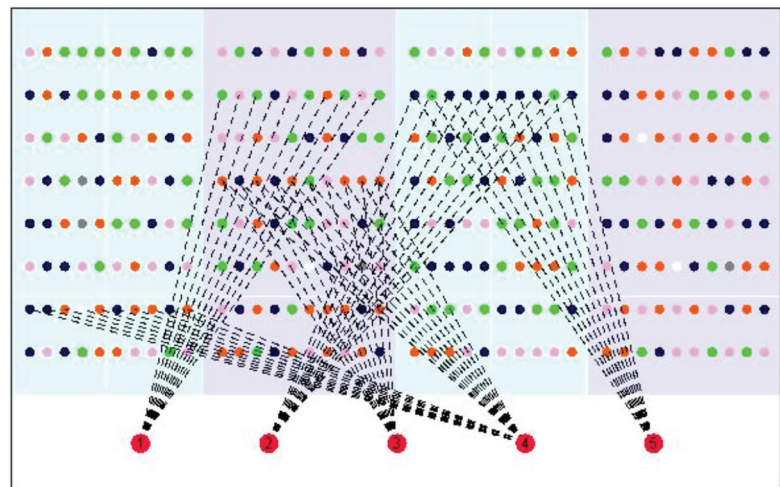


図 3-8 グループへ情報発信する場合

3.4.3. 研究テーマ空間

図 3-9 は発生したチーム研究テーマが募集期間内に成立する、あるいは不成立に終わるまでの推移をグラフィカルに示す空間である。水色の正方形が研究者を募集中のチーム研究テーマであり、左上に表示されているのが必要な人数、右上に表示されている円の色が必要な専門性、下部の円が参加している研究者を示す。

1 ステップごとに一定確率で生成され、生成直後は最上段に存在する。ステップが経過するごとに下に推移し、最下段に到達したとき募集が締め切られ、必要なだけの研究者が参加しなければ不成立となる。

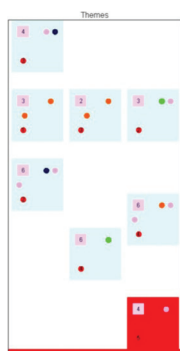


図 3-9 テーマ空間

3.4.4. 時系列グラフ

構築したシミュレータでは以下の 図 3-10, 図 3-11, 図 3-12 のように、シミュレーション開始時からの、全テーマ数に対する成立テーマ数の割合、研究者の経験値の最高値・最低値・平均値、研究者の状態割合の推移をそれぞれ表示するようにし、結果を即座に把握できるようにした。

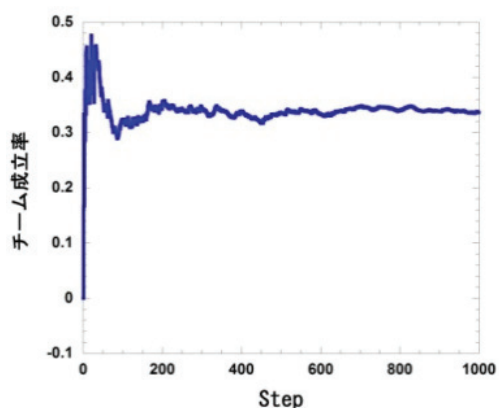


図 3-10 チーム成立率の表示

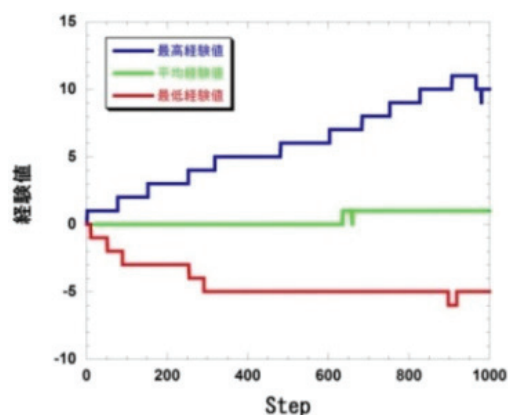


図 3-11 研究者経験値の表示

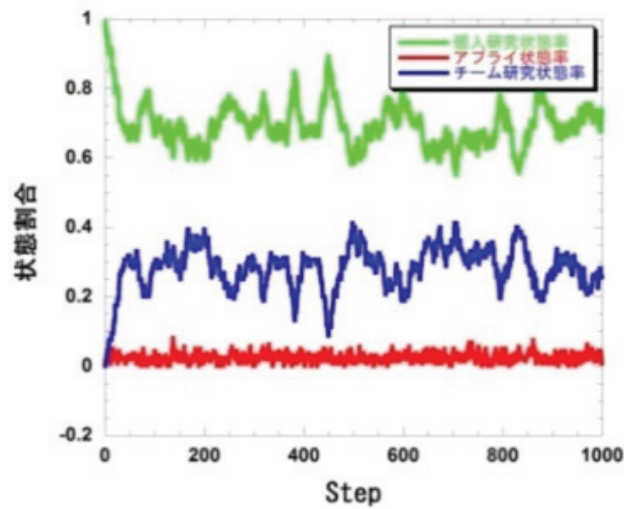


図 3-1 2 状態割合の表示

3.5. シミュレーション結果

構築したシミュレータを用いて、いくつかの条件での実験と比較を行った。まず組織の部局数 4，グループ数 4，研究者数 160，コーディネータ数は 5，専門性の分散は 0.1 で，情報発信の傾向を変更してチーム成立率の変化を観察したところ，図 3-1 3 のようになった。

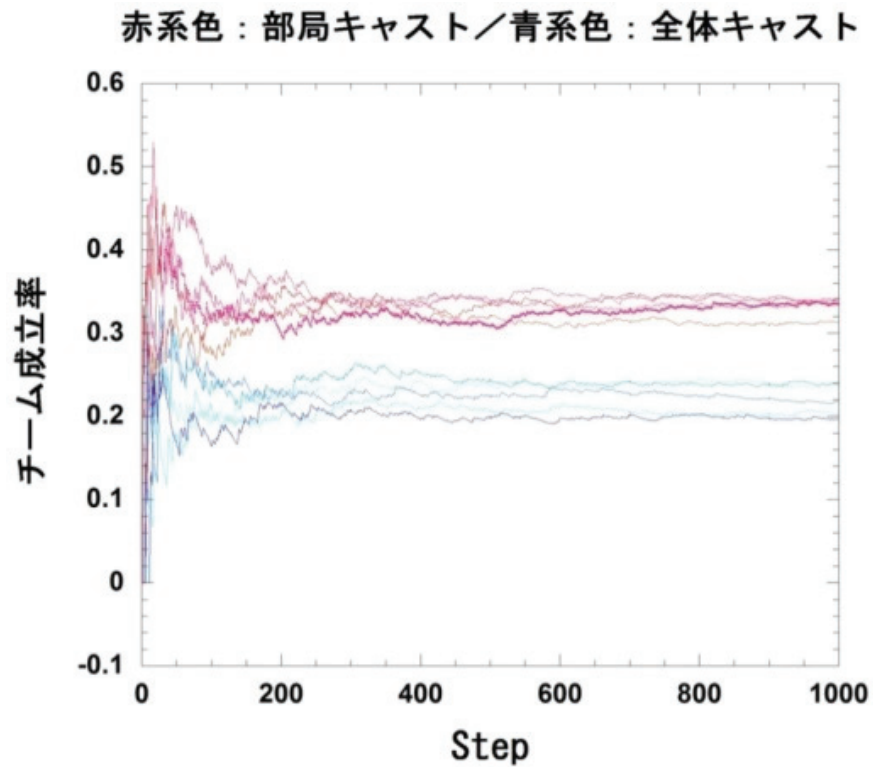


図 3-1 3 情報発信方法によるチーム成立率推移の比較

図 3-1 3 に示された結果から、広く浅く呼びかけることになる全体への発信に比べ、比較して狭く深く呼びかける部局ごとの発信への方が成立率が明らかに良いという差異が見られた。そこで同様の組織の部局数 4、グループ数 4、研究者数 160 で他の諸条件をさまざまに変えてシミュレーションしたところ、1000 ステップ目におけるプロジェクト成立率は表 3-1 のような結果となった。また、全体への情報発信、部局への情報発信、グループへの情報発信のそれぞれの情報発信戦略について、専門性の分散と 1000 ステップ時のチーム成立率（5 回の試行の平均値）の関連は図 3-1 4 のようになった。

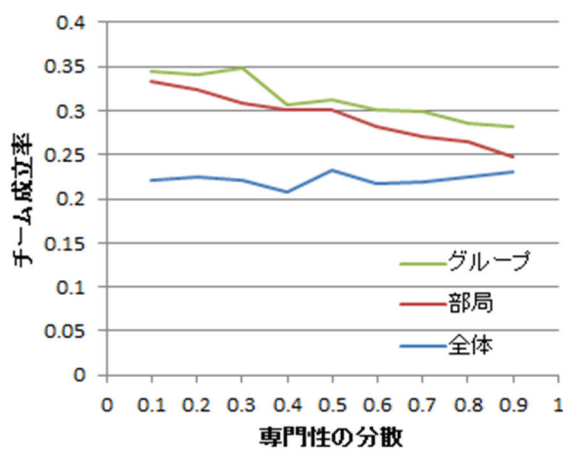


図 3-1 4 各方法と分散におけるチーム成立率

表 3-1 さまざまな条件におけるチーム成立率

	A	B	C	D	E	F	G	H	I
専門性分布	0.1	0.1	0.1	0.9	0.9	0.9	0.1	0.1	0.1
情報発信	全体	部局	グループ	全体	部局	グループ	部局	部局	部局
コーディネータ数	5	5	5	5	5	5	1	5	20
1 回目	0.217	0.334	0.334	0.203	0.257	0.253	0.195	0.334	0.356
2 回目	0.202	0.342	0.358	0.211	0.555	0.304	0.213	0.342	0.412
3 回目	0.240	0.336	0.294	0.263	0.247	0.276	0.215	0.336	0.367
4 回目	0.237	0.312	0.370	0.255	0.208	0.264	0.250	0.312	0.413
5 回目	0.197	0.343	0.364	0.245	0.291	0.269	0.208	0.343	0.357
平均	0.219	0.333	0.344	0.235	0.251	0.273	0.216	0.333	0.381

表 3-1 において、G, H, I とコーディネータ数を増やすことでチーム成立率が向上したのは当然の結果であるが、注目すべきは B, C と E, F の比較、及びそれと A, D そして A, G の比較である。全体への無差別なアプローチはコーディネータ数の増加、組織構造の整備のいずれも効果を表さないことが分かる。

図 3-1 4 から分かるように、分散の度合いによらず全体よりも各部局、部局よりも各グループと、より対象を絞り込んでアプローチする方がチーム成立率は改善された。しかしながら、専門性の分散がごく小さいとき、即ち各部局にそれぞれの専門性を持った研究者がまとまっている時は、部局キャストとグループキャストはごく近い結果となった。逆に専門性の分散が大きく、部局ごとでのまとまりが薄い場合、部局キャストは全体への無差別なキャストと近い結果となった一方で、グループへのキャストは一段高い数値を保持していた。

このことから、部局ごとに特定の専門分野の研究者をまとめる組織構造の整備、および組織構造に合わせた情報発信戦略の選択が十分になされていない場合、それを行うことによって共同プロジェクトへの取り組みには著しい改善が見られると考えられる。

第4章 空間的競争モデル

競合する複数の小売店舗がどこに立地するのかなどを扱う空間的競争の研究に対してエージェントベースモデリングの考え方を導入し、解析的な手法で研究されてきた既存の競争モデルをエージェントベースなモデルへと拡張した。本章ではまず4.1節にてこの研究におけるGBSPの実践について述べた上で、4.2節以降であらためて研究内容や研究成果について述べる。

4.1. GBSPの実践

本章で述べる研究は、異分野の研究者が協力して取り組んだ分野横断型の共同研究である。この研究が対象とする空間的競争は、マーケティング学の分野で扱われてきたトピックであり、マーケティング分野の研究者と、エージェントベースアプローチな社会シミュレーションに取り組む研究者が協力することで既存研究の枠を超えたモデルの拡張を目指している。そこで、異分野の研究者同士が認識の齟齬なくスムーズに対話するための手段としてGBSPによる協働モデリングを実践した。

4.1.1. 協働モデリングへの参加者

本章で記述する研究においては、ステークホルダとして下記のようなメンバーがGBSPを用いた協働モデリングに参加した。

- 専門家

研究を企図するエージェントベースモデリングの専門家という立場で、本論文の著者が協働モデリングに参加した。

- ステークホルダ（研究者）

この研究が対象とする企業や小売店の行動について専門的な知識を有するマーケティング分野の研究者が協働モデリングに参加した。

4.1.2. GBSPの実践結果

この研究では、空間的競争のエージェントベースモデルを構築し洗練させるにあたって、継続的に行った議論の中で何度もGBSPを行った。そのようなGBSPの成果として得られたゲームの例を図4-1に示す。得られたゲームの外観は単純であり、写真ではその活用方法が分かりづらいが、配置したトークンを動かしながら対話し、専門知について内容や言葉

を確認しながら少しずつ共有することで、専門分野の枠を超えて認識の齟齬なく協働でモデリングを行うことができた。その結果として4.5節で示すようなマルチエージェントシミュレーション構築することができた。図 4-2に GBSP で得られたゲームと、構築したシミュレーションの要素の対応を示す。



図 4-1 GBSP の成果物の例

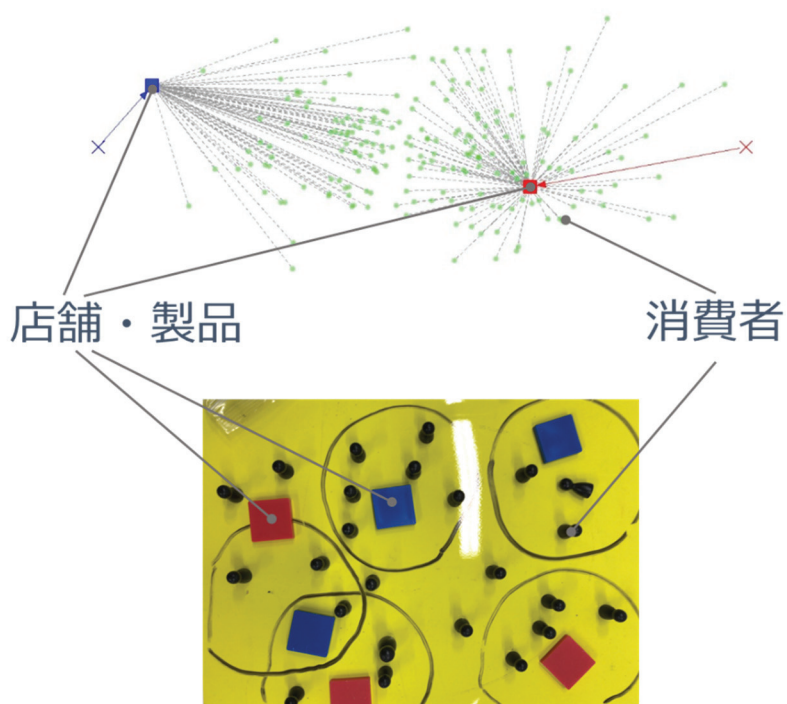


図 4-2 GBSP から得られたゲームとシミュレータとの対応

4.1.3. GBSP の効果

今回の研究において GBSP は、専門分野を横断して共同で研究に取り組む際の翻訳装置としての役割を果たした。本研究の目的は既存の空間的競争のモデルをエージェントベースなものへと拡張することであり、最初にこれまでのマーケティング学の分野における空

間的競争の研究やそこで構築されてきたモデルについて共有するところからはじめる必要があった。分野が異なると、研究やモデルにおいて用いられる言葉や概念も異なり、そのような差が共同研究に取り組む際の認識の齟齬を生み出す。今回の研究では GBSP を用いて言葉や概念を確認しながらモデリングを行うことで、そのような認識の齟齬を起こすことなく、効果的に既存研究の共有や新しいモデルに関する議論を行うことができた。その結果、次節以降に示すように、既存研究を拡張する形で、新しいモデルとシミュレーションを構築することができた。

4.2. 研究の背景・目的

本研究では、マルチエージェントシミュレーションを用いて、ホテルモデルを拡張することに取り組む。ホテルモデルは、2人の売り手の立地競争の帰結を描写したモデルであり、戦略的状況における最適化問題を解析的に解くことによって、メーカーの最小差別化や小売店の隣接を説明するのに引用されている。エージェントベースなアプローチでホテルモデルを再構築することによって、解析的なアプローチでは描写しえない複雑な状況をモデル化することが可能になる。とりわけ、消費者嗜好の時間変化と売り手のリポジショニングの関係や、消費者と小売業者の相互作用プロセスとしての買い物弱者問題などに広く応用可能であると考えられる。

競合する2つの小売店が隣接して立地していたり、同一製品カテゴリーにおいて、類似した製品が市場に氾濫していたりという現象はしばしば観察される。こうした企業の間質化行動は、ホテルモデルによって説明されてきた[68]。ホテルモデルが示唆することには、消費者が一様に分布する線分市場において、同質財を販売する2人の売り手はどちらも中央に立地するというのである。これを最小差別化原理という。しかしながら、同様の線分市場を想定したとしても、ある条件の下では「2人の売り手はそれぞれ線分市場の両極に立地する」という最大差別化原理が成り立つという[69]。たとえば、競い合う売り手がまず立地を選択して、次に価格を設定するという2段階ゲームの状況で、なおかつ消費者の移動コストが2次関数の場合には最大差別化原理が成立する。最小差別化原理と最大差別化原理のいずれが成立するかは各モデルが想定する状況によって異なっている。

空間的競争に関する既存研究の基本的なアプローチは共通している。それは、立地や価格という戦略変数を有する2人のプレイヤーのゲーム的状況について最適化問題を解析的に解くというアプローチである。換言すれば、既存研究は、各々が設定した状況について、ナッ

シュ均衡または部分ゲーム完全均衡を解析的に求め、どのような条件の下で、最小差別化原理または最大差別化原理が成り立つのかに焦点を合わせている。一方、本研究では、マルチエージェントシミュレーションを用いてホテルモデルの拡張を目指す。マルチエージェントシミュレーションとは、複数のエージェント（たとえば消費者や企業）が各々のルールに従って同時進行的に相互作用することによって、どのようなマクロ現象が創発するのかをコンピュータ上の仮想空間でシミュレートする方法である[70]。こうした拡張によって、解析的なアプローチでは解くことのできなかつた複雑な状況をモデル化することが可能になると考えられる[71]。

本章の構成は以下の通りである。まず、ホテルモデルを 2 次元に拡張した Tabuchi の空間的競争モデル[72]を定式化し、次に Tabuchi のモデルをエージェントベースの形にモデル化し、マルチエージェントシミュレーションを構築する。そしていくつかの条件下でのシミュレーション結果を示した上で、今後の展望について触れる。

4.3. 2次元上の空間的競争モデル

本節では Tabuchi による 2 次元の空間的競争モデル[72]について概要を示す。このモデルでは、2 次元の市場空間 (x, y) において、 N 人の消費者が均一に分布しているとする。ただし、 $x = [0, c]$ および $y = [0, 1/c]$ である。 c は自然数である。まず、企業 1 は (x_1, y_1) に企業 2 は (x_2, y_2) にそれぞれ同時に立地する。次に、各々の立地を所与としたうえで、企業 1 と 2 はそれぞれ価格 p_1 と p_2 を設定する。 (x_i^c, y_i^c) に位置する i 番目の消費者は、以下の通り価格と移動コストの和が低い企業

$$\min \{p_1 + td_i^{12}, p_2 + td_i^{22}\} \quad (1)$$

から財を 1 単位だけ購入する。ここで、 d_i^1 および d_i^2 はそれぞれ企業 1 および企業 2 と i 番目の消費との間の距離であり、次のようになる。

$$d_i^1 = \sqrt{(x_i^c - x_1)^2 + (y_i^c - y_1)^2} \quad (2)$$

$$d_i^2 = \sqrt{(x_i^c - x_2)^2 + (y_i^c - y_2)^2} \quad (3)$$

ここまでに述べた消費者、企業が持つ変数と各記号の関係について図としてまとめたものを図 4-3 に示す。

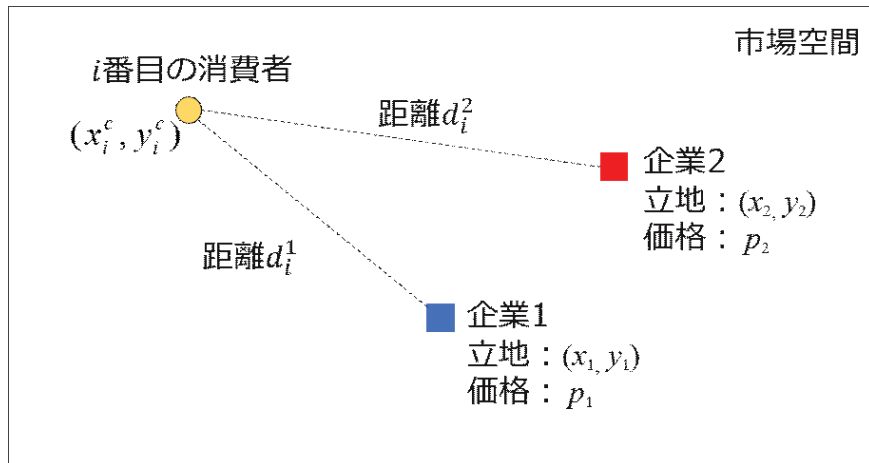


図 4-3 消費者と企業に関する変数と記号

ここで、市場空間のうち企業1から財を購入した消費者が分布する領域の面積を D とすると、 D は次のように計算できる。

$$D = \int \{(x, y) | p_1 + t(x - x_1)^2 + t(y - y_1)^2 \leq p_2 + t(x - x_2)^2 + t(y - y_2)^2\} dx dy \quad (4)$$

このとき、企業1と2の利潤はそれぞれ

$$\pi_1 = p_1 D \quad (5)$$

$$\pi_2 = p_2 (1 - D) \quad (6)$$

となる。

こうした設定のゲームを解析的に解くと、市場空間が細長い長方形になるほど、企業はそれぞれ1つの次元については最大限に製品差別化を行うが、いま1つの次元については製品差別化を行わなくなる。このとき、製品差別化を行う軸は市場空間の長い軸（差別化の余地が大きい軸）であり、短い軸の方では差別化は起こらない。このようにして、Tabuchiのモデルでは製品差別化により価格競争が回避される[72]。

4.4. エージェントベースモデル

前節で示したモデルに基づいて競争する企業の行動をエージェントベースモデルとして記述し、マルチエージェントシミュレーションを構築することを考える。このモデルにおけるエージェントは、競争する企業エージェント（企業1および企業2）、そして2次元市場に分布した N 人の消費者エージェントである。企業エージェントがそれぞれ立地と価格を決めたとき、消費者エージェントは(1)式に従って各企業を評価し、選択した企業から財

を購入する．前節で示した Tabuchi モデルではシェアは (4) 式のように市場空間における面積として算出されるが，エージェントベースモデルではそれぞれの企業を選択した消費者エージェントの数で置き換える．企業 1 から財を購入する消費者の数を数えるために，次のような変数 n_i を考える．

$$n_i = \begin{cases} 1 & \text{if } p_1 + t(x_i^c - x_1)^2 + t(y_i^c - y_1)^2 \leq p_2 + t(x_i^c - x_2)^2 + t(y_i^c - y_2)^2 \\ 0 & \text{if } p_1 + t(x_i^c - x_1)^2 + t(y_i^c - y_1)^2 > p_2 + t(x_i^c - x_2)^2 + t(y_i^c - y_2)^2 \end{cases} \quad (5)$$

上記の n_i は， i 番目の消費者が企業 1 から財を購入する場合は 1，企業 2 から財を購入する場合は 0 の値をとる．このような変数を用いると企業 1 から財を購入する消費者の数 D_1 と，企業が得る利潤 π_1 と π_2 は次のようになる．

$$D_1 = \sum_i n_i \quad (6)$$

$$\pi_1 = p_1 D_1 \quad (7)$$

$$\pi_2 = p_2 (N - D_1) \quad (8)$$

となる．企業エージェントはそれぞれの利潤を最大化するよう戦略的に行動する．企業エージェントの行動は立地選択と価格設定の 2 段階からなり，それぞれがゲーム的状况を構成するため，ここでは 2 段階ゲームの基本的な解法である後ろ向き帰納法[73]をベースにして行動ルールを定める．

まず，企業エージェントの立地を所与としたときの価格設定を考える．企業 1 と企業 2 が取り得る価格の組み合わせの全てとそれぞれにおける利潤を算出すると計算負荷が非常に大きくなってしまうため，今回，提案するシミュレーションでは，企業 1 と企業 2 が交互に価格を更新し，更新による価格変化が一定の範囲に収まったところを均衡状態として扱う．このとき，消費者エージェントから見た留保価格を更新価格の候補とすることで，少ないステップ数で均衡状態周辺に到達するよう工夫する．企業 2 の価格が p_2 のとき， (x_i^c, y_i^c) に位置する i 番目の消費者エージェントの企業 1 の財に対する留保価格 \bar{p}_1^i は次のようになる．

$$\bar{p}_1^i = p_2 + t(x_i^c - x_2)^2 + t(y_i^c - y_2)^2 - t(x_i^c - x_1)^2 - t(y_i^c - y_1)^2 \quad (7)$$

(7) 式によって算出された留保価格 $\{\bar{p}_1^i | i = 0, 1, \dots, N - 1\}$ のそれぞれに対して消費者エージェントによる企業選択と (5) 式による利潤計算を行い，最も高い利潤を与える留保価格

を企業1の更新価格として採用する. 図4-4のように, もし企業1の価格 p_1 が, 留保価格の集合 $\{\bar{p}_i^1 | i = 0, 1, \dots, N-1\}$ に含まれない値に設定された場合, その価格より大きい留保価格のうちで最も小さい価格(このような留保価格を \hat{p}_1 とする)まで p_1 を引き上げて顧客数(企業1を選択する消費者の数) D_1 は変化しない. そのため, 少なくとも価格を p_1 と設定するよりも \hat{p}_1 と設定する方がより大きな利益を得ることが期待できる. 以上の理由から更新価格の候補を留保価格に絞ってよいことが分かり, そのような絞り込みの工夫を取り入れることでシミュレーションにおける計算量を大幅に減らすことができる.

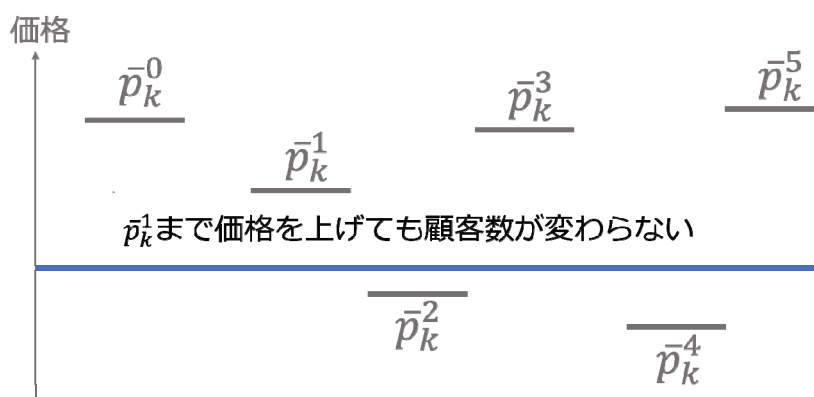


図 4-4 更新価格の候補としての留保価格

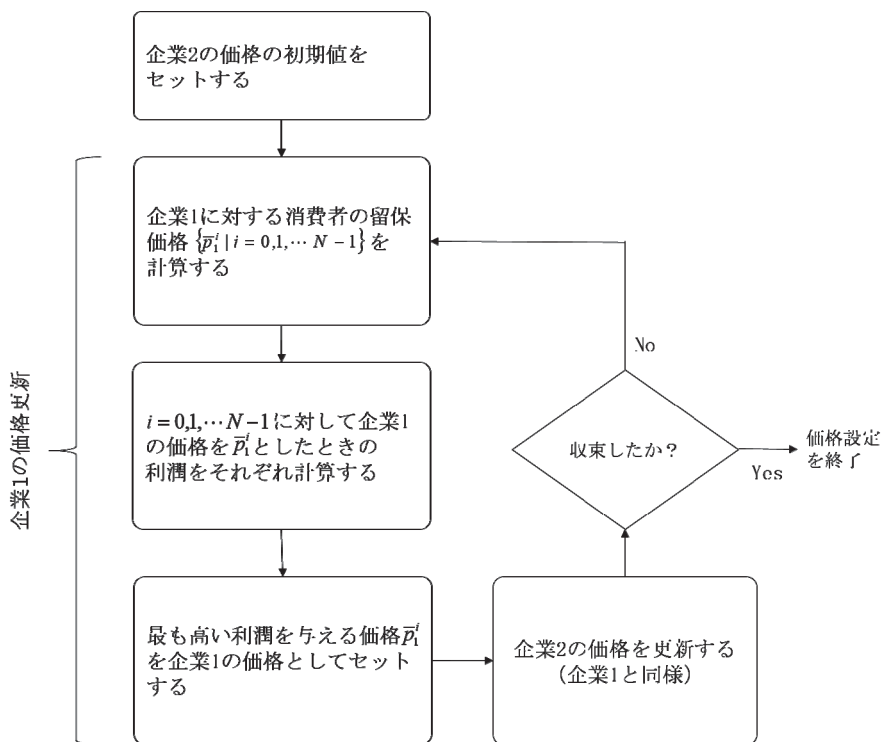


図 4-5 価格設定の流れ

同様にして企業 2 の価格更新を行い、更新による価格変化が一定の範囲に収まるまでそれを繰り返す。このプロセスを図 4-5 に示す。また、留保価格を用いた価格更新の後、さらに精度を高めるために、小さい価格更新幅 Δp を用いて交互に価格更新を行う。

次に、立地の選択を考える。価格設定のときと同じく、全ての取り得る立地の組み合わせについて扱うことは計算の都合上現実的ではないため、企業 1 と企業 2 が交互に立地選択を繰り返すとする。立地の候補となる点を市場空間の中に格子状にとり、その数を N_s とする。まず、企業 2 の立地 (x_2, y_2) を固定したときに企業 1 が立地を選択する。このとき、立地となる候補の点 $\{(x_k^s, y_k^s) | k = 0, 1, \dots, N_s - 1\}$ のそれぞれに対し、企業 1 がその点を選択した場合の価格設定プロセスを行う。全ての立地候補点に対して価格設定プロセスを実行した後、企業 1 に最も高い利潤を与える候補点を立地点として採用し、そのときの価格設定結果として得られる均衡状態の価格を、企業 1 および企業 2 の価格として設定する。

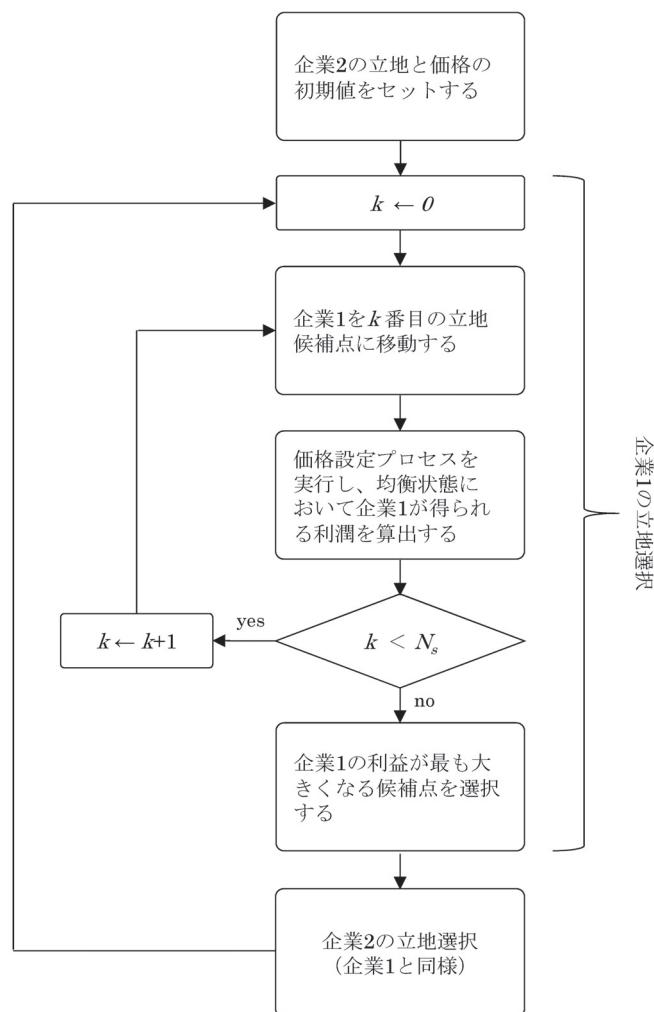


図 4-6 立地選択の全体の流れ

こうして選択された企業 1 の立地に対して、同様のプロセスで企業 2 が立地を選択する。企業エージェントがこのような交互の立地選択と価格設定を繰り返すことで立地選択が行われる。立地選択の全体の流れを図 4-6 に示す。

4.5. シミュレーション結果

前節で述べたモデルに基づいて、artiscoc[87]を用いてマルチエージェントシミュレーションを構築した。この節では表 4-1 に示す各条件のもとに実行したシミュレーションの結果を示す。

表 4-1 各シミュレーションケースの消費者分布条件

	消費者数	消費者の分布
ケースI	100	線形かつ均一に分布
ケースII	200	2次元空間のグリッド上に規則正しく分布
ケースIII	200	ランダム (一様分布)
ケースIV	200	ランダム (2つの正規分布の重ね合わせ)

ケース I

ケース I は実質的には 1 次元のモデルとなり、ホテリングのモデルをエージェントベースモデルで書き直し、移動コストの考え方を導入したものに相当する。シミュレーションの結果は図 4-7 のようになり、企業 1 と企業 2 はそれぞれ空間の端に移動して距離を最大化した。これは文献[69]の最大差別化原理の結果と一致する。なお、シミュレーション結果の図では消費者が緑色のマーカー、企業 1 と 2 がそれぞれ青と赤の四角のマーカーで表示され、×のマーカーは各企業の移動の履歴を表す。図 4-8 から図 4-10 も同様である。



図 4-7 ケース I のシミュレーション結果

ケース II

ケース 2 は Tabuchi のモデル[72]を離散化した状況に相当する。シミュレーション結果は図 4-8 のようになり、ケース I と同じく 2 つの企業は距離を最大化する方向に動いた。この結果は市場空間のうち長い軸の方向に最大差別化原理が働くという文献[72]の主張と一致し、ケース I の結果と合わせて、本研究で構築したモデルが既存研究と矛盾しない結果を生成することを示している。

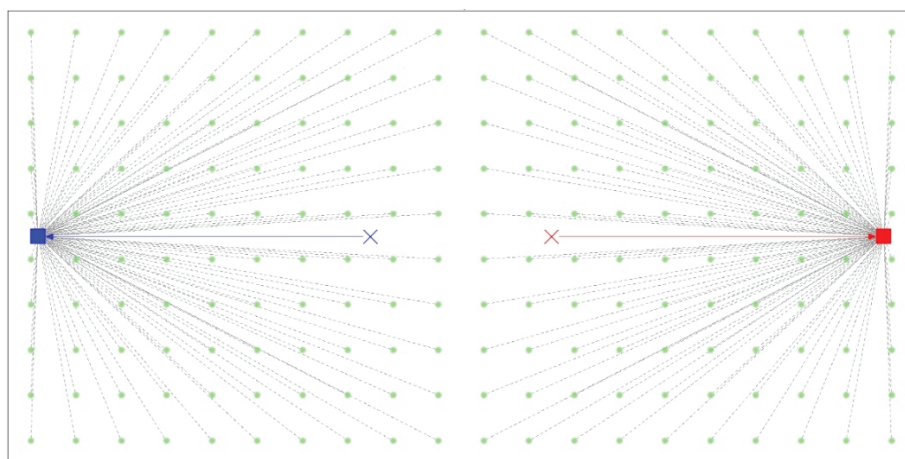


図 4-8 ケース II のシミュレーション結果

ケース III

ケース III とケース IV では消費者を規則正しく配置するのではなく、ランダムに配置した。そのうちケース III では一様分布から得られた乱数によって消費者を配置した。図 4-9 にシミュレーションの結果を示す。

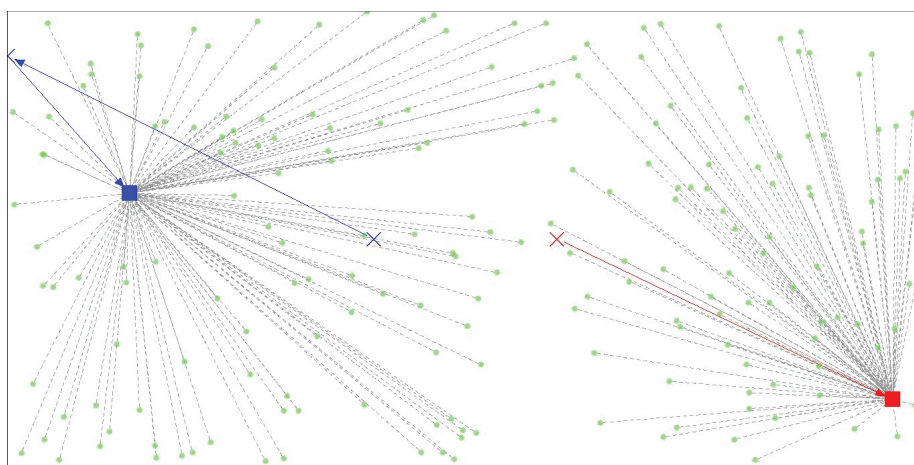


図 4-9 ケース III のシミュレーション結果

ケース IV

ケースIVではケースIIIと同じく乱数を用いて消費者をランダムに配置したが、ケースIIIのような一様な分布ではなく、2つの正規分布の和からなるような分布を用いた。消費者は正規分布の平均値に相当する2つの座標を中心に、密集している2つのエリアと、ほとんど消費者が分布しないエリアに別れた配置された。シミュレーション結果は図4-10になった。ケースIIIもケースIVも概ね企業がお互いに距離を大きくするように動いたが、ケースIIのように距離を最大化するのではなく、消費者の分布に応じて異なる位置を選択した。

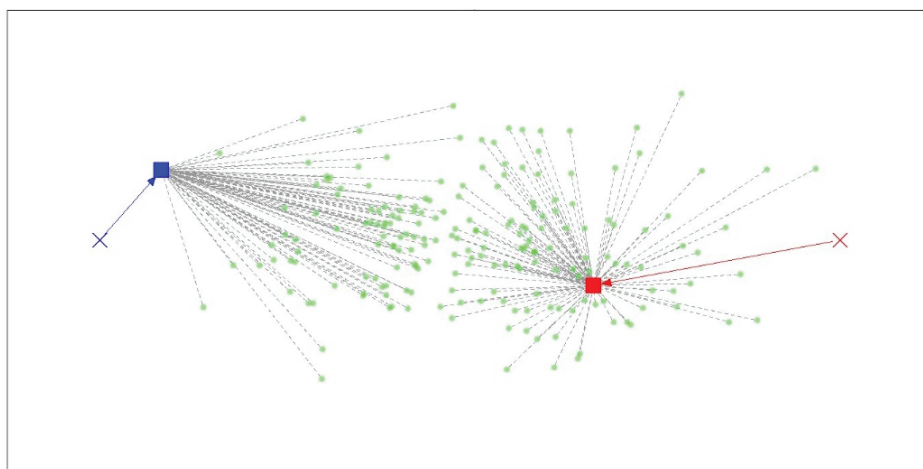


図 4-10 ケースIVのシミュレーション結果

ケースIIIとケースIVのような、市場空間にランダムに分布した消費者に対する企業の行動は、既存研究のような解析的な手法では扱うことができず、本研究のようにエージェントベースモデルとして記述することで初めて扱うことができる。

4.6. まとめ

本研究ではホテリングモデルを2次元に拡張した空間的競争モデルをさらにエージェントモデルに拡張し、マルチエージェントシミュレーションを構築した。今後は、今回構築したシミュレーションに、解析的なアプローチでは解くことのできなかつた複雑な状況を導入し、様々な状況における空間的競争について調べていく予定である。例えば、消費者の分布が一様でなかったり時間変化したりする場合、消費者による企業選択のモデルが多様な場合、市場空間における企業の移動範囲が何らかの要因で制限される場合などが考えられる。また、買い物弱者問題などの消費者と小売業者の相互作用プロセスとして生じる社会問題にも応用可能であると考えられる。

第5章 分散型電力需給制御モデル

電力ネットワークにおける分散型の需給制御アルゴリズムの構築とシミュレーションによる検証を目的とした研究において GBSP を実践した。電力を発電・送電・消費するプレイヤーをエージェントとしてモデルを構築し、マルチエージェントシミュレータを開発することで提案する需給制御則が正しく機能することを確認した。本章ではまず5.1節にてこの研究における GBSP の実践について述べた上で、5.2節以降であらためて研究内容や研究成果について述べる。

5.1. GBSP の実践

本章で述べる研究は分散協調型のアルゴリズムの開発と評価を目的としており、元よりエージェントベースなアプローチで取り組まれてきた分野の研究である。そのため、第4章の研究のような分野横断型ではなく、エージェントベースモデリングに関する専門性を持つ複数の研究者が共同で取り組む形で研究を進める形となった。GBSP は専門性の異なる研究者同士や専門性を持たないステークホルダと協働モデリングを目的とした手法ではあるが、同じ専門性を持つ研究者同士の議論であっても、エージェントベースモデルを自然かつ簡単な形で可視化しながら対話することは、議論やモデリングの作業を効果的することや、認識の齟齬をなくすといった側面で役立つと考えられる。また、研究に参加する学生は、研究を主導する専門家と比較すると専門性を持たない（または少ない）初学者であると見なすことができ、本研究におけるステークホルダに含まれる。そこで本章における研究では、モデリングや議論を効果的に進めたり、構想や研究状況を整理したり、研究に参加する学生とのコミュニケーションを円滑にしたりするという目的で、研究活動の一部分において GBSP を実践した。

5.1.1. 協働モデリングへの参加者

本章で記述する研究においては、ステークホルダとして下記のようなメンバーが GBSP を用いた協働モデリングに参加した。

■ 複数の専門家

研究を企図するエージェントベースモデリングの専門家という立場で、本論文の著者が協働モデリングに参加した。また、同じ立場と同じ専門性を持つ別の研究者も参加した。

■ ステークホルダ（初学者）

研究に参加する学生（工学部生および工学研究科大学院生）が協働モデリングに参加した。専門家と比較して相対的に低い専門性しか持たないため、専門家ではなくステークホルダであると考えられる。

5.1.2. GBSP の実践結果

この研究では、地域に分割された電力ネットワークのモデルや、そのネットワーク上で行動する電力の需要家、発電家、送電家らの行動モデル、さらにそれらのモデルに適用される分散型の制御アルゴリズムなどについて議論したりモデルを洗練したりする際に GBSP を活用した。GBSP の活用は議論の場面に応じて適宜行われ、その都度議論や共有の対象となる部分にフォーカスしてゲーム化が行われた。また、協働モデリングの形ではなく、専門家が一人で研究に関する構想や課題を整理する際にも GBSP の手法に沿った可視化を援用した。図 5-1 に、そのような際に構築されたゲームの例と、各要素が 5.3 節で論じるモデルの要素とどのように対応するのかを示す。

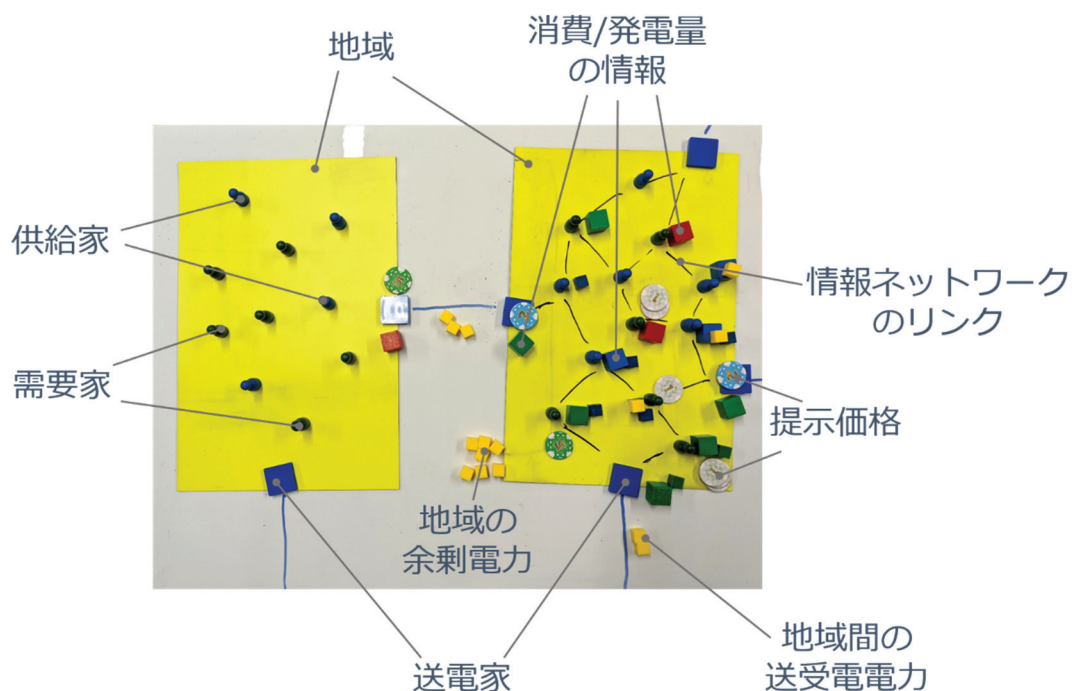


図 5-1 GBSP の成果物の例

5.1.3. GBSP の効果

本章の研究では、5.3.4 節で述べるような計算手法上の問題点を扱う場面がある。これ

は、計算や通信に一定の時間がかかるために価格更新の計算を始めるのに必要な情報の取得と価格提示のタイミングが合わないという問題であるが、そのような問題があることを理解したり伝えたりするのは、数式で記述したモデルや文面での説明だけでは難しく、学生らと共同で研究を進めたり、専門家自身が問題を整理したりする際の障害となり得た。しかし、GBSPを活用し、図 5-1 のようなゲームとしてのモデルの上でトークンを動かしたりしながら議論することで、そのような分かりにくい状況を扱う際にも効果的な議論やアイデアの検討を行うことができた。その結果、次節以降で述べる研究や、その関連研究を円滑に進めることができ、国際的に評価されている学術誌に論文が掲載される[79]など、優れた研究成果を生み出すことができた。

5.2. 研究の背景・目的

化石燃料の価格高騰や原子力発電所の安全性問題など、エネルギーに関する諸問題への関心が世界中で高まる中、太陽光発電や風力発電などの自然エネルギーの導入が積極的に進められている[73]。自然エネルギー利用の大きな問題点の一つに、供給量の不安定性がある。自然エネルギーによる供給量は時刻、天候等の条件によって大きく変化し、その予測が難しい。一方で、電力系統内の周波数を保つために、系統内での供給量と消費量が一致している必要がある。そこで、自然エネルギー利用の割合が大きい電力ネットワークにおいては、供給量予測が難しい中でいかにして供給量と需要量を一致させるかが課題となり、様々な手法が提案・研究されている[75]。これらのうち本研究では、実時間における価格調整によって消費量と供給量のバランスをとるリアルタイムプライシング[76]に着目する。

リアルタイムプライシングの基本的原理は、需要量が大き過ぎる時は電力価格を上げることで消費量の抑制と供給量の増加を促し、供給量が過剰な時は電力価格を下げることで消費量の増加と供給量の抑制を促すというものである。このとき、リアルタイムプライシングを伴う電力システムは、消費量や供給量を観測量、電力価格を制御入力とするフィードバック制御系とみなすことができる[77]。[75]-[77]に挙げた文献をはじめ、リアルタイムプライシングの先行研究においては、システム全体の情報(例えば総需要供給量など)を必要とする集中型の価格調整則が用いられることが一般的である。集中型のリアルタイムプライシングは、計算・通信の負荷が高い中央処理装置を必要とし、中央処理装置に関連する障害によってシステム全体が影響を受けてしまう。これに対して本研究では、集中的な処理を必要とせず、各々の需要供給家がローカルな情報のみを用いて適切な価格を導出できるような

分散型の価格調整則を実現することを目指している。この方法は、中央処理装置を必要とせず、スケーラビリティとディペンダビリティに優れている[78]。

これまでに著者らは、リアルタイムプライシングの問題を制約付き最適化問題として扱い、分散協調制御の理論を応用した分散型の価格調整則を提案してきた[79][80]。これは、各プレイヤー(需要家, 供給家および送電家)が自身の需要供給量に基づいた情報を周囲のプレイヤーと交換するプロセスを経て、適切な価格を推定するという方法である。これらの研究では、情報取得や価格推定のプロセスが極めて短い時間で行われ、即座に価格が提示されることが仮定されている。しかし、実システムにおいては通信容量, 通信時間, 記憶容量, 計算時間などの制限から、情報交換や価格推定のプロセスに一定の時間を要する。これによって、価格は情報を取得した時刻よりも遅れて提示されることになる。このような時間遅れはフィードバックシステムを不安定化する、あるいは性能を劣化させることが知られている[81]。

本稿ではこの問題に対応し、過去のデータから各プレイヤーの需要供給量の予測を行うことによって、価格提示の遅れによる電力システムの不安定化・性能劣化を解消する方法を提案する。また、分散型リアルタイムプライシングについて様々な条件下でのシミュレーションと結果に対する考察を効率良く行うために開発したグラフィカルシミュレータについても紹介し、提案手法のシミュレーション結果と合わせて示す。

5.3. 問題設定

本節では対象とする電力ネットワークのモデルと制約付き最適化問題として扱うリアルタイムプライシングを定式化し、分散型価格調整則に関する問題設定を行う。

5.3.1. ネットワークモデル

本研究では、リアルタイムプライシングを適用する電力ネットワークとして、図5-2に示すような地域に分割された電力ネットワークを想定する。これは、各地域内に多数の需要供給家が存在し、地域ごとに需給バランスが求められる一方、送電家によって地域間で電力を融通しあうような電力ネットワークである。ここでの「地域」は、必ずしも地理的なまとまりや行政区分を表すのではなく、例えば同一の電力事業者と契約している需要供給家の集まりからなるサブネットワークとすることもできる。また、各地域内のプレイヤーは情報ネットワークを構成しており、計算・通信を担うスマートメータを介して、隣接するプレイヤー間で需要供給量などの情報を交換することができるとする。

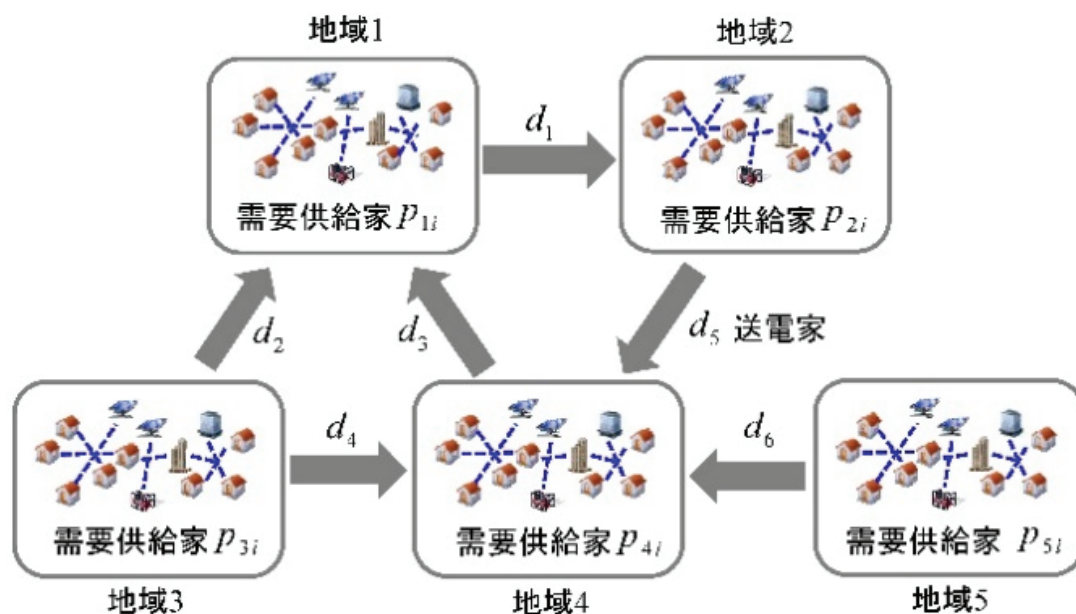


図 5-2 ネットワークモデル

上記のネットワークにおいて、各地域における消費量と供給量が一致するように価格を調整することを考える。地域の数を $n \in \mathbb{N}$ とし、地域集合を $\mathcal{R} = \{1, 2, \dots, n\}$ とする。ある地域 $r \in \mathcal{R}$ には $n_r \in \mathbb{N}$ の需要供給家が含まれるとし、その集合を $\mathcal{P}_r = \{1, 2, \dots, n_r\}$ とする。 \mathcal{P}_r に含まれる i 番目の需要供給家の消費量または供給量 $x_{ri} \in \mathbb{R}$ とする。ここで x_{ri} が負の場合は消費量を、正の場合は供給量を意味する。地域間にはある地域から別の地域へ電力を送電する送電家が存在して、その集合を $\mathcal{D} = \{1, 2, \dots, m\}$, $m \in \mathbb{N}$ とする。ここで、地域 $r \in \mathcal{R}$ を送電側とする送電家の集合を $\mathcal{D}_r^- \subset \mathcal{D}$ とし、地域 $r \in \mathcal{R}$ を受電側とする送電家の集合を $\mathcal{D}_r^+ \subset \mathcal{D}$ とする。さらに、送電家 $k \in \mathcal{D}$ の送電量を $\bar{x}_k \in \mathbb{R}$ とあらわす。 $k \in \mathcal{D}_r^-$ のとき地域 r から見た送電家 k は消費量が \bar{x}_k の需要家と同等であり、 $k \in \mathcal{D}_r^+$ のとき地域 r から見た送電家 k は供給量が \bar{x}_k の供給家と同等である。

各地域内の情報ネットワークにおいて、プレイヤー $i \in \mathcal{P}_r$ と隣接するプレイヤーの集合を \mathcal{N}_i とする。価格調整にあたっては、各プレイヤーは \mathcal{N}_i に含まれるプレイヤーと情報を交換することはできるが、他のプレイヤーから直接情報を得ることはできないとする。隣接集合 \mathcal{N}_i に含まれるプレイヤーとの情報交換は、実システムにおいては、例えばスマートメータ間の無線通信などに相当し、通信や計算には無視できない一定の時間を要すると考えられる。本研究ではこのような将来問題になりうる環境を想定し、一定の時間間隔ごとでしか情報交換を行うことができない場合を考える。

次に、各プレイヤーの行動モデルについて示す。各プレイヤーは提示された電力価格に応じて消費量や供給量を変化させるとする。このようなモデルは、消費者による事前の設定に従って、提示価格に応じて自動で電力使用量を変化させるオートデマンドレスポンス装置[82]の導入を想定している。ここで、各プレイヤーの消費量や供給量の変化を特徴づける量として効用を導入し、地域 $r \in \mathcal{R}$ に含まれる i 番目の需要供給家の効用を $U_{ri}(x_{ri}) \in \mathbb{R}$ 、 k 番目の送電家の効用を $\bar{U}_k(\bar{x}_k) \in \mathbb{R}$ と表す。効用は、各プレイヤーがある消費量や供給量をとったときの満足度の度合いを表す関数である。各需要家にとって、消費電力量が所望量よりも小さいときは、使用目的に十分な電力を使うことができず、満足度が下がると考えられる。逆に消費電力量が所望量よりも大きいときは、電気を使い過ぎてしまっている状態であり、やはり満足度が下がると考えられる。また、供給家や送電家にとっての所望量は発電設備の能力や仕様に依って定まる量であり、発電量が少な過ぎるときも多過ぎるときも満足度が下がる。このように、効用は各プレイヤーが持つ所望量に等しい消費量や供給量対して最大値をとるような、上に凸な関数であると考えられる。各プレイヤーは式(1)に従って需要供給量を変化させるとする。ここで λ は電力価格、 k は需要供給量変化のステップ数、 α は適当な正数である。

$$x_{ri}[k+1] = x_{ri}[k] + \alpha \left(\frac{\partial U_{ri}}{\partial x_{ri}}(x_{ri}[k]) + \lambda \right) \quad (1)$$

$x_{ri}[k]$ が収束するとき、その収束値においては、

$$\frac{\partial U_{ri}}{\partial x_{ri}} \left(\lim_{k \rightarrow \infty} x_{ri}[k] \right) + \lambda = 0 \quad (2)$$

が成り立つ。式(2)は電力価格 λ と限界効用が釣り合う点を表す。

5.3.2. 最適化アルゴリズムによる価格調整

本研究では、価格調整によって需要と供給のバランスを得るというリアルタイムプライシングの問題を、以下のような制約付き最適化問題として扱う。プレイヤー(需要供給家および送電家)の消費量、供給量、送電量をまとめて決定変数として扱い、

$$x = (x_1, x_2, \dots, x_n, \bar{x}_1, \bar{x}_2, \dots, \bar{x}_m) \in \mathbb{R}^v$$

とする。ここで x_r は地域 $r \in \mathcal{R}$ に含まれるプレイヤーの消費量・供給量・送電量を表し、 $x_r = (x_{r1}, x_{r2}, \dots, x_{rn_r})$ である。このとき、決定変数ベクトルの次数は $v = \sum_{r=1}^n n_r + m$ である。目

的関数を $F(x)$ 、制約関数を $G(x)$ とすると、最適化問題は

$$\begin{cases} \underset{x \in \mathbb{R}^v}{\text{maximize}} & F(x) \\ \text{subject to} & G(x) = 0 \end{cases} \quad (3)$$

となる。目的関数 $F(x) \in \mathbb{R}$ はプレイヤーの効用の和であり、

$$F(x) = \sum_{r=1}^n \left(\sum_{i=1}^{n_r} U_{ri}(x_{ri}) \right) + \sum_{k=1}^m \bar{U}_k(\bar{x}_k) \quad (4)$$

とかける。一方、制約関数 $G(x) \in \mathbb{R}^n$ は需給バランスに相当し、その r 番目の要素 $G_r(x) \in \mathbb{R}$ は下記のように書ける。

$$G_r(x) = \sum_{i=1}^{n_r} x_{ri} + \sum_{k \in \mathcal{D}_r^+} \bar{x}_k - \sum_{k \in \mathcal{D}_r^-} \bar{x}_k \quad (5)$$

問題(3)は目的関数 $F(x)$ が上の凸のとき最適解を持つ[83]。前節で示したようにプレイヤーの効用は上に凸であり、その総和である $F(x)$ も上に凸となるため、問題(3)は最適解 $x^* \in \mathbb{R}^v$ を持つ。更新変数 $x(s)$ を導入すると、最適解は

$$\lim_{s \rightarrow \infty} x(s) = x^* \quad (6)$$

と表すことができる。このような制約付き最適化問題に対しては、ラグランジュの未定乗数法に基づく方法が適している[83]。まず、ラグランジュの未定乗数

$$\lambda = (\lambda_1, \lambda_2, \dots, \lambda_n) \in \mathbb{R}^n$$

を導入し、ラグランジアンを(7)のように定義する。

$$L(x, \lambda) = F(x) + \sum_{r=1}^n \lambda_r G_r(x) \quad (7)$$

一般的なラグランジュの未定乗数法[83]に基づき、 x の更新則は以下に示す手順**A**のように記述できる。

(手順 A)

A-i. 初期値 $x^{[0]} \in \mathbb{R}^v$, $\lambda^{[0]} \in \mathbb{R}^n$ と正数 α , β , 自然数 K を与え, ステップ $s = 0$ とする.

A-ii. 各プレイヤーが需要供給量 $x_{ri}^{[s]}$ の更新を行う.

A-ii-1. $k = 0$, $x_{ri}[0] = x_{ri}^{[s]}$ とする

A-ii-2. 以下の式(8)に従って $x_{ri}[k]$ を $k = 0$ から $k = K$ まで逐次的に計算する.

$$\begin{aligned} x_{ri}[k+1] &= x_{ri}[k] + \alpha \frac{\partial L}{\partial x_{ri}}(x[k], \lambda^{[s]}) \\ &= x_{ri}[k] + \alpha \left(\frac{\partial U_{ri}}{\partial x_{ri}}(x_{ri}[k]) + \lambda_r^{[s]} \right) \end{aligned} \quad (8)$$

A-ii-3. $k = K$ における値をステップ $s + 1$ における初期値とする.

$$x_{ri}^{[s+1]} = x_{ri}[K] \quad (9)$$

A-iii. 未定乗数 $\lambda_r^{[s]}$ の更新を行う.

$$\begin{aligned} \lambda_r^{[s+1]} &= \lambda_r^{[s]} - \beta \frac{\partial L}{\partial \lambda_r}(x^{[s+1]}, \lambda^{[s]}) \\ &= \lambda_r^{[s]} - \beta G_r(x^{[s+1]}) \end{aligned} \quad (10)$$

A-iv. ステップを s から $s + 1$ に更新し, A-ii へ戻る.

手順 A において式(1)と式(8)は同じ形であり, 最適化問題の解法におけるラグランジュの未定乗数が, 電力価格に相当するものであることが分かる[84]. 図 5-3 に手順 A をフローチャートで表したものを示す.

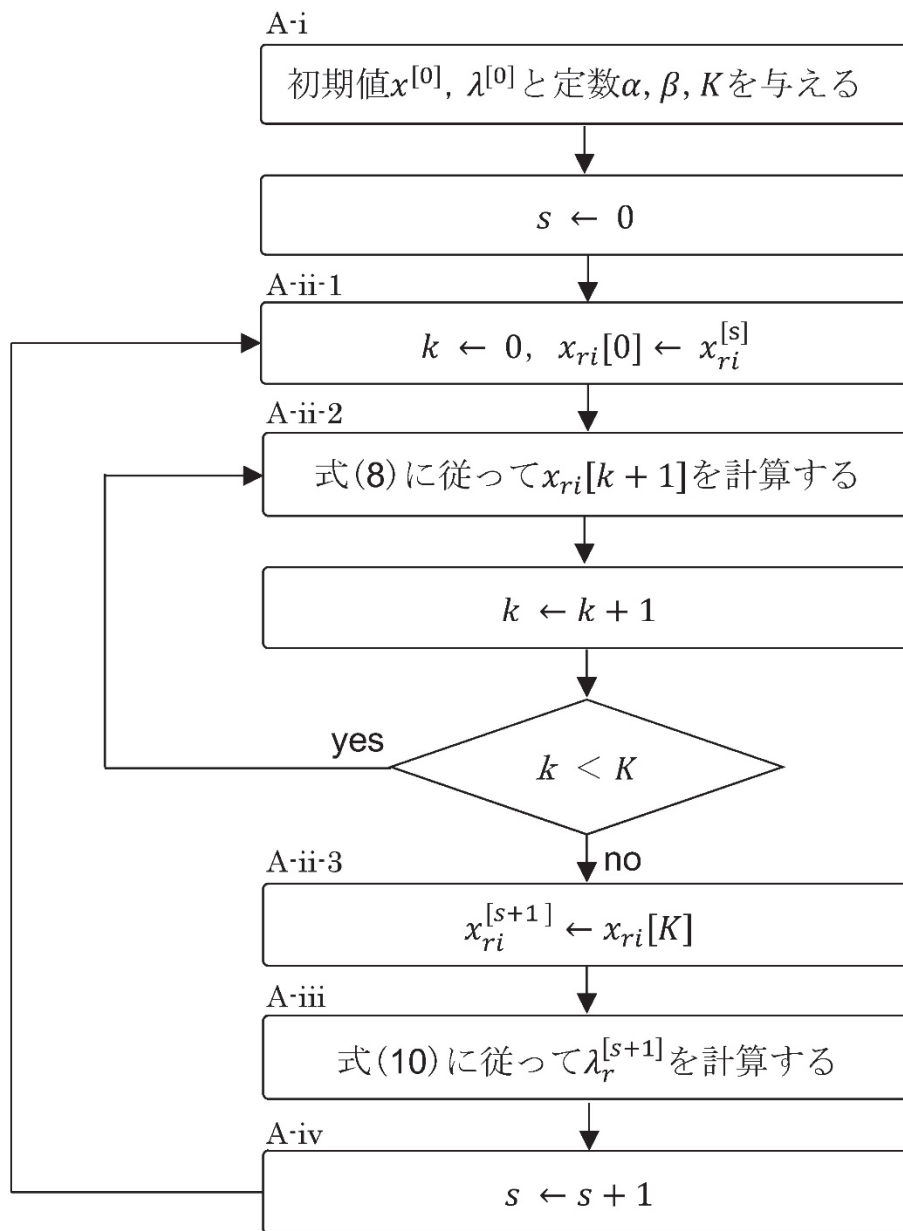


図 5-3 フローチャート

5.3.3. 従来の分散型価格調整法

適切な正数 α, β を設定すれば、手順Aに従う更新プロセスを繰り返すことで最適解 x^* を得ることができる[83]。ここで、式(10)で用いる $G_r(x^{[s]})$ は地域に含まれる全てのプレイヤーの情報を含んでいる。しかし、各プレイヤーは地域全体の情報を直接得ることができず、ネ

ネットワークにおいて隣接するプレイヤーの情報を一定の間隔（手順 **A** におけるステップ s の更新間隔）毎に取得することができるだけである。そのため、このままでは手順 A-iii の式(10)による価格更新を計算することができない。

このような問題に対して、文献[79][80]において、著者らのグループは、分散協調制御の理論を応用した分散型の価格調整則を提案している。これは、各地域内において構成された情報ネットワークを通して、各プレイヤーが隣接するプレイヤーと情報を反復的に交換することで、適切な価格を推定するという方法である。これによって、各プレイヤーは、情報ネットワークにおける隣接集合 \mathcal{N}_i に含まれるプレイヤーから得られる局所的な情報 $x_{rj}^{[s]}, j \in \mathcal{N}_i$ および s ステップ目における価格と自身の需要供給量の初期値 $x_{ri}^{[s+1]}$ を用いて、手順 A-iii. の式(10)と等しい結果を得る計算を行うことができ、 $s+1$ ステップ目における価格更新を実行できる。具体的には、次の更新則を用いる。

$$\begin{cases} \theta_{ri}[0] = \hat{\lambda}_{ri}^{[s]} - cx_{ri}^{[s+1]} \\ \theta_{ri}[h+1] = \theta_{ri}[h] - \gamma \sum_{j \in \mathcal{N}_i} (\theta_{ri}[h] - \theta_{rj}[h]) \\ \hat{\lambda}_{ri}^{[s+1]} = \theta_{ri}[H] \end{cases} \quad (11)$$

ここで c, γ は適当な正数、 $\hat{\lambda}_{ri}^{[s]}$ は s ステップ目における需要供給家 $i \in \mathcal{P}_r$ に提示される価格、 $\theta_{ri}[h]$ は提示価格を求めるための状態変数である。エージェントが送電家の場合も、同様の式で表すことができる。文献[79][80]の議論により、各地域の情報ネットワークが連結の場合、反復ステップ数 H を十分に大きくとれば、式(11)は同一地域内の全てのプレイヤーについての状態変数の終端値 $\theta_{ri}[H]$ が一致する。また、合意した収束値 $\theta_{ri}[H]$ は式(10)によって得られる $s+1$ ステップ目の価格 $\lambda_r^{[s+1]}$ に等しくなる。

5.3.4. 従来法の問題点

(11)式の分散型価格調整法は、手順 A-iii.の式(10)の代わりに利用され、 $s+1$ ステップ目における価格 $\hat{\lambda}_{ri}^{[s+1]}$ を計算する。ここで、問題となるのは、(11)式の計算は、(10)式とは異なり、情報交換を伴う繰り返し計算から構成されており、通信時間および計算時間がかかることである。これによって、各プレイヤーが式(8)に従って需要供給量を変化させる際に、価格 $\hat{\lambda}_{ri}^{[s]}$ がまだ求まっていないことがある。このような場合は、1ステップ前の価格 $\hat{\lambda}_{ri}^{[s-1]}$ を利用

しなければならない。これは、フィードバックシステムの時間遅れに相当し、システムが不安定化する、あるいは性能が劣化する可能性がある。実際、第4章のシミュレーションにおいても、aaaに示すようにシステムが不安定となり、需要供給量が発散してしまうことが確かめられている。

このような問題を解決するため、本論文では、 $x_{ri}^{[s+1]}$ を事前に予測することで(11)式の計算を $s+1$ ステップ目がはじまる前に実行し、時間遅れが生じないようにする手法を提案する。この際、各プレイヤーが得ることのできる情報は、 $s-1$ ステップ目における需要供給量 $x_{ri}^{[s]}$ と価格 $\hat{\lambda}_{ri}^{[s]}$ のみであるとする。これ以前における各値も情報としては入手できるが、各プレイヤーの持つデバイスのメモリが大きくないものとして、直前の $s-1$ ステップ目の情報のみが利用できるものとする。以上をまとめ、次のような問題を考える。

(問題1)

$s-1$ ステップ目までの需要供給量 $x_{ri}^{[s]}$ と価格 $\hat{\lambda}_{ri}^{[s-1]}$, $\hat{\lambda}_{ri}^{[s]}$ を用いて、 s ステップ目の需要供給量 $x_{ri}^{[s+1]}$ を推定せよ。

5.4. 提案手法

本章では、需要供給量 $x_{ri}^{[s]}$ と価格 $\hat{\lambda}_{ri}^{[s-1]}$, $\hat{\lambda}_{ri}^{[s]}$ のみを用いて $x_{ri}^{[s+1]}$ を予測する手法として、逐次線形回帰を用いた需要供給量予測法を提案する。ここでは、上に凸なプレイヤーの効用関数を、上に凸な2次関数であると近似することで、価格 $\hat{\lambda}_{ri}^{[s]}$ と需要供給量の収束値 $x_{ri}^{[s+1]}$ が線形な関係となり、逐次的で計算負荷の小さい需要供給量予測が可能となることを利用する。

ある価格 $\hat{\lambda}_{ri}^{[s]}$ が提示された場合の需要供給量の収束値 $x_{ri}^{[s+1]}$ について、式(2)が成り立つため

$$\frac{\partial U_{ri}}{\partial x_{ri}}(x_{ri}^{[s+1]}) + \hat{\lambda}_{ri}^{[s]} = 0$$

$$x_{ri}^{[s+1]} = \left(\frac{\partial U_{ri}}{\partial x_{ri}}\right)^{-1} \hat{\lambda}_{ri}^{[s]} \quad (12)$$

となる。ここで、プレイヤーの効用関数が上に凸な2次関数 $\hat{U}_{ri}(x_{ri})$ であると近似し、実数の係数 $A_{ri}, B_{ri}, C_{ri} \in \mathbb{R}$ を用いて

$$\hat{U}_{ri}(x_{ri}) = A_{ri}(x_{ri} - B_{ri})^2 + C_{ri} \quad (13)$$

と書く．このときの需要供給量の収束値 $\hat{x}_{ri}^{[s+1]}$ は式(12)より

$$\hat{x}_{ri}^{[s+1]} = -\frac{1}{2A_{ri}}\hat{\lambda}_{ri}^{[s]} + B_{ri} \quad (14)$$

となり，提示価格と需要供給量の間には線形な関係が成り立つ．式(14)の係数をそれぞれ $a_{ri}^{[s]} = -1/(2A_{ri})$ ， $b_{ri}^{[s]} = B_{ri}$ と表記すると，

$$\hat{x}_{ri}^{[s+1]} = a_{ri}^{[s]}\hat{\lambda}_{ri}^{[s]} + b_{ri}^{[s]} \quad (15)$$

と書き換えることができる．ここで，過去の提示価格 $\hat{\lambda}_{ri}^{[0]}, \hat{\lambda}_{ri}^{[1]}, \dots, \hat{\lambda}_{ri}^{[s-1]}$ とそれに対する需要供給量収束値 $x_{ri}^{[1]}, x_{ri}^{[2]}, \dots, x_{ri}^{[s]}$ の組に対して最小二乗法を用いると式(16)のように回帰直線の係数が得られる[86]．

$$\begin{cases} a_{ri}^{[s]} = \frac{s \sum_{l=0}^{s-1} \hat{\lambda}_{ri}^{[l]} x_{ri}^{[l+1]} - \sum_{l=0}^{s-1} \hat{\lambda}_{ri}^{[l]} \sum_{l=0}^{s-1} x_{ri}^{[l+1]}}{s \sum_{l=0}^{s-1} (\hat{\lambda}_{ri}^{[l]})^2 - \left(\sum_{l=0}^{s-1} \hat{\lambda}_{ri}^{[l]} \right)^2} \\ b_{ri}^{[s]} = \frac{\sum_{l=0}^{s-1} (\hat{\lambda}_{ri}^{[l]})^2 \sum_{l=0}^{s-1} x_{ri}^{[l+1]} - \sum_{l=0}^{s-1} \hat{\lambda}_{ri}^{[l]} x_{ri}^{[l+1]} \sum_{l=0}^{s-1} \hat{\lambda}_{ri}^{[l]}}{s \sum_{l=0}^{s-1} (\hat{\lambda}_{ri}^{[l]})^2 - \left(\sum_{l=0}^{s-1} \hat{\lambda}_{ri}^{[l]} \right)^2} \end{cases} \quad (16)$$

式(16)によって計算した係数 $a_{ri}^{[s]}$ ， $b_{ri}^{[s]}$ を用いると，式(15)により需要供給量収束値の推定値 $\hat{x}_{ri}^{[s+1]}$ を得ることができる．ここで，需要供給量収束値の真の値は式(12)で与えられるため，提示価格 $\hat{\lambda}_{ri}^{[s]}$ に対応する推定値 $\hat{x}_{ri}^{[s+1]}$ の残差は

$$x_{ri}^{[s+1]} - \hat{x}_{ri}^{[s+1]} = \left(\frac{\partial U_{ri}}{\partial x_{ri}} \right)^{-1} \hat{\lambda}_{ri}^{[s]} - \left(a_{ri}^{[s]} \hat{\lambda}_{ri}^{[s]} + b_{ri}^{[s]} \right) \quad (17)$$

である．ここでは最小二乗法を用いているため，式(16)によって得られる $a_{ri}^{[s]}$ ， $b_{ri}^{[s]}$ は残差の二乗平均

$$\frac{1}{s+1} \sum_{l=0}^s \left(x_{ri}^{[s+1]} - \hat{x}_{ri}^{[s+1]} \right)^2 = \frac{1}{s+1} \sum_{l=0}^s \left\{ \left(\frac{\partial U_{ri}}{\partial x_{ri}} \right)^{-1} \hat{\lambda}_{ri}^{[l]} - \left(a_{ri}^{[l]} \hat{\lambda}_{ri}^{[l]} + b_{ri}^{[l]} \right) \right\}^2 \quad (18)$$

を最小にする．プレイヤーの効用を2次関数と近似したとき，式(15)および式(16)による需要供給量の推定は，誤差の二乗平均の面で最も良い推定を与える．

真の効用 $U_{ri}(x_{ri})$ が2次関数のとき、式(18)の値は0となり、 $U_{ri}(x_{ri})$ の形が2次関数から離れるほど、式(18)の値が大きくなり、推定の誤差が大きくなる。

式(16)では $l=0$ から $l=s-1$ までの全てのステップの提示価格 $\hat{\lambda}_{ri}^{[l]}$ と決定変数の収束値 $x_{ri}^{[l+1]}$ の和を計算しており、一つ前のステップにおける情報である需要供給量 $x_{ri}^{[s]}$ と提示価格 $\hat{\lambda}_{ri}^{[s-1]}, \hat{\lambda}_{ri}^{[s]}$ のみしか利用できないという**問題 1**の条件を満たしていない。そこで、和記号の部分を書式(17)~(20)のような形で個別のパラメータとして考える。

$$u_{xri}^{[s]} = \sum_{l=1}^s x_{ri}^{[l]} \quad (17)$$

$$u_{\hat{\lambda}ri}^{[s]} = \sum_{l=0}^{s-1} \hat{\lambda}_{ri}^{[l]} \quad (18)$$

$$\sigma_{\hat{\lambda}ri}^{[s]} = \sum_{l=0}^{s-1} \left(\hat{\lambda}_{ri}^{[l]} \right)^2 \quad (19)$$

$$\sigma_{x\hat{\lambda}ri}^{[s]} = \sum_{l=0}^{s-1} \hat{\lambda}_{ri}^{[l]} x_{ri}^{[l+1]} \quad (20)$$

このとき、式(16)は次式のように書き換えられる。

$$a_{ri}^{[s]} = \frac{s\sigma_{x\hat{\lambda}ri}^{[s]} - u_{\hat{\lambda}ri}^{[s]} u_{xri}^{[s]}}{s\sigma_{\hat{\lambda}ri}^{[s]} - \left(u_{\hat{\lambda}ri}^{[s]} \right)^2} \quad (21)$$

$$b_{ri}^{[s]} = \frac{\sigma_{\hat{\lambda}ri}^{[s]} u_{xri}^{[s]} - \sigma_{x\hat{\lambda}ri}^{[s]} u_{\hat{\lambda}ri}^{[s]}}{s\sigma_{\hat{\lambda}ri}^{[s]} - \left(u_{\hat{\lambda}ri}^{[s]} \right)^2} \quad (22)$$

あとは、 $u_{xri}^{[s]}, u_{\hat{\lambda}ri}^{[s]}, \sigma_{\hat{\lambda}ri}^{[s]}, \sigma_{x\hat{\lambda}ri}^{[s]}$ を一つ前のステップの情報である $x_{ri}^{[s]}$ と $\hat{\lambda}_{ri}^{[s-1]}, \hat{\lambda}_{ri}^{[s]}$ のみを用いて更新することができれば、式(14)に相当する係数を逐次計算することができるため、**問題 1**の解となる需要供給量の推定則が構築される。このような手順をまとめたものを以下に示す。

(手順 B)

B-i. 予測のための係数 $a_{ri}^{[s]}$, $b_{ri}^{[s]}$ と逐次更新されるパラメータ $u_{xri}^{[s]}$, $u_{\hat{\lambda}ri}^{[s]}$, $\sigma_{\hat{\lambda}ri}^{[s]}$, $\sigma_{x\hat{\lambda}ri}^{[s]}$ を準備し, 初期値を

$$a_{ri}^{[0]} = b_{ri}^{[0]} = u_{xri}^{[0]} = u_{\hat{\lambda}ri}^{[0]} = \sigma_{\hat{\lambda}ri}^{[0]} = \sigma_{x\hat{\lambda}ri}^{[0]} = 0$$

とする.

B-ii. ステップ $s-1$ の終わりに各パラメータを更新する. これまでのステップで提示された価格 $\hat{\lambda}_{ri}^{[s-1]}$ と需要供給量の収束値 $x_r^{[s]}$ を用いて, 各パラメータを以下のように更新する.

$$u_{xri}^{[s]} = u_{xri}^{[s-1]} + x_{ri}^{[s]} \quad (23)$$

$$u_{\hat{\lambda}ri}^{[s]} = u_{\hat{\lambda}ri}^{[s-1]} + \hat{\lambda}_{ri}^{[s-1]} \quad (24)$$

$$\sigma_{\hat{\lambda}ri}^{[s]} = \sigma_{\hat{\lambda}ri}^{[s-1]} + \left(\hat{\lambda}_{ri}^{[s-1]}\right)^2 \quad (25)$$

$$\sigma_{x\hat{\lambda}ri}^{[s]} = \sigma_{x\hat{\lambda}ri}^{[s-1]} + \hat{\lambda}_{ri}^{[s-1]} x_{ri}^{[s]} \quad (26)$$

さらに, $a_{ri}^{[s]}$, $b_{ri}^{[s]}$ を式(21), (22)を用いて更新する.

B-iii. 式(15)に従って $\hat{\lambda}_r^{[s]}$ から需要供給量の予測値 $\hat{x}_{ri}^{[s+1]}$ を求める. このようにして得られた $\hat{x}_{ri}^{[s+1]}$ を $x_{ri}^{[s+1]}$ の代わりに用いて式(11)を実行する.

B-vi. 手順 A におけるステップ s の更新にあわせて, B-ii へ戻る.

手順 B により, $x_{ri}^{[s]}$, $\hat{\lambda}_{ri}^{[s-1]}$, $\hat{\lambda}_{ri}^{[s]}$ のみを用いて, s ステップ目の需要供給量 $x_{ri}^{[s+1]}$ を推定することができる. これにより, **問題 1**が解決される.

提案手法では各プレイヤーの計算デバイスは過去の履歴データを保持しておく必要はなく, 式(17)~(20)で定義されるパラメータの最新の値だけを保持し, 式(23)~(26)に従って逐次的に更新していけばよい. このため, スマートメータなどの計算デバイスには多量のデータを用いた高負荷処理や大きなメモリを必要とせず, 分散的なシステムに適した手法となっている.

5.5. シミュレーション

5.5.1. シミュレータの開発

提案する分散的リアルタイムプライシングについて、様々な条件下での振る舞いを詳細に調べるために、グラフィカルかつスケーラブルな独自のシミュレータを開発した。図 5-4 に開発したシミュレータの動作画面を示す。

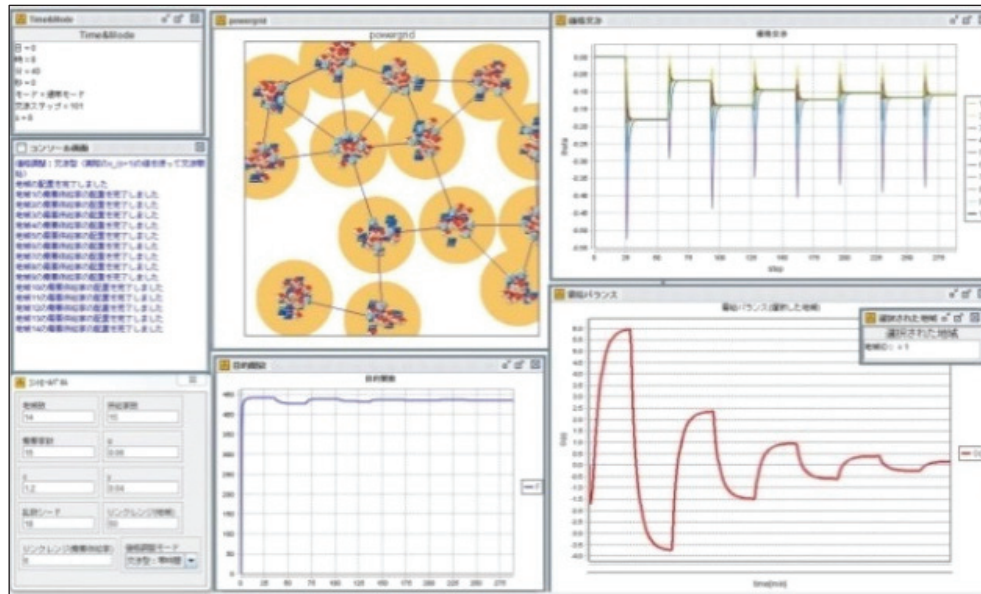


図 5-4 開発したシミュレータの画面キャプチャ

シミュレータの開発には、構造計画研究所が開発したマルチエージェントシミュレーションプラットフォーム「artisoc3.0」を用いた。artisoc はマルチエージェントシミュレーションの構築に便利な各種ライブラリを備え、エージェント行動の記述や出力設定を簡易な操作で行うことができる開発プラットフォームであり、JAVA を介して動作するグラフィカルなシミュレータを構築することができる[87]。

このシミュレータでは、GUI によるコントロールパネルから地域数、供給家数、需要家数、 α 、 c 、 γ などのパラメータ、ネットワークの粗密などの各種パラメータを設定したり、需要供給予測の有無を切り替えたりすることができる。

シミュレーションの開始時には、コントロールパネルから設定された地域数とエージェント数に応じて送電ネットワークおよび情報ネットワークが自動で生成され図 5-5 のようにグラフィカルな形で表示される。

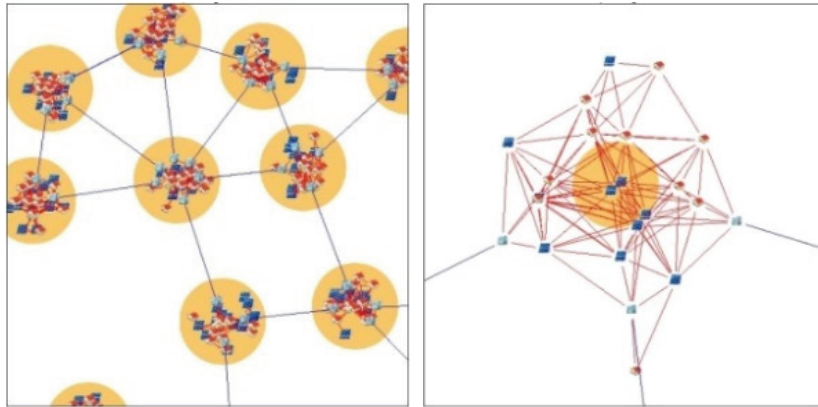


図 5-5 ネットワークの出力

5.5.2. シミュレーション結果

前節で紹介した独自のシミュレータを用いて、本稿で提案する手法の有効性を示すためのシミュレーションを行った。今回のシミュレーションケースでは、地域数を 15 とし、各地域に需要供給家を 20 ずつ配置して図 5-5 のようなネットワークを構築した。各プレイヤーの効用関数としては式(27)で示すような 2 種類の 2 次関数を頂点で接合したものを設定した。ここで $a_{ri}, b_{ri}, d_{ri}, e_{ri}$ は各プレイヤーに固有のパラメータである。

$$U_{ri}(x_{ri}) = \begin{cases} d_{ri}(x_{ri} - a_{ri})^2 + b_{ri} & \text{if } x_{ri} < a_{ri}, \\ e_{ri}(x_{ri} - a_{ri})^2 + b_{ri} & \text{if } x_{ri} \geq a_{ri} \end{cases} \quad (26)$$

また、ゲインパラメータとして $\alpha = 0.06, c = 0.6, \gamma = 0.03$ を設定した。

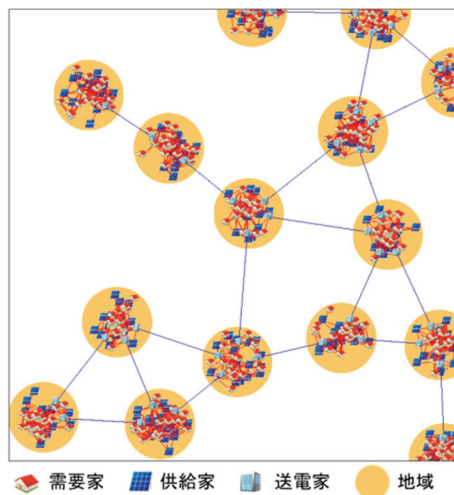


図 5-6 シミュレーションに用いたネットワークモデル

上記のような条件のもとに数値シミュレーションを行った結果を以下に示す。まず、ケース I では、提案手法による需要供給量予測を行わず、式(11)第一式において $x_{ri}^{[s+1]}$ の代

わりにその時点で得ることのできる最新の需要供給量である $x_{ri}^{[s]}$ を用いた。この場合の各地域の制約関数 $G_r(x)$ の時間変化のグラフを図 5 に示す。 $G_r(x)$ が正の場合は電力過剰，負の場合は電力不足を意味する。図 5-7 から分かるように， $G_r(x)$ の制御が上手くできず，発散してしまっている。

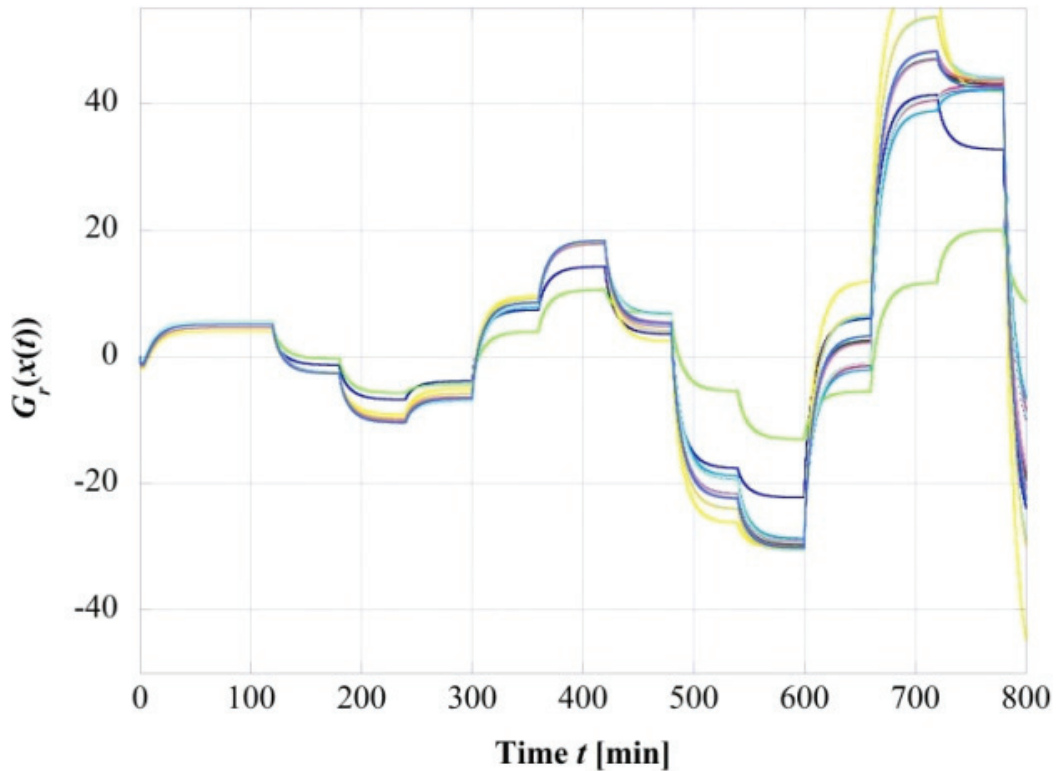


図 5-7 ケース I における需給バランスの時間変化 $G_r(x)$

ケース II では第 4 章で示した逐次線形回帰による需要供給量予測を適用し，予測値 $\hat{x}_{ri}^{[s+1]}$ を用いることで式(11)による価格調整を行った。図 5-8 はケース II における，ある地域における分散的な価格決定の経過を示すグラフであり，横軸は経過時間，縦軸は式 (11) における価格推定のための状態変数 $\theta_{ri}[h]$ の値である。各ステップにおいて各エージェントの状態変数が 30 [min] 程度の時間で同一の値に収束していることが分かる。今回のシミュレーションでは価格更新の間隔を 60 [min] としているので，価格更新に間に合うように次の価格を得ることができている結果となっている。

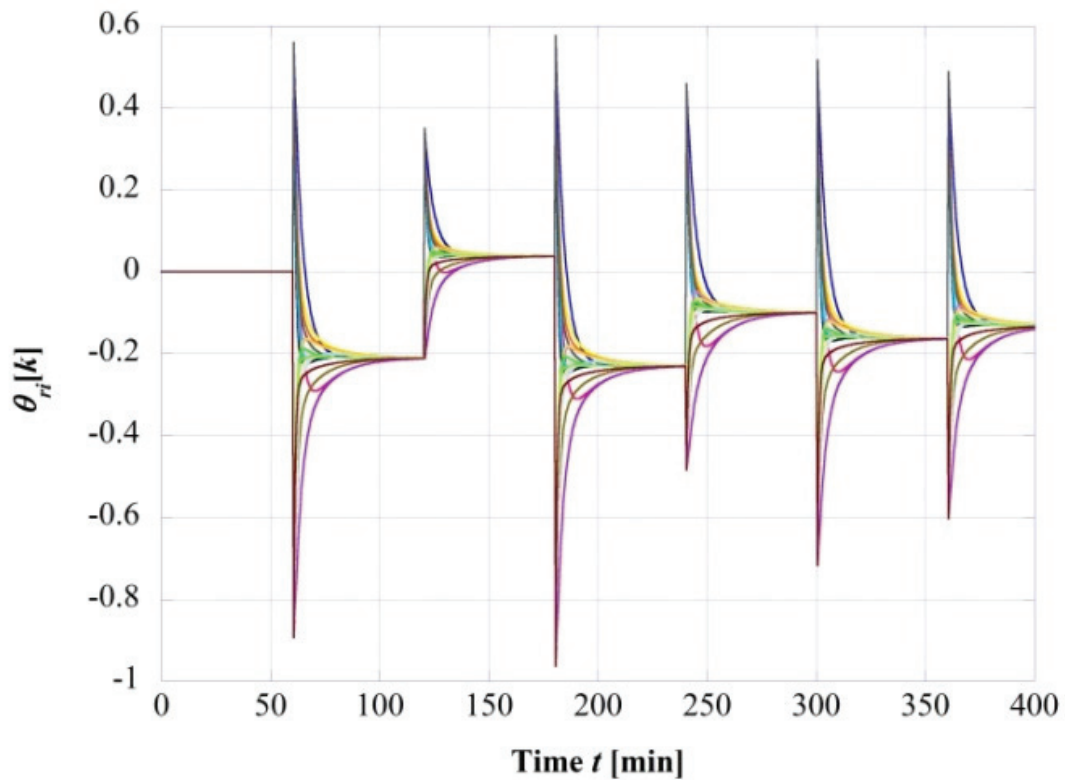


図 5-8 状態変数 $\theta_{ri}[h]$ の時間変化

図 5-9 にはケース II における各地域の制約関数 $G_r(x)$ の時間変化のグラフを示した。ケース I の場合とは異なり、各地域の制約関数 $G_r(x)$ がゼロに収束し、需給バランスがとれた状態を実現できていることが分かる。この結果では、 $G_r(x)$ がゼロ付近に収束するまでには 10 回程度の価格調整が必要となるが、この間はバックアップ電源等を利用することを想定している。

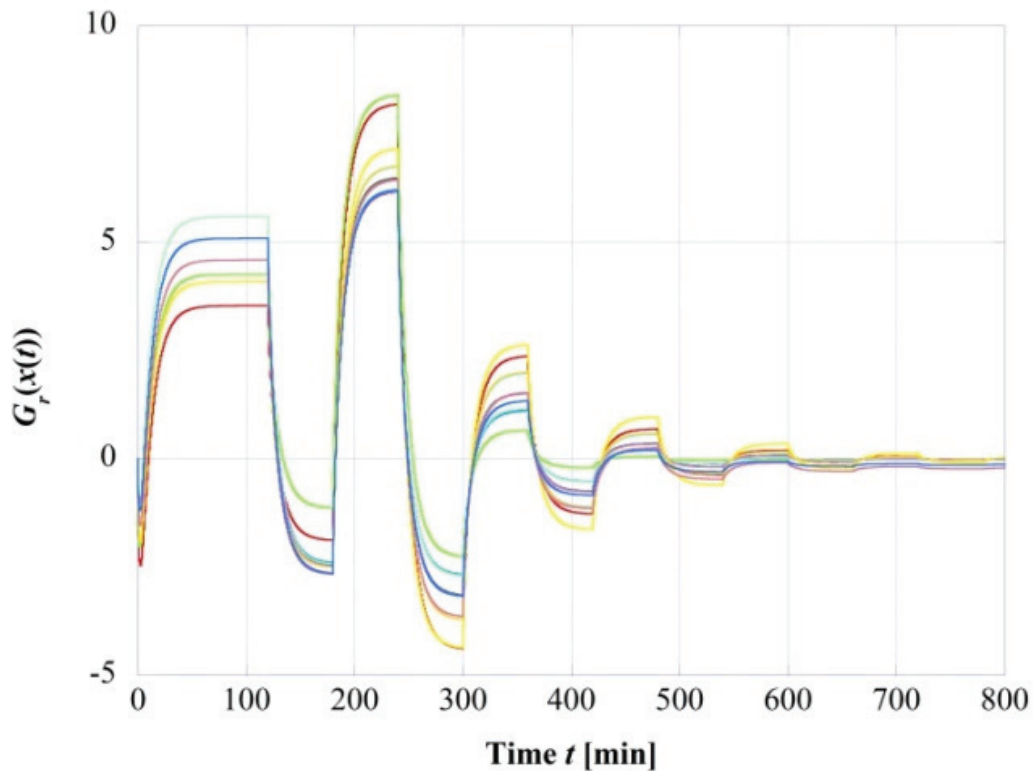


図 5-9 ケースⅡにおける需給バランスの時間変化 $G_r(x)$

5.6. まとめ

本研究ではリアルタイムプライシングの問題を制約付き最適化問題として扱い、分散協調制御の理論に基づく分散型価格調整則を提案している。本稿では、これまでの研究において考慮していなかった情報取得と価格提示のタイミングを考慮し、それにより生じる問題について議論した。そして線形回帰を用いた需要供給量予測の手法を提案し、計算負荷の低い逐次計算に適した形で定式化した。また、様々な規模、条件下における分散型リアルタイムプライシングの振る舞いを調べるために開発した独自のシミュレータについて紹介し、それを用いたシミュレーション結果によって提案手法の有効性を示した。

第6章 初学者によるモデリングへの適応

エージェントベースモデリングや社会シミュレーションに関する専門性を持たないステークホルダとの協働モデリングを可能とする GBSP は、初学者同士や初学者と教員との間での対話にも有用である。そこで、初学者向けの教育プログラムの中に GBSP を用いた活動を導入することで、効果の高い教育を実践することを試みた。本章ではまず6.1節にてこの取り組みにおける GBSP の実践について述べた上で、6.2以降であらためて提案する教育プログラムの内容や教育実践の成果について述べる。

6.1. GBSP の実践

本章で提案する教育プログラムでは、学習者自身がマルチエージェントシミュレーションを構築するという実践的な活動に効果的に取り組むことを可能にするため、GBSP による協働モデリングのプロセスを導入した。GBSP は、学習者同士や学習者と教員間のコミュニケーションにおいて有用であると考えられ、その導入は教育効果の高いプログラムの構築につながると期待できる。

6.1.1. 協働モデリングへの参加者

■ 専門家

エージェントベースモデリングやマルチエージェントシミュレーションに関する専門性を持ち、教育プログラムにおける指導者として協働モデリングに参加した。

■ ステークホルダ（初学者）

教育プログラムに参加する学習者たちは、エージェントベースモデリングやシミュレーション構築に関心を持っているものの専門性は有していないため、ステークホルダとして協働モデリングに参加した。

6.1.2. GBSP の実践結果

本章で提案する教育プログラムの内容のうち、学習者のグループが自分たち自身でマルチエージェントシミュレーションの構築に取り組む活動の中で、GBSP による協働モデリングを実践した。図 6-1 の左側に、ある学習者グループが GBSP に取り組んでいる様子を示す。このグループは林業をテーマとしたマルチエージェントシミュレーションの構築に取り組んでおり、図 6-1 の右側には学習者らが構築したシミュレーションの画面を示し

ている。GBSP で使用している各種トークンが、シミュレーションにおけるエージェントなどに対応しており、図 6-1 には主要要素の対応についても示した。この他、図 6-5 にも学習者による GBSP の様子を示している。

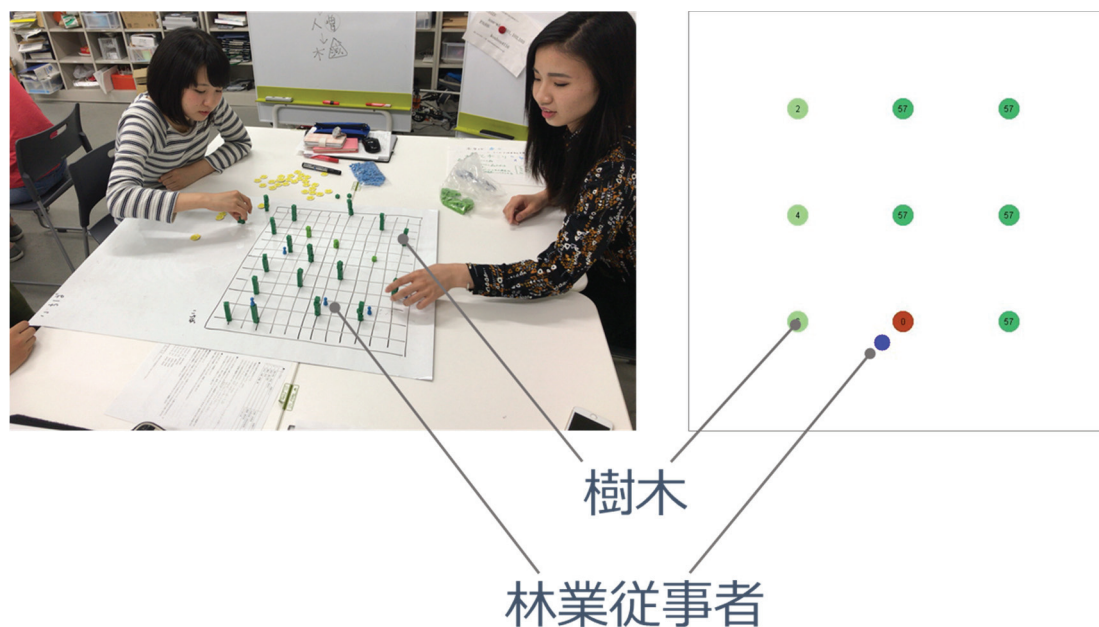


図 6-1 学習者による GBSP の様子 (左) と構築したシミュレーション (右) および要素の対応関係

6.1.3. GBSP の効果

本章で提案する教育プログラムの特徴は、学習者自身がマルチエージェントシミュレーションを構築するという実践的な活動に主体的に取り組むということであり、これにより高い教育効果を得られると期待できる。しかし、教育プログラムの中で確保できる時間や期間は限られており、モデリングやシミュレーション構築は素早く効率的に行わなければならない。専門性を持たない初学者が限られた時間の中でモデリングとシミュレーション構築を的確に進めるのは一般的には困難であるが、今回の提案では GBSP を導入することでそのようなことが可能となった。また、GBSP によって学習者グループの議論がゲームとして可視化されることで、教員が学習者の考えや活動状況を把握することが容易となり、的確な指導を行うことができた。このようにして、GBSP を活用することで、効果と独自性の高い教育プログラムを実現することができた。次節以降では、構築した教育プログラムに関して、GBSP 導入以外の工夫も含めて全体像を述べる。

6.2. 背景・目的

複雑系科学の考え方は、従来の科学がベースとしてきた要素還元的なパラダイムに対して、要素間の関係性や全体としての振る舞いに目を向けた新しいパラダイムであり、ますます複雑化する現代社会および未来社会の諸問題を扱うにあたって、極めて重要な考え方である。複雑系は社会、経済、生命、環境など実に幅広い領域の事象において現れ[88]、複雑系科学の考え方はそれらを理解するための基盤的なアプローチとなる。そこで著者は、複雑系科学の考え方が専門分野を問わず幅広い領域の学生にとって非常に重要な教養になると考え、その基礎的な視点と手法を身につけることを目的とした入門授業を企画し、主に大学初年次生を対象とした教養教育として実践している。

複雑系科学はネットワーク理論、ゲーム理論、進化計算など広範で高度な研究手法を含むが、本稿で紹介する入門授業ではマルチエージェントシミュレーションに焦点を当てた。複雑系の最も基本的で重要な性質は「創発性」と呼ばれ[8]、部分の単純な挙動が全体の高度な秩序を生み出すプロセスを有することである。部分の単純な挙動を自然な形でモデリングし、全体の高度な秩序をシミュレートするエージェントベースモデリングとマルチエージェントシミュレーションは、複雑系を理解するための最も基盤的な手法の一つであると言える。そこで、本章で述べる複雑系科学入門授業では、学習者が主体的に複雑系現象をモデリングし、マルチエージェントシミュレーションを構築するという実践的な活動に取り組む中で、複雑系の基盤的な考え方や視点を習得し、創発性について理解を深めることを目指した。一方で、教養教育の対象者は複雑系科学やシミュレーションに関する基礎的な知識を持たない初学者であり、上記のような授業を実現するためには、初学者でも自身の力でシミュレーションを構築することができるよう、授業設計や指導方法を工夫する必要がある。この課題に対して著者の取り組みでは、初学者によるマルチエージェントシミュレーション構築に適した開発プラットフォームである `artisoc`[91]を活用するとともに、独自のチュートリアル教材やモデリング手法を開発し、授業の中に取り入れている。

6.3. 授業設計

本節では著者が実践している複雑系科学入門授業「複雑な社会をシミュレーションする」の授業設計について述べる。「複雑な社会をシミュレーションする」は鳥取大学における全学共通科目（教養科目）として開講しており、課程規則上は全学年の学生が受講することができるが、開講時限と科目区分の都合により実際の受講生はほとんどが初年次生であ

る。そこで授業設計にあたっては、複雑系科学やプログラミングに関する基礎知識を全く持たない初学者が対象であることを想定し、ディスカッションや演習を通して徐々に複雑系科学の考え方やマルチエージェントシミュレーション構築の方法を修得できるような構成とすることを心がけた。本稿ではこれ以降、特に断らない場合は「授業」という記述は教養科目「複雑な社会をシミュレーションする」のことを指し示すものとする。

6.3.1. 達成目標

授業の目的である「複雑系科学の基本的な考え方、視点とマルチエージェントシミュレーション構築の基礎的な方法を知ること」に対応して、下記のような達成目標を設定した。

1. 「複雑系」や「創発現象」等の基礎的な用語について、自分の言葉で説明できるようになること
2. 複雑系科学を学ぶ意義や応用可能性について自分の言葉で説明できるようになること
3. エージェントベースモデリングのプロセスと、それに基づくシミュレーションを構築する基礎的な方法を修得すること

表 6-1 授業のスケジュール

週	活動内容
1	ガイダンス
2	ディスカッション①：複雑系とは何か？
3	シミュレーション構築チュートリアル①
4	ディスカッション②：様々な創発現象
5	シミュレーション構築チュートリアル②
6	ディスカッション③：複雑系科学の応用例
7	シミュレーション構築チュートリアル③
8	テーマ設定
9~15	グループ活動によるシミュレーション構築
16	プレゼンテーション

6.3.2. 授業の流れ

この授業は2単位の科目であり、1週あたり1コマ、計16週で構成される。授業の全体スケジュールは表6-1のようになっており、大きく前半（1週目~7週目）と後半（8週目~16週目）に分けることができる。

授業の前半では、受講生たちは複雑系科学の考え方をディスカッション形式で学ぶ活動と MAS 構築の基礎的な方法をチュートリアル形式で学ぶ活動に取り組む。ディスカッションの週とチュートリアルの週を交互に配置し、ディスカッションにおける資料作成課題とチュートリアルにおけるプログラミング課題のそれぞれについて 2 週間ずつの取り組み期間を持てるようにした。シミュレーション構築のチュートリアルでは、シンプルながらも創発現象を分かりやすく観察することのできる「サメー小魚モデル」を教材とし、モデリングからコーディングまで、マルチエージェントシミュレーションを構築するプロセスを体験的に学ぶ。これについては 6.4 節で詳しく紹介する。

授業の後半では受講生が 3~4 名ずつのグループに分かれ、受講生自身が設定したテーマに基づくシミュレーション構築を実践する。初学者である受講生たちが 8 週間程度の比較的短い期間でエージェントベースモデリングとプログラミングを含むシミュレーション構築を確実に実現できるよう、独自の協働型モデリングプロセスである GBSP や、クラウド型システムモデリングツールを用いた指導などを導入している。これらの工夫については 6.6 節および 6.8 節において紹介する。授業の最終週には、各グループが構築したマルチエージェントシミュレーションについてプレゼンテーションを行う機会を設けている。

さらに希望者がいる場合、授業終了後にもシミュレーションを改良する活動に取り組み、毎年 3 月に開催される MAS コンペに参加することを目指す。平成 26 年度の授業では授業終了後も継続して活動に取り組んだグループが第 15 回 MAS コンペに参加し、優秀賞を受賞した[89]。この活動や授業の様子については[90]の文献で報告している。

6.4. マルチエージェントシミュレーションのチュートリアル

6.4.1. 開発プラットフォーム

教養科目「複雑な社会をシミュレーションする」では、シミュレーションを構築するプラットフォームとして artisoc を用いている。artisoc は株式会社構造計画研究所が公開しているマルチエージェントシミュレーション開発プラットフォームであり、教育機関に対しては無償で提供されている[91]。グラフィカルでユーザーフレンドリな操作性を特徴とし、プログラミングに不慣れな場合でも素早くマルチエージェントシミュレーションを構築することができる。また、日本語で書かれた解説書[87]やマニュアルが豊富であり、初学者がマルチエージェント構築に取り組む場合に適しているプラットフォームである。

6.4.2. サメ-小魚モデル

0 節で述べたように、授業の前半では受講生たちはシミュレーションを構築するプロセスをチュートリアル形式で学ぶ。この際、例題として「サメ-小魚モデル」という独自の教材を活用している。サメ-小魚モデルは非常にシンプルなルールで記述することができ、マルチエージェントシミュレーションの構築が容易なモデルである。その一方で、サメ-小魚モデルのシミュレーションを実行することで、個々のエージェントの行動ルールからは予測することのできない全体的な振る舞い、すなわち創発現象を分かりやすく観察することができる。学習者各自が `artisoc` を用いてサメ-小魚モデルのシミュレーションを構築する活動に取り組むことで、マルチエージェントシミュレーションを構築する流れと基本的な手法、そして創発現象について効果的に学ぶことができる。

本節ではサメ-小魚モデルについて、エージェントの行動ルールとシミュレーションの実行結果について述べる。詳細については[93]の文献にまとめている他、本稿の付録にソースコードとモデル構築方法の資料を示す。

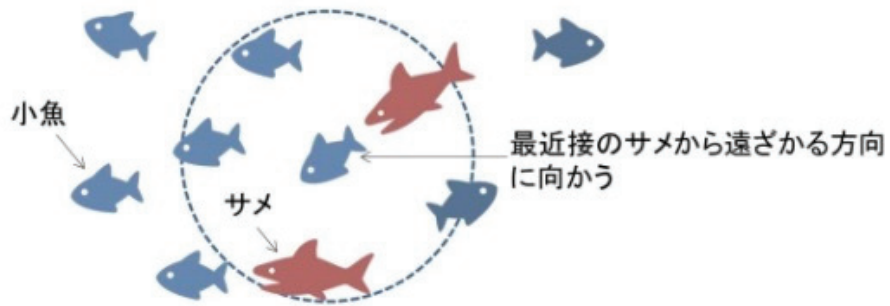
6.5. 行動ルール

サメ-小魚モデルではサメと小魚の 2 種類のエージェントにそれぞれ視野が設定されており、各エージェントは下記の(1)~(4)の行動ルールに従って動く。

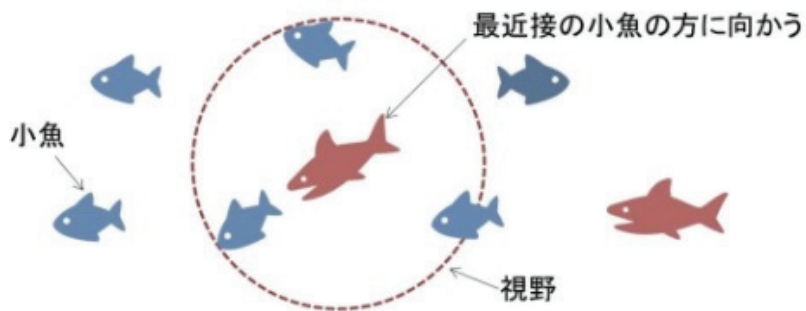
- (1) 小魚は視野内にサメがいる場合、最も近くにいるサメから逃げる（サメとは反対方向を向く）。
- (2) サメは、視野内の最も近くにいるエージェントが小魚の場合に、その小魚を追いかける（小魚の方向を向く）。
- (3) サメは、視野内の最も近くにいるエージェントがサメの場合に、そのサメから逃げる（反対方向を向く）。
- (4) 上記の(1)~(3)に当てはまらない場合は向きを変えずに進む。

上記のルールを図示したものを図 6-2 に示す。

(1) 小魚から見て、視野内にサメがいる場合



(2) サメから見て、視野内の最近接エージェントが小魚だった場合



(3) サメから見て、視野内の最近接エージェントがサメだった場合

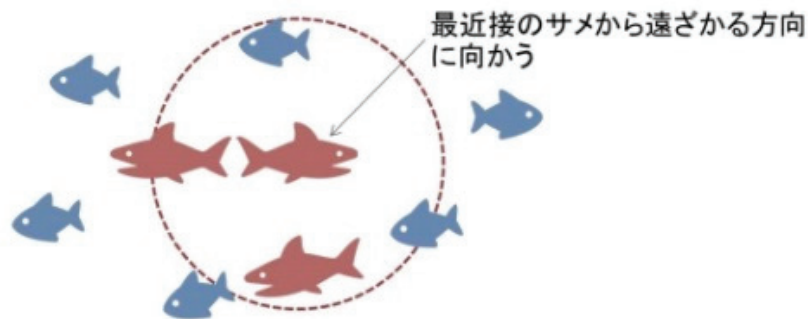


図 6-2 サメー小魚モデルの行動ルール

6.6. シミュレーション結果

artisoc を用いて構築したサメー小魚モデルのシミュレーションについて、エージェントの数や視野の大きさを変化させながら実行した結果のキャプチャを図 6-3 の(a)~(c)に示す。図中ではサメは赤、小魚は青のマークで表されている。図 6-3 (a)はサメの数を 10、小魚の数を 500、両者の視野の大きさを 10 に設定した場合である。この状態から、小魚の視野を大きくして 15 に設定すると、図 6-3 (b) のように小魚たちが特定の場所に集まって塊になる。視野が広がった小魚たちから見ると、前後左右どこを向いてもサメがいると

いう状況になり、サメに囲まれて同一の場所に凝縮するように集まる。次にエージェント数のバランスを大きく変えて、サメの数を 100、小魚の数を 3 に設定する。すると、図 6-3 (c)のようにサメが規則正しく結晶のように並ぶ。これは、サメ同士は避け合うようなモデルになっているので、環境中にサメがたくさんいると、個々のサメは他のサメに囲まれてしまって身動きできなくなるためである。

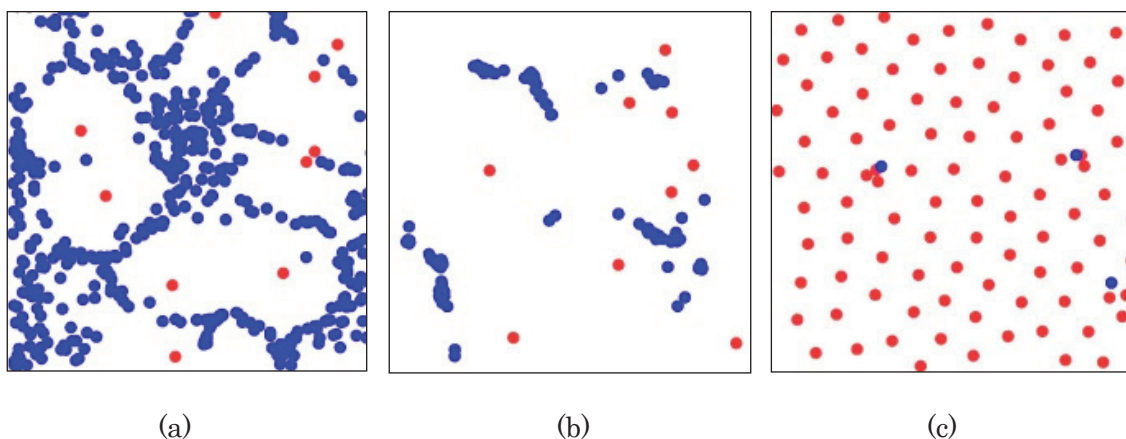


図 6-3 サメ-小魚モデルの MAS 実行結果

サメや小魚の行動ルールには「一箇所に集まる」や「規則正しく並ぶ」といった命令は入っていない。それにもかかわらず、条件によっては図 6-3 (b)のように小魚が特定の場所に集まったり、図 6-3 (c)のようにサメが規則正しく整列したりする。これらは、個々の構成要素の性質や振る舞いからは直接導き出すことのできない全体的な振る舞いであり、創発現象の分かりやすい例となっている。

6.7. モデリングプロセス

0 節で述べたように、授業の後半では少人数グループに分かれた受講生が社会現象のエージェントベースモデリングとマルチエージェントシミュレーションの構築に主体的に取り組む。この授業の主な対象は人文・社会系学部を含む初年次生であり、そのような初学者でもモデリングに関する議論やモデルの記述を効果的に行うことができるよう、モデリングの進め方や指導方法にいくつかの工夫を加えた。本節では指導上の主な工夫点をいくつか取り上げる。

6.7.1. 協働型モデリング手法 GBSP の導入

受講生グループによる社会現象のモデリングにおいて、協働型モデリング手法である GBSP (Game-Based Situation Prototyping) を導入した。GBSP を用いて「協働でゲー

ムをつくる」という形をとることで、モデリングやシミュレーションに対するハードルを下げ、専門知識を持たない非専門家や初学者でも協働でモデリングを行うことができる。また、アウトプットがゲームのプロトタイプであり、ルールが明示されていることから、シミュレーションコーディングやフローチャート記述への接続性が高いというメリットも得られる。本稿で紹介する授業では、GBSPを用いることで、初学者で構成される受講生グループが協働で効果的なエージェントベースモデリングを実現することができる。

6.8. クラウド型システムモデリングツールの活用

授業におけるシミュレーション構築の活動では、GBSPを用いた協働モデリングと artisoc におけるプログラミング作業の間に、ダイアグラムモデル（システムモデル）を介したシミュレーション設計のプロセスを取り入れた。これは、シミュレーションを構成する要素とシミュレーションにおける処理の流れの詳細を、それぞれストラクチャーダイアグラムとフローチャートを用いて整理し、確認するというプロセスである。ここで、ストラクチャーダイアグラムは、シミュレーションを構成するエージェント、変数、環境、相互作用などの要素の階層関係をツリー図で表現したものである。フローチャートはソフトウェア設計において一般的に用いられるものであり、シミュレータの設計においてはシミュレーション全体の処理の流れを示したものと、各エージェントの行動に対応する処理の流れを示したものをそれぞれ記述するようにした。授業では、これらのダイアグラムを構築するプラットフォームとして、著者らが開発したシステムモデリングツール BALUS (図 6-4) を活用した[96][105]。

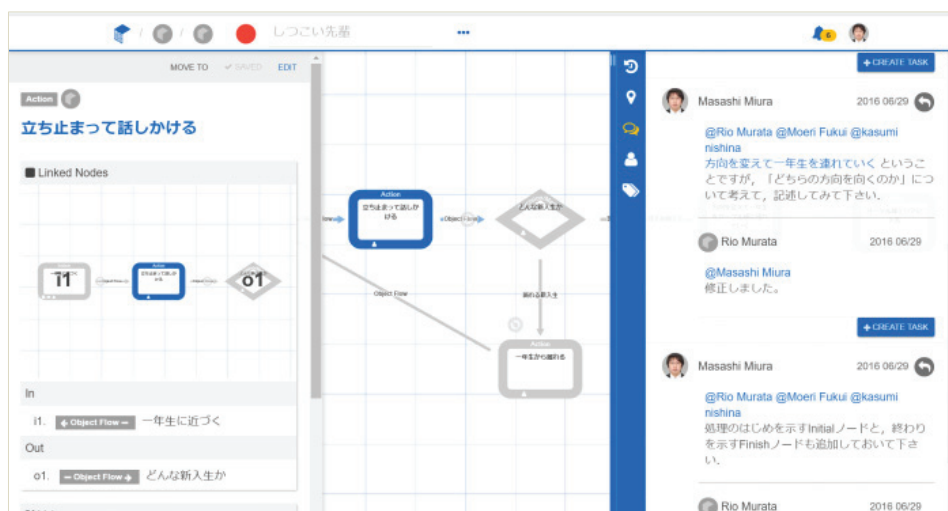


図 6-4 BALUS の画面キャプチャ

BALUS はロボットなどの複雑なシステムの設計を支援するクラウド型のツールであり、次のような機能・特徴を持っている[106].

- WEB ブラウザ上で各種ダイアグラムを素早く構築し、チームメンバに共有することができる
- ダイアグラムの要素に紐付けたコメントを投稿することができ、チームメンバがコミュニケーションしながらダイアグラムを構築することができる
- 更新履歴の保存、更新差分の表示、バージョン管理機能など、チームによる非同期なダイアグラム構築をサポートしている

これらの機能を活用することで、受講生同士または受講生と教員の間でダイアグラムモデルを介した円滑なコミュニケーションが可能となり、シミュレーションの構成や処理の流れに関する細かい議論、指導、アドバイスを伝えることができる。

6.9. 実践結果

本節では H28 年度における授業の様子とアンケート評価の結果について紹介する。H28 年度の受講生は初年次生 10 名であり、5 名が工学部、3 名が地域学部、2 名が医学部という構成であった。

6.10. 授業の様子

ここでは 0 節で示した全体スケジュールのうち、後半（8 週目～16 週目）の受講生主体によるシミュレーション構築活動の様子について写真を交えて簡単に紹介する。

シミュレーション構築に向けた最初の活動は、どのような社会現象に着目してエージェントベースモデリング、テーマを設定することである。H28 年度の授業では、3 つのグループがそれぞれ下記のようなテーマを選んだ。

- 入学シーズンにおけるサークルの勧誘活動
- 遊園地におけるアトラクションの待ち行列
- 戦国時代における合戦

次に、6.7.1 節で紹介した GBSP の手法を用いて、着目する現象について協働でモデリングを行った。GBSP を用いることで、モデリングに関する議論がスムーズかつ活発に行われ、グループメンバー間で合意を得ながらエージェントベースなモデルについて考えることができた。受講生による GBSP の様子を図 6-6 に示す。



図 6-5 受講生による GBSP の様子

GBSP を用いたモデリングの後は、6.8 節で紹介したクラウド型システムモデリングツール BALUS を用いて、シミュレーションの構成要素とその関係性および詳細な処理の流れについて、ダイアグラムモデルを用いて整理した。BALUS を用いることで、授業時だけでなく、授業外の時間においても非同期でダイアグラムを介したコミュニケーションをとることができ、グループで協力して、または教員からのアドバイスに基づいてシミュレーションの詳細設計を構築していくことができた。図 6-6 に、グループ活動において BALUS を活用している受講生の様子を示す。



図 6-6 BALUS を活用している様子

最後に、ダイアグラムモデルを用いた詳細設計に従って、`artisoc` を用いてシミュレーションを構築する作業に取り組んだ。どの受講生も 6.4.2 で紹介したサメ-小魚モデルを例題としたチュートリアルを通して `artisoc` によるシミュレーション構築の基本的なプロセスを修得していたので、スムーズにシミュレーションを構築することができた。図 6-7 に受講生グループが構築したシミュレーションの出力結果を示す。

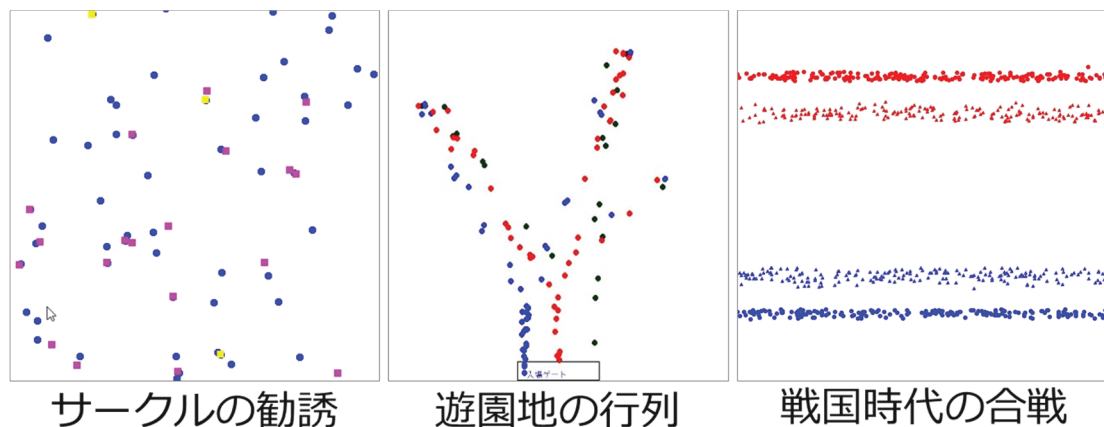


図 6-7 受講生グループが構築した MAS

6.11. アンケート評価

授業の受講前後にプレ調査とポスト調査という形で同一の質問項目によるアンケート調査を行った。アンケートでは表 6-2 に示すような各質問に対してリッカート尺度による選択肢回答と自由記述形式の回答を求めた。ここでは尺度選択肢への回答の集計結果を報告する。

表 6-2 質問項目と尺度

質問 No.1	複雑系とは何か理解していますか？
質問 No.2	複雑系や創発現象の考え方がどんなことに役立つかわかりますか？
質問 No.3	マルチエージェントシミュレーションとは何か、理解していますか？
質問 No.4	エージェントベースモデリングやマルチエージェントシミュレーションなどの考え方がどんなことに役立つかわかりますか？
質問 No.5	マルチエージェントシミュレーションを構築する方法がわかりますか？
尺度 (共通)	A:全く分からない, B:わからない, C:理解している, D:とても良く理解している

表 6-3 にプレ調査とポスト調査における回答の変化についてまとめたものを示す。全ての質問項目において、80%以上の受講生が、理解度が向上する方向への変化を示した。特に No.1 と No.4 の項目については理解度が向上する方向への変化が大きかった。これらの結果から、今回実施した授業が、複雑系およびマルチエージェントシミュレーションに関する理解の向上について効果的であったことが分かる。

表 6-3 プレ調査結果とポスト調査結果

回答尺度の変化	No.1	No.2	No.3	No.4	No.5
ネガティブな反応がポジティブな反応になった(A/B→C/D)	9	4	7	8	6
ネガティブな反応であることは変わらなかったが、理解度が向上した (A→B)	1	4	1	1	2
変化しなかった (A→A など)	0	2	2	1	1
理解度が下がった (B/C/D→A, C/D→B, D→C)	0	0	0	0	1
その他の変化	0	0	0	0	0

6.12. まとめ

本章では、初学者によるマルチエージェントシミュレーションの構築活動を伴う教養教育について、授業設計、独自の教材や指導方法、実践の様子などについて示した。授業の主な対象は複雑系科学やプログラミングに関する基礎知識を持たない初学者であり、限られた時間の中で初学者が主体的にシミュレーションを構築できるようサポートすることが最大の課題であった。本稿で紹介したサマー小魚モデルを用いたチュートリアル、GBSP による協働モデリング、クラウド型システムモデリングツールを用いたダイアグラムモデリングなどは、その課題に対応した工夫の例である。これらの手法を導入することで、初学者が自分自身の力でマルチエージェントモデリングを行うとともにシミュレーションを構築し、実践的に学ぶことのできる機会を創出することができた。

第7章 有効性評価

7.1. 目的

GBSPの協働モデリングのアプローチでは、専門家とステークホルダまたはステークホルダ同士が対話しながら着目する状況に関する認識、知見、意見を共有し、ゲームとしてのモデルを協働で構築していく。これまでのGBSPの実践では研究活動や教育における協働モデリングが状況共有に効果的であることが主観的に確認されている[97]が、その効果や有効性について必ずしも十分な評価ができていない。そこで、本章ではGBSPが持つ側面のうち「状況の共有」に焦点をあて、その有効性評価について論じる。

「状況の共有」に対する有効性を定量的に評価する一般的な既存手法は存在しないため、本研究ではそのような評価を行う方法と指標を開発することを目指す。また、GBSPにおいて行われる対話や行動のうち、どのようなものが状況の共有に有効であるかを抽出するための質的な評価にも取り組む。以上をまとめて、本章で論じる研究の目的として下記の3つを設定する。

1. 協働モデリングにおいて、状況の共有を定量的に評価するための手法を開発する。
2. 開発した手法を用いて、GBSPの有効性を評価する。
3. GBSPにおいてどのような行動が状況の共有に結びついているのかを質的に評価する。

この目的設定のもと、本研究ではGBSPによる状況共有と、別手法による状況共有を比較する評価実験を行った。

7.2. 状況共有に関する評価実験

7.2.1. 問題状況

GBSPの適用が期待される実際の状況は、参加者が異なる価値観や情報を持っている。評価実験では、問題状況を人工的に作成し、被験者には認知できる問題状況の情報が非対称的になるように分配した。すなわち、各被験者には重複する部分を持ちながらもそれぞれに割り当てられた仮想の役割しか知らない事柄を含む情報と設定を配布した。今回作成した人工的な問題状況は地方の住環境や職業環境と人口問題に関するものであり、情報および設定として分配した文章については付録に掲載する。

人工的な問題状況を作成するにあたっては、まず題材とする問題状況の構造を表す因果ループ図を作成し、それを分割および文章化したものを被験者にわたす情報とした。作成し

た因果ループ図を図 7-1 に示す. この図では黒線は正の関係性で, 赤線が負の関係性を表している. さらに, GBSP の適用の目的である「異なる価値観や情報を持った人達の間での状況共有」を考慮し, 被験者には「住民 A」「住民 B」「地方自治体 C」「専門家 D」という役割をもたせることで異なる価値観と情報の非対称性を再現した. 被験者はそれぞれの役割に対応した情報の文章を分配され, グループ内でロールプレイしつつ GBSP および比較手法を用いて各自が持つ情報や設定から導かれる問題状況を共有するという活動に取り組んだ.

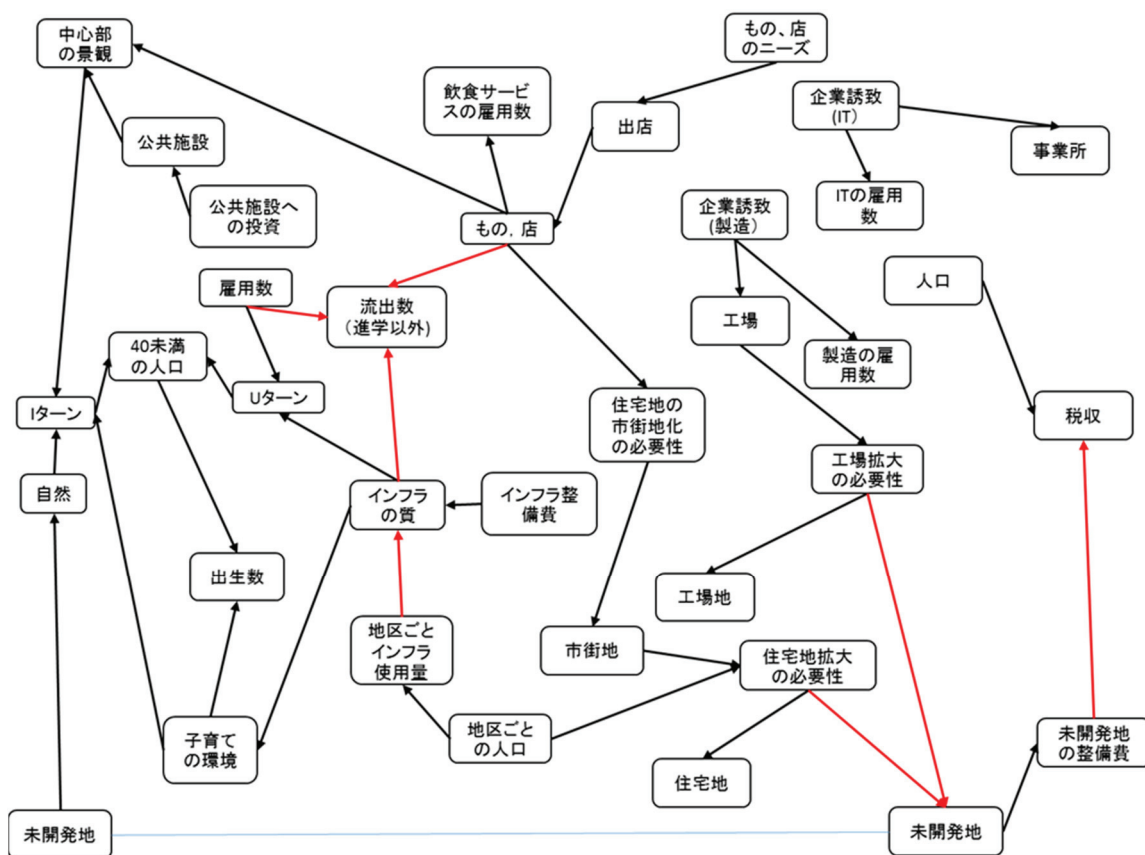


図 7-1 題材の因果ループ図

7.2.2. 比較手法

有効性評価のための実験は, GBSP を用いた実験と別の手法を用いた対照実験からなり, 状況共有の比較を行う. 対象実験は, なるべく自由な討論による状況共有が可能な方法論であることが重要である. 今回は対照実験に用いる手法としてソフトシステム方法論[98][99]におけるリッチピクチャ (以下, 図などでは PR と表記) を取り上げ, 共通の問題状況に対して GBSP を用いて状況共有を行うグループとリッチピクチャを用いて状況共有を行うグ

ループを用意し、共有の結果を比較することで有効性の評価を行う。

ソフトシステム方法論とは、問題状況に対する概念モデル(理想状態)を表現し問題状況の改善案について考える方法論であり、問題状況に対する認識が関与者により異なるときにとくに有効である[100]。ソフトシステム方法論の適用手順である7つのステージのうちで、リッチピクチャは問題状況を記述する第2ステージで利用され、問題状況を自由に表現して共有する手法である[101]。リッチピクチャでは、状況に含まれる関与者や要素とそれらの相互作用をネットワーク図として自由に表現し、参加者間で状況を共有する。今回の実験では記述道具としてホワイトボードシート、ポストイット、ペンを用いて実験を行った。図7-2にリッチピクチャによる状況共有を行った予備実験において作成された図の例を示す。

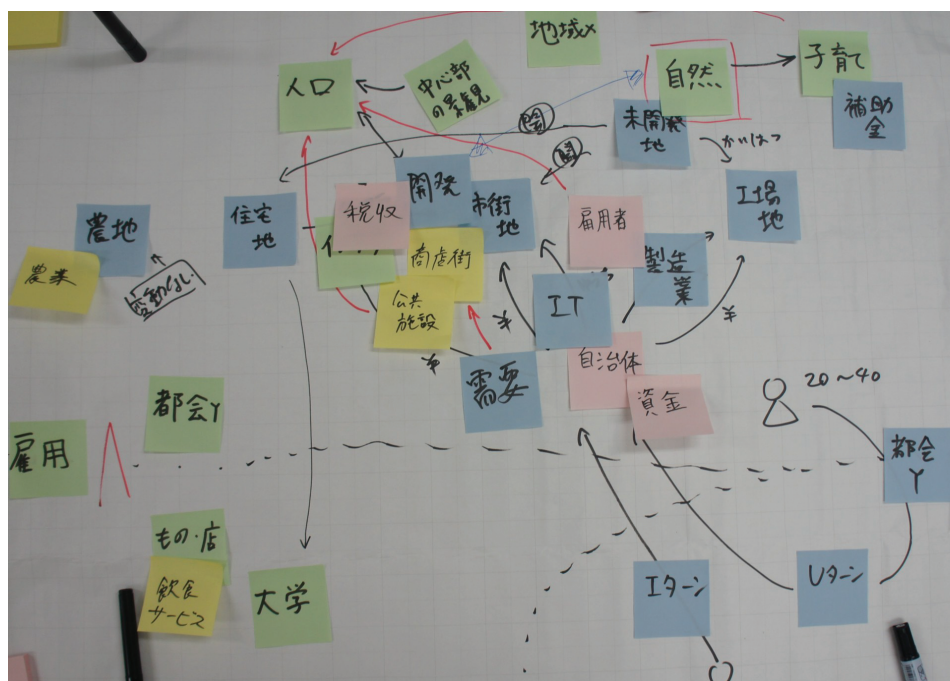


図 7-2 リッチピクチャによる状況共有で作成された図

7.2.3. 実験の流れ

評価実験は大学生と大学院生合計28名を対象として実施した。各グループ4名で7グループを構成し、グループ1~4はGBSP、グループ5~7はリッチピクチャの手法を用いて状況共有を行った。

実験の期間として2日間を設定し、1日目は被験者がそれぞれの手法のプロセスに習熟するためのチュートリアルを行った。チュートリアルではGBSPとリッチピクチャのそれぞれの方法を用いて、現実の問題と人工的に作成した問題の2つの題材に対して状況共有を

実践し、各手法について十分に理解できるようにした。

2 日目の本実験では、人工的に作成した題材「地域 X の人口増減」に関して GBSP とリッチピクチャのそれぞれの方法を用いてグループごとに状況共有を行った。本実験の全体の流れは下記の①から⑤の通りである。

<本実験の流れ>

- ① 情報の読み込み (5 分間)
- ② プレテスト (20 分間)
- ③ 問題状況の共有 (70 分間)
- ④ ポストテスト (80 分間)
- ⑤ ヒアリング (各人 8 分間)

①役割情報の読み込み

各グループの 4 名にはそれぞれ「住民 A」「住民 B」「地方自治体の職員 C」「専門家 D」の役割を割り当てる。被験者にそれぞれの役割に関する情報文章を紙媒体で配布し、配布された情報を読み込んで理解を促す時間をとる。配布した文章は付録に掲載する。

②プレテスト

各手法による問題状況の共有の前に、被験者全員に同じペーパーテストを行う。被験者は自分の役割が持つ情報に従って、状況に関する問題に回答する。プレテストの内容は付録に掲載する。

③問題状況の共有

それぞれのグループが、人工的な題材「地域 X に人口増減」に関する状況共有を行う。いくつかのグループは GBSP が割り当てられ、残りのグループにはリッチピクチャが割り当てられる。各グループはそれぞれ割り当てられた手法を用いて、問題状況に関してグループ内で共有する。質的評価に用いるため、各グループの活動の様子をビデオで撮影する。

④ポストテスト

問題状況の共有を終了した後、被験者全員に同じペーパーテストを行う。被験者は自分の役割が持つ情報や、問題状況の共有において同じグループの他のメンバーから共有された内容に基づいて回答する。プレテストの内容は付録に掲載する。

⑤ヒアリング

各被験者に対して個別に諮問を行い、各グループの成果物（GBSP の場合はゲーム，リッチピクチャの場合はネットワーク図）を用いて説明する．諮問の内容は，人工的な題材「地域 X に人口増減」に関する状況の一部について詳細を問うものである．諮問の内容を付録に掲載する．

7.2.4. 共有された情報の評価

実験後，プレテスト／ポストテストおよびヒアリングの結果を用いて，各被験者がグループにおける状況共有を通して得た情報を評価する．

プレテスト／ポストテストによる評価

プレテストおよびポストテストの結果を用いて，実験による共有の前後における情報の増加量をあらかじめ作成した正解と比較することで算出する．各テストでは「ある要素に関係する要素」や「要素の説明」について問い，あらかじめ用意していた正解と比較する．プレテストによるポストテストへの影響を排除するため，ポストテストではプレテストと独立した問題も出題する．独立した問題とは，題材の問題状況を因果ループ図として表した際に，答えに相当する要素(ノード)が直接結びついていない要素を出題対象とするということである．問題の具体例を以下に示す．

<例1>

問い「中心地の景観はどのような要因で良くなると考えられますか。」

正解「もの，店数の増加」「公共施設数の増加」

例1は，図 7-3 のように題材の問題状況の一部を因果ループ図として構造化した際の要素(ノード)について尋ねている．評価者は，被験者のプレテスト／ポストテストの回答から図 7-3 のような部分的な因果ループ図を描き，予め用意した正答に相当する因果ループ図と比較することで，要素や関係性が正しく共有されているかどうか評価する．

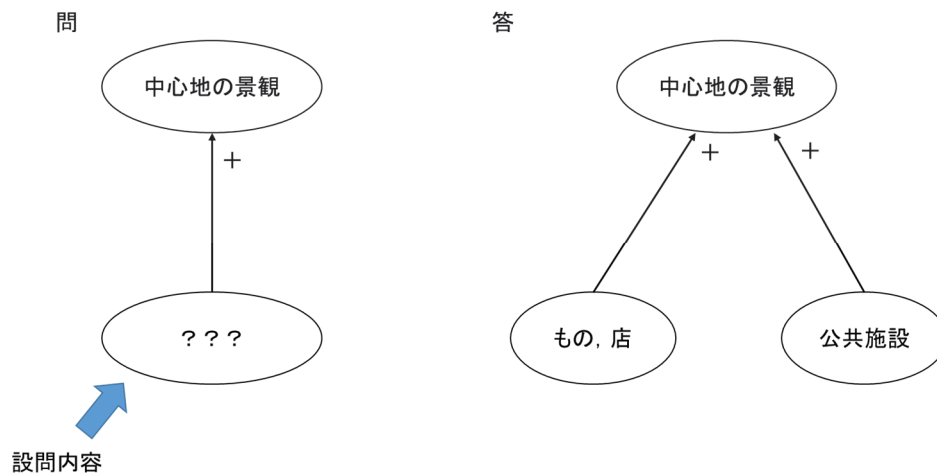


図 7-3 プレテスト/ポストテストの問いと正答に相当する因果ループ図

ヒアリングによる評価

ヒアリングでは、問題状況のある部分について質問を投げかけ、GBSP の成果物としてのゲームまたはリッチピクチャの成果物としてのネットワーク図を用いて、被験者が関連する要素や関係性を説明する。ヒアリングを行うことにより「ある要素や登場人物の行動による、他の要素への影響」についての共有に関して確認することが出来る。ヒアリングの際の問いかけの具体例を以下に示す。

<例 2 >

質問「もの、店のニーズが増加したときの U,I ターン者に与える影響について関連する要素をあげながら説明して下さい」

正解 図 7-4 に示した因果ループ図に相当する内容

プレテスト/ポストテストと同様にヒアリングにおいても、評価者は、被験者の回答に基づいて因果ループ図を描き、正答として用意した因果ループ図との比較によって共有された内容を評価する。

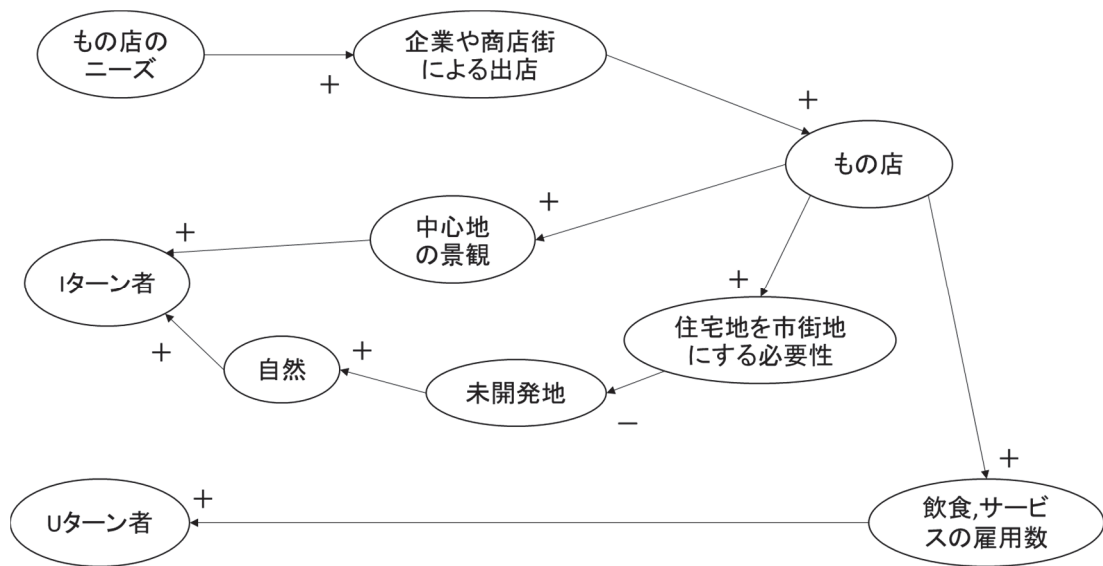


図 7-4 例2における正答に相当する因果ループ図

評価指標

7.1節で述べたように、今回の評価に関する取り組みでは、状況共有の評価を行うための手法を開発することも目的の一つとしている。ここまで述べた実験を設計し、実施した上で、プレテスト、ポストテスト、ヒアリングの結果および録画した状況共有の様子から、共有度、共通認識度などの指標を導出したり、形式概念分析による質的評価の手法を構築したりした。これらについては7.3節および7.4節の中でそれぞれ詳しく述べる。

7.3. 状況共有の定量的評価

7.3.1. プレテスト/ポストテスト結果の点数化

ペーパーテストとして実施したプレテスト/ポストテストの結果を一定の基準で採点し、点数化を行った。点数化にあたってはまず、評価者が被験者の回答を因果ループ図に描き起こした。因果ループ図として描けないような構造のない回答については点数を0とした。被験者の回答に基づいて描いた因果ループ図と、あらかじめ正答として用意した因果ループ図を比較し、一致しているかどうかで点数をつけた。一致している場合は点数1、部分的に一致しているものの関係性や要素の一部に不足のあるものは点数0.5、一致していない場合は点数0とした。ここで正答として用意した因果ループ図は、配布した問題状況を作成する際に構築した因果ループ図の一部を切り取ったものである。

正答との比較の際、単語が正確でなくとも、内容や意図が一致している場合は「一致している」と判断した。例えば、正答の要素が「自治体の整備費」である問題に対して、その要

素に相当する回答として「自治体の補助金」と記述してあった場合などは一致していると判断した。また、関係性に飛躍があったり、要素の一部が抜けていたりする場合は点数 0.5 としているが、この例としては図 7-5 のような場合がある。

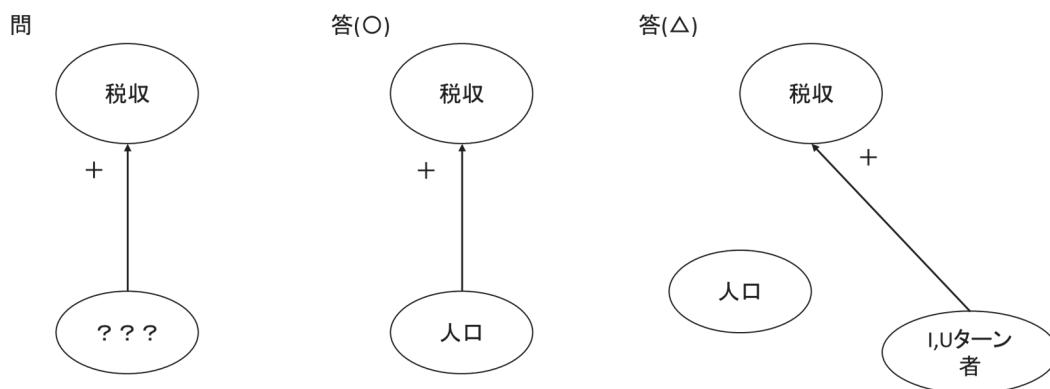


図 7-5 関係性に飛躍のある因果ループ図

図 7-5 の例は「税金を増やす要因はどのようなものがありますか」という問いであり、正答は「人口の増加」である。ここで、作成した問題状況では「I,U ターンの増加」によって「人口の増加」が成り立つとしているので、上の問に「I,U ターンの増加」と回答した場合は関係性の飛躍とみなして点数 0.5 とした。また、正答が複数の要素な場合にその一部しか回答しなかった場合も点数 0.5 とした。

7.3.2. ポストテストの結果

各被験者のポストテストを点数化した結果をおよ表 7-1 および図 7-6 に示す。図 7-6 において青の棒グラフが GBSP のグループ（以下、GBSP 群と表記）、橙色の棒グラフがリッチピクチャ（RP）のグループ（以下、リッチピクチャ群と表記）を表す。なお、ポストテストの満点は 32 点である。

表 7-1 ポストテストの結果（グループ・役割別）

GBSP	グループ 1				グループ 2				グループ 3				グループ 4			
	A	B	C	D	A	B	C	D	A	B	C	D	A	B	C	D
	18.5	19.5	18	20.5	20	19.5	22.5	18	12.5	15.5	21	15	14.5	19.5	18.5	15.5
RP	グループ 5				グループ 6				グループ 7							
	A	B	C	D	A	B	C	D	A	B	C	D				
	16	22.5	16.5	12	24.5	18.5	20	18.5	18	18	16.5	19				

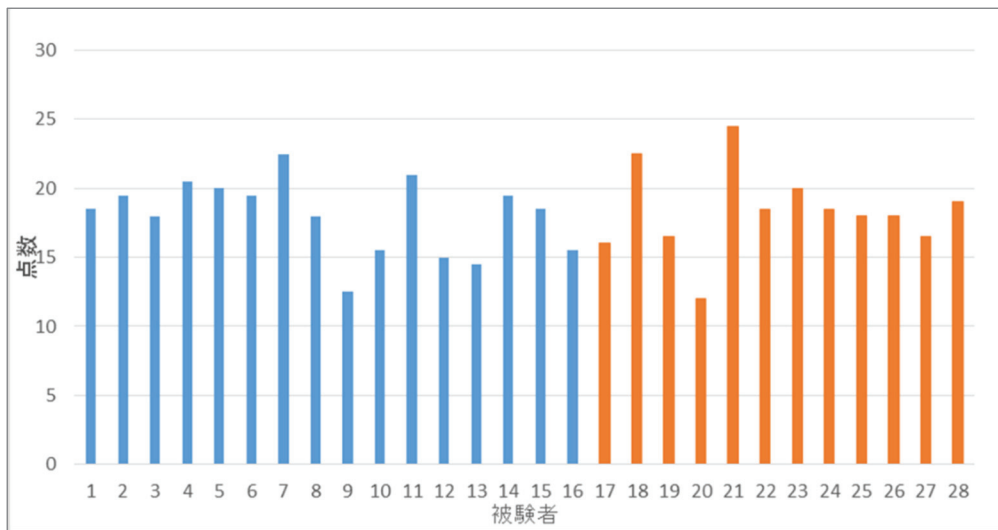


図 7-6 ポストテストの結果

GBSP 群とリッチピクチャ群の 2 群間の平均値における検定は分散が等しくないと仮定した 2 標本による検定(Welch の t 検定)を用いて検定した結果、ポストテストの結果では 2 群間に有意な差はなかった。

7.3.3. プレテスト／ポストテストの比較

ポストテストはプレテストと共通した問題と、プレテストとは独立した内容の問題から構成されている。独立した内容の問題の点数は図 7-7 のようになり、Welch の t 検定の結果、2 つの群の間に有意な差は見られなかった。

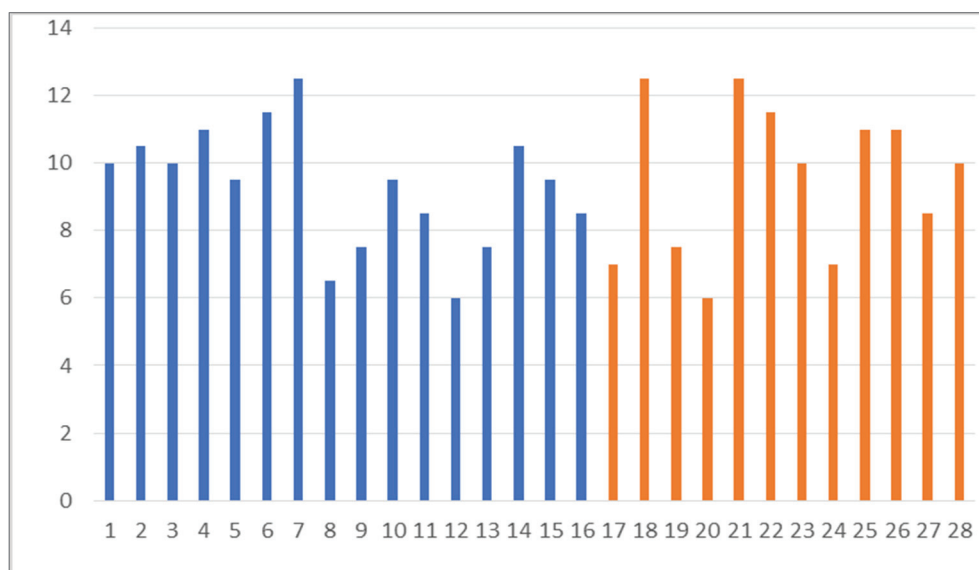


図 7-7 独立した問題の点数

プレテストとポストテストに共通した問題について、点数の差（ポストテストの点数からプレテストの点数を差し引いたもの）は図 7-8 のようになり、Welch の t 検定の結果、2 つの群の間に有意な差は見られなかった。

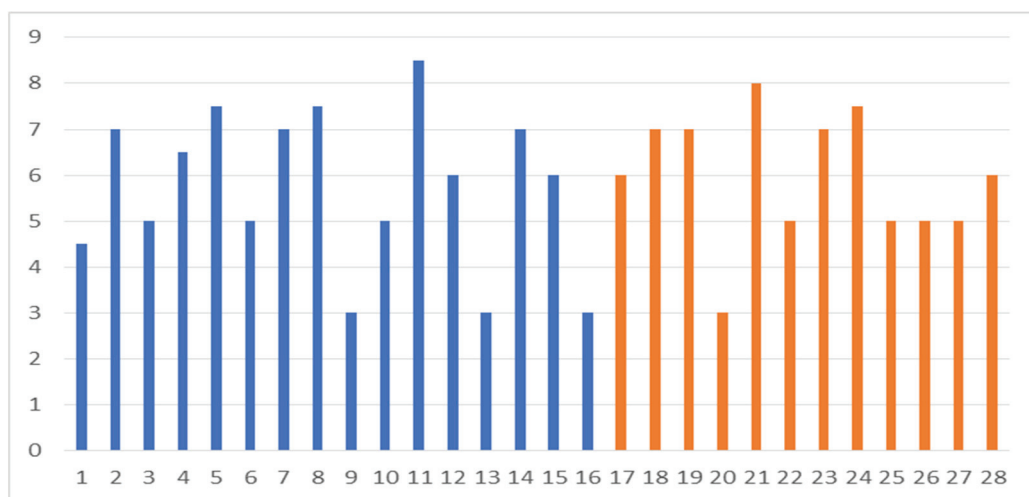


図 7-8 共通の問題における点数差

前節と本節で見たように、要素が正しく共有されたかどうかを個別に評価する点数化の結果に関しては、GBSP 群とリッチピクチャ群の間に有意な差は見られず、GBSP の有効性を示唆する結果を得ることはできなかった。

7.3.4. 共有度による評価

ここまでに見たように、プレテストやポストテストの結果に対して、特定の要素が正しく共有されたかどうかを個別に評価する指標では GBSP 群とリッチピクチャ群の間に有意な差は見られなかった。そこで、各被験者の回答結果を個別に評価するのではなく、各被験者から見て自身の役割情報をグループの他のメンバーにどの程度共有することができたかを評価する指標である「共有度」を考案し、共有度による評価を試みた。

共有度は「自分の役割情報に関する問題について自分が正しく回答しているとき、グループの他のメンバーのうち何人がその問題に対して正しく回答しているか」で算出する定義する。例としてグループ 5 のメンバーの回答を取り上げ、共有度の算出について説明する。ポストテストにおいて「中心部の景観のような要因でよくなると考えられますか」という問に対して、グループ 5 の各メンバーの回答は表 7-2 のようになった。ここで A,B,C,D は各メンバーの役割であり、○は該当の要素について正しく回答できていた場合、×は正しく

回答できていなかった場合を示す。ここで、問題に対する正答となる要素の一つである「公共施設の増加」については、役割 D にのみ配布された情報であるが、リッチピクチャによる共有の結果、ポストテストでは他の役割メンバーA,B,C のいずれもが正しく回答できるようになっているので、共有度は3となる。同様に、「もの・店の増加」については役割 A のみが持っていた情報であり、共有後は A と C のみが正しく回答しているので、共有度は1となる。

表 7-2 共有度の算出の例

問) 景観が良くなる要因	初期の情報保有者	グループ 5				共有度
		A	B	C	D	
答 1) 公共施設の増加	D	○	○	○	○	3
答 2) もの店の増加	A	○	×	○	×	1

このようにして設定した共有度という指標を用いてポストテストの結果を評価したところ、各グループの共有度の平均は表 7-3 のようになった。GBSP 群とリッチピクチャ群におけるこの結果に対して t 検定を行ったところ、有意な差は得られなかった。

表 7-3 各グループの共有度の平均

GBSP	グループ 1	グループ 2	グループ 3	グループ 4
		1.92	1.92	1.38
リッチピクチャ	グループ 5	グループ 6	グループ 7	
	1.37	1.78	1.48	

GBSP では議論が特定の部分に集中して行われており、議論の中で触れられている部分については、グループメンバーの多くが正答できていたため共有度が高くなったと考えられる。一方で、議論において中心的でないところは初期の情報保有者ですら正答できていない場合も見られた。共有度の指標ではこのような場合の結果は数値に反映されない。これに対してリッチピクチャでは議論が全体的であり、どの問においても正答している人とそうでない人のばらつきが大きいいため、共有度が下がったと考えられる。

7.3.5. ヒアリング結果と共通認識度

ヒアリングでは各被験者に対して同じ質問をし、各被験者は共有活動の成果物（GBSP グループの場合はゲーム、リッチピクチャグループの場合はネットワーク図）を用いて状況を部分的に説明する。評価者はヒアリングで得た各被験者の認識について因果ループ図を作成し、グループの他メンバーの結果と比較することでその類似度を評価する。類似度の評価にあたっては、グループ内でどの程度の共通認識が得られていたかを分析するための指標として以下に説明する「共通認識度」を定義する。

ある質問に対して被験者による説明をヒアリングして得られた因果ループ図をグループごとにまとめた結果、全部で p 種類のノード（要素）、 q 種類のエッジ（関係性）が得られたとき、該当のグループにおける該当の質問に対する「要素の共通認識度」と「関係性の共通認識度」を以下のように定義する。

要素の共通認識度

ヒアリングで抽出された因果ループ図に含まれるノードをそれぞれノード $_1$ 、ノード $_2$ 、...、ノード $_p$ としたとき、要素の共通認識度を以下のように定義する。

$$\sum_{i=1}^p \frac{\text{ノード}_i \text{を認識している人数}}{p} \quad (1)$$

関係性の共通認識度

ヒアリングで抽出されたエッジをそれぞれエッジ $_1$ 、エッジ $_2$ 、...、エッジ $_q$ としたとき、その因果ループ図のエッジの共通認識度を以下のように定義する。

$$\sum_{k=1}^q \frac{\text{エッジ}_k \text{を認識している人数}}{q} \quad (2)$$

具体例

共通認識度について、具体例を挙げて説明する。「もの・店のニーズ」が増えたときの「U,Iターナー」への影響について、関連する要素を挙げながら説明せよという問いに対して、グループ1のメンバーによる説明からは図7-9のような因果ループ図が得られた。ここで、各ノードとエッジに付記されている人数は、4人のグループメンバーのうち、そのノードや

エッジを認識しているメンバーの人数である。表 7-4 および表 7-5 に各ノードとエッジを認識しているメンバーの人数をまとめた。ここで「もの・店のニーズ」のノードについては、発問の中に出てくる要素であるため、共通認識度の評価対象からは省く。図 7-9 では 5 種類のノードと 7 種類のノードが抽出されたので、今回の例では $p = 5$, $q = 7$ となる。

これらの結果から、この例における要素と関係性の共通認識度は次のように算出できる。

$$\text{要素の共通認識度} = \frac{2 + 4 + 1 + 3 + 4}{5} = 2.8$$

$$\text{関係性の共通認識度} = \frac{2 + 2 + 2 + 1 + 3 + 1 + 3}{7} = 2.0$$

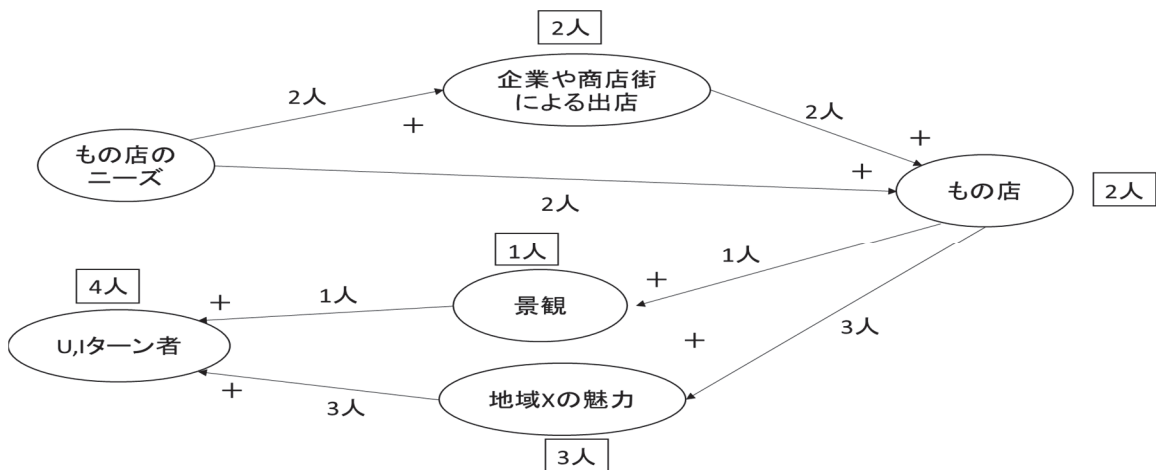


図 7-9 グループ 1 へのヒアリングにおいて抽出された因果ループ図

表 7-4 各ノードの認識人数

ノード	認識している人数
出店数	2 人
もの店	4 人
景観	1 人
魅力	3 人
U, Iターナー者	4 人

表 7-5 各エッジの認識人数

エッジ	認識している人数
もの店ニーズが増すと出店数が増える	2人
出店数が増えるとももの店が増える	2人
もの店ニーズが増えるとももの店が増える	2人
もの店が増えると景観が良くなる	1人
もの店が増えると魅力が上がる	3人
景観が良くなると U,I ターン者数が増加	1人
魅力が上がると U,I ターン者数が増加	3人

7.3.6. 共通認識度による評価

実験では 4 つの質問に関してヒアリングを行い、質問ごとの共通認識度を算出し平均を算出した。各質問に対するの要素と関係性の共通認識度は表 7-6 のとおりになり、グループごとの平均を手法群別に整理したところ表 7-7 のようになった。

表 7-6 各質問に対する共通認識度

グループ	Q1		Q2		Q3		Q4		平均	
	要素	関係性	要素	関係性	要素	関係性	要素	関係性	要素	関係性
1	4.0	4.0	2.0	1.6	4.0	3.0	3.2	2.7	3.3	2.8
2	3.5	3.5	4.0	4.0	2.4	2.5	2.8	2.0	3.2	3.0
3	2.5	2.5	2.8	1.8	3.8	3.3	2.8	2.0	3.0	2.4
4	1.7	1.7	1.5	1.5	3.0	3.0	2.0	2.0	2.0	2.0
5	2.0	1.5	2.3	1.7	1.6	1.3	2.5	2.2	2.1	1.7
6	1.3	1.3	2.5	1.6	3.0	2.9	2.0	1.2	2.1	1.7
7	2.7	2.7	3.0	1.8	2.6	1.9	2.7	2.5	2.7	2.2

表 7-7 グループごとの共通認識度の平均

手法	GBSP				リッチピクチャ		
グループ	1	2	3	4	5	6	7
要素	3.3	3.2	3.0	2.0	2.1	2.1	2.7
関係性	2.8	3.0	2.4	2.0	1.7	1.7	2.2

表 7-7にまとめた結果に対して、t 検定を行ったところ、要素の共通認識度については有意水準 10%で、関係性の共通認識度については有意水準 5%で GBSP を用いた場合の方が高い共通認識度を得られるという結果を得た。この結果より、特に関係性の共通認識度について GBSP を用いた共有が有効であることが分かる。

7.3.7. 定量的評価のまとめ

7.3.1 節から 7.3.6 節に示したことをまとめると、以下のようになる。

- 被験者各自のプレテスト/ポストテストの結果を点数化し、「共有された状況に対して要素や関係性を正しく認識できたか」について評価をした結果、GBSP と対照手法との間で有意な差は見られなかった。
- 「自分の役割情報に関する問題について自分が正しく回答しているとき、グループの他のメンバーのうち何人がその問題に対して正しく回答しているか」という形で「共有度」という指標を定義し、実験結果を評価した。結果としては、GBSP 群と対照群の間に有意な差は見られなかった。
- 「問題状況に関する要素と要素間の関係性について、それぞれグループ内でどの程度の共通認識を得ることができたか」という形で「要素の共通認識度」および「関係性の共通認識度」を定義し、実験結果を評価したところ、どちらの共通認識度においても GBSP が有効であることを示すことができた。特に関係性の共通認識度において高い有効性を持つことが分かった。

7.4. 形式概念分析による質的評価

7.3 節で得られた共通認識度に関する GBSP の有効性がどのような要因から生じているのかを質的に評価するため、形式概念分析[102][103]を用いた分析を行う。本節ではまず形式概念分析の概要を説明し、手法として具体例をあげつつどのように分析していくのかを示しつつ、今回の実験結果に対する適用方法と結果について述べる。

7.4.1. 形式概念分析の方法

形式概念分析は、対象とその属性に関するコンテキスト表を作成し、コンテキスト表において表現される空間において成り立つ含意関係などの関係性を表現する質的分析手法である。示される関係が、名義尺度や順序尺度などに制約されることがなく、社会現象における複雑性や多様性を犠牲にすることなく扱うことができ、分析の手続きが数学的に与えられているため客観的であるという特徴がある。以下では性別と所属している部活動を属性とする簡単な例を用いて形式概念分析の具体的な手順について説明する。

(1) コンテキスト表の作成

形式概念分析を行うためには、まずはコンテキスト表を作成する。ここでは、2つの集合 G と M 及び $G - M$ 間の関係 I からなる形式文脈 $K = (G, M, I)$ において、 G の要素を「対象」、 M の要素を「属性」という。そして「対象 g は属性 m を有する」ことを gIm または $(g, m) \in I$ と表現する。コンテキスト表は各対象と属性について gIm を整理したものであり、図 7-10 のように表現される。ここでは、 \times の印がその属性を有することを意味している。

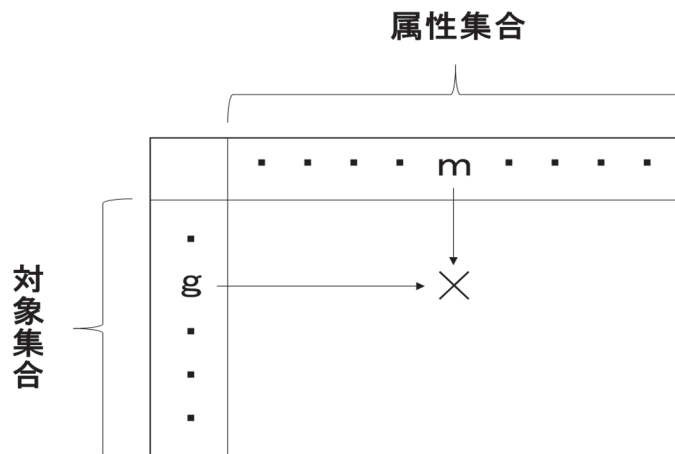


図 7-10 対象 g が属性 m を持つことを表現するコンテキスト表

(2) 形式概念の抽出

$A \subseteq G$ と $B \subseteq M$ に対して対象の集合 A が共通して持つ属性の集合が B であり、属性の B が共通して持つ対象の集合が A であるようなとき、 (A, B) の組を形式概念とする。例えば、ある集団において、対象（個人）とその対象がそれぞれ持つ属性が表 7-8 のようなコンテキスト表で表された場合、表 7-9 のような形式概念が抽出される。

表 7-8 ある集団から得られたコンテキスト表

		属性 M				
		男	女	サッカー部	野球部	手芸部
対象 G	田中($g1$)	×		×		
	鈴木($g2$)	×		×		
	山田($g3$)	×			×	
	佐藤($g4$)		×	×		
	加藤($g5$)		×			×
	木下($g6$)		×			×

表 7-9 得られる形式概念

No.	形式概念
1	({男}, { $g1, g2, g3$ })
2	({女}, { $g4, g5, g6$ })
3	({サッカー部}, { $g1, g2$ })
4	({野球部}, { $g3$ })
5	({手芸部}, { $g5, g6$ })
6	({男, サッカー部}, { $g1, g2$ })
7	({男, 野球部}, { $g3$ })
8	({女, サッカー部}, { $g4$ })
9	({女, 手芸部}, { $g5, g6$ })

(3) 形式概念を用いた Hasse 図の作成

一方の属性集合が他方の属性集合に含まれている場合の属性の種類が少ない方を上位概念、多い方を下位概念とし、最上位概念を ($\{\text{全対象}\}, \{\}$)、最下位概念 ($\{\}, \{\text{全属性}\}$) とし、それぞれの上位概念、下位概念を線で結んだ、全ての形式概念が含まれる図を

Hasse 図として表現する. 表 7-9 で示した形式概念について Hasse 図を描くと図 7-1 1 のようになる. Hasse 図において, 各形式概念はノードで表現される. 図 7-1 1 においてノードに付記した番号は表 7-9 に対応する. Hasse 図では, 下位概念は線で結ばれた上位概念の属性を含んでいることになる. 例えば図 7-1 1 から, 形式概念 8 のノードはノード 2, 3 の属性を含むことが分かる.

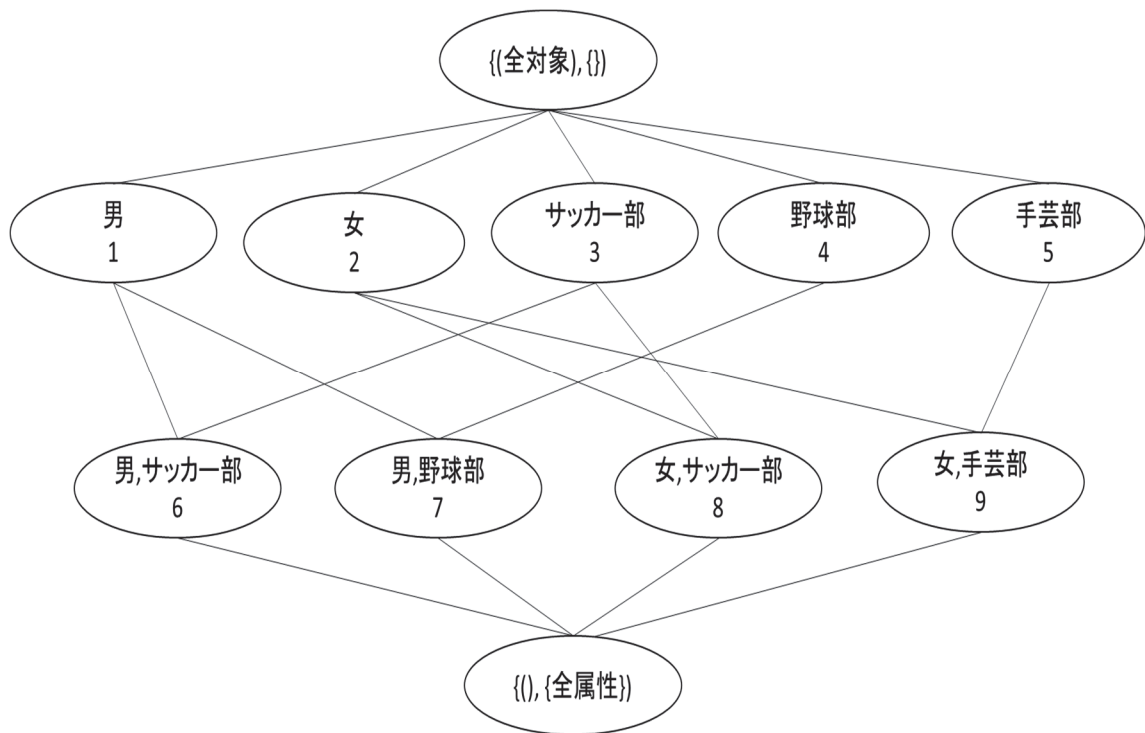


図 7-1 1 Hasse 図の例

(4) アソシエーション分析

アソシエーション分析とは, 多数のデータ内の要素間の関係をルールとして抽出する手法である. A を条件部, B を結論部としたとき, アソシエーションルールは $\{A \Rightarrow B\}$ と表現される. A が成り立つとき同時に B も成り立つ確率を $\{A \Rightarrow B\}$ の確信度と呼び, 本研究では確信度を評価指標の一つとして用いる. 確信度は次のように表される.

$$\{A \Rightarrow B\} \text{ の確信度} = \frac{N(A \cap B)}{N(A)} \quad (3)$$

属性集合 X, Y に対して、 X が成り立つときの Y が成り立つといったアソシエーションルールの確信度は、形式概念のもつ対象数から算出することができ、Hasse図を用いることで対象数を数え上げることができる。アソシエーションルール $\{X \Rightarrow Y\}$ の確信度は次のように表される。

$$\{X \Rightarrow Y\} \text{の確信度} = \frac{\text{属性}X \cap Y \text{の形式概念の対象数}}{\text{属性}X \text{の形式概念の対象数}} \quad (4)$$

例えば、図 7-1 1 において、属性{男},{サッカー部}の関係性を見たい場合、形式概念 1,3,6 に注目すればよい。それぞれの対象数を付記すると図 7-1 2 のようになる。ここで{男}ならば{サッカー部}となる確信度は以下のようになる。

$$\frac{\text{形式概念 6 の対象数}}{\text{形式概念 1 の対象数}} = \frac{2}{3}$$

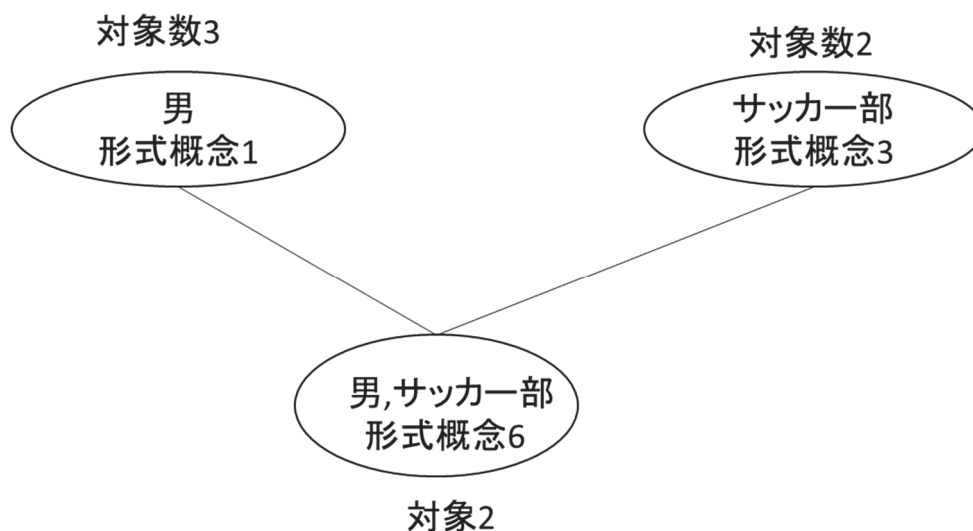


図 7-1 2 形式概念間の対象数の例

7.4.2. 形式概念分析の適用

前節で説明した形式概念分析の手法を本研究の評価実験に適用し、被験者の行動と状況共有の関係を説明することを試みる。

本研究ではGBSPとリッチピクチャの2つの共有方法を比較することでGBSPの有効性を評価している。その結果、関係性の共通認識度という指標においてGBSPが有効であり、

問題状況における要素間の関係性について参加者の中で共通認識を得ることに對して高い効果があることが分かった。ここでは、GBSP におけるどのような場面がそのような結果に結びついたのでかを分析する。

ここでは、形式概念分析における「対象」をヒアリングで抽出された因果ループ図のエッジとし、「属性」を各手法（GBSP とリッチピクチャ）における行動およびその行動の対象に関する共通認識度とする。そのような「対象」と「属性」についてコンテキスト表を作成し、共通認識度を結論部に含むアソシエーションルールとその確信度を得ることで、行動と共有認識度の関係を導く。それぞれの属性は、実験の様子を撮影したビデオデータ（70 分×6 グループ※グループ 4 についてはビデオデータが破損していたため、分析対象からは除いた）から、手法ごとに特徴的な共有に関する行動を選択することで抽出した。

7.4.3. GBSP 群に対する適用

「対象」は GBSP 群の被験者から抽出された因果ループ図のエッジの事象であり、「属性」は表 7-10 に示す通りである。

表 7-10 GBSP 群の実験結果から抽出された属性

属性 1 {発言}	事象が発言されている
属性 2 {トークンなどを用いた議論}	事象の要素がトークンなどで表現され、事象の関係性が議論されている
属性 3 {ゲーム化}	事象がゲームの一部としてテストプレイされている
属性 4 {共通認識度 0~4}	事象のグループ内の共通認識度

それぞれの関係性（エッジ）に対して表 7-10 の属性 1~4 があり、表 7-11 のようなコンテキスト表を作成することができる。エッジは 91 種類抽出されたが、そのうち 1 から 5 と番号付けしたエッジに関するコンテキストを表 7-11 に示している。

表 7-1 1 GBSP における形式概念分析のコンテキスト表

	発言	トークなどを用いた議論	ゲーム化	共通認識度 0	共通認識度 1	共通認識度 2	共通認識度 3	共通認識度 4
エッジ 1	×	×	×					×
エッジ 2	×			×				
エッジ 3	×	×				×		
エッジ 4	×			×				
エッジ 5					×			

例えば、表 7-1 1 において「対象」のエッジ 1 は{発言}, {トークンなどを用いた議論}, {ゲーム化}, {共通認識度 4}の「属性」を持っている。このことは、エッジ 1 の事象は関係性の発言がされ、トークンなどを用いた議論とテストプレイがなされており、共通認識度が 4 であることを意味する。

7. 4. 4. リッチピクチャ群に対する適用

「対象」はリッチピクチャ群の被験者から抽出された因果ループ図のエッジの事象であり、「属性」は表 7-1 2 に示す通りである。

表 7-1 2 リッチピクチャ群の実験結果から抽出された属性

属性 1{発言}	事象が発言されている。
属性 2 {ポストイットでの表現○/△/×}	問題状況の因果ループ図において関係性を持つ 2 つの要素が、ともにポストイット記載されていた場合は○、不完全な場合は△、表現がなされていない場合は×とする。
属性 3{関係性の記載○/△/×}	関係性を持つ要素が 2 つともポストイットで記載されていた場合、その間の関係性が共有された内容と同じで正しく記載されていた場合は○、不完全な場合や矢印のみの場合は△、記載がなかった場合を×とする。
属性 4{共通認識度 0~4}	事象のグループ内の共通認識度

それぞれの関係性（エッジ）に対して表 7-1 2の属性 1~4 があり，表 7-1 3のようなコンテキスト表を作成することができる．エッジは 102 種類抽出されたが，そのうち 1 から 5 と番号付けしたエッジに関するコンテキストを表 7-1 3に示している．

表 7-1 3 リッチピクチャにおける形式概念分析のコンテキスト表

	発言	ポストイットでの表現○	ポストイットでの表現△	ポストイットでの表現×	関係性の記憶○	関係性の記憶△	関係性の記憶×	共通認識度0	共通認識度1	共通認識度2	共通認識度3	共通認識度4
エッジ1	×	×				×					×	
エッジ2				×			×	×				
エッジ3				×			×		×			
エッジ4				×			×		×			
エッジ5				×			×		×			

例えば，表 7-1 3において「対象」のエッジ 1 は{発言}，{ポストイットでの表現○}，{関係性の記載△}，{共通認識度 3}の「属性」を持っている．このことは，エッジ 1 の事象は，関係性の発言がされ，エッジが結んでいる要素 2 つがポストイットでともに表現されているがその要素の関係性の記載が不完全であり，共通認識度が 3 であることを意味する．

7. 4. 5. GBSP 群の分析結果

分析に際しては形式概念分析の支援ツールである Concept Explorer[103]を使用し，各エッジに対する行動と共通認識度を属性とするコンテキスト表と Hasse 図からアソシエーションルールを抽出するとともに確信度を計算した．

GBSP 群について，コンテキスト表から得られたすべての形式概念を含む Hasse 図を図 7-1 3に示す．ノードに記載した数字の左側は対象の数を表し，右側のパーセンテージは全体の対象のうちそのノードが保持する対象数の割合を表す．Hasse 図のうち，共有に強く関わると考えられる{発言}，{トークンを用いた議論}，{ゲーム化}，{共通認識度 4}の属性を含む部分を抽出したものを図 7-1 4に示す．

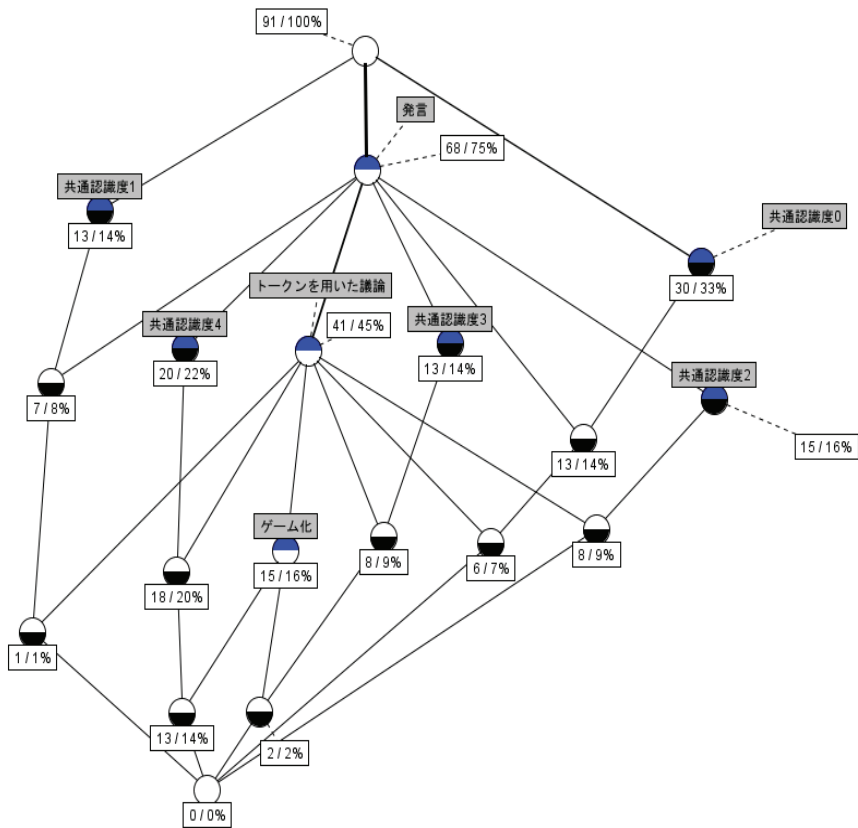


図 7-1 3 GBSP 群の形式概念分析により得られた Hasse 図 (全体)

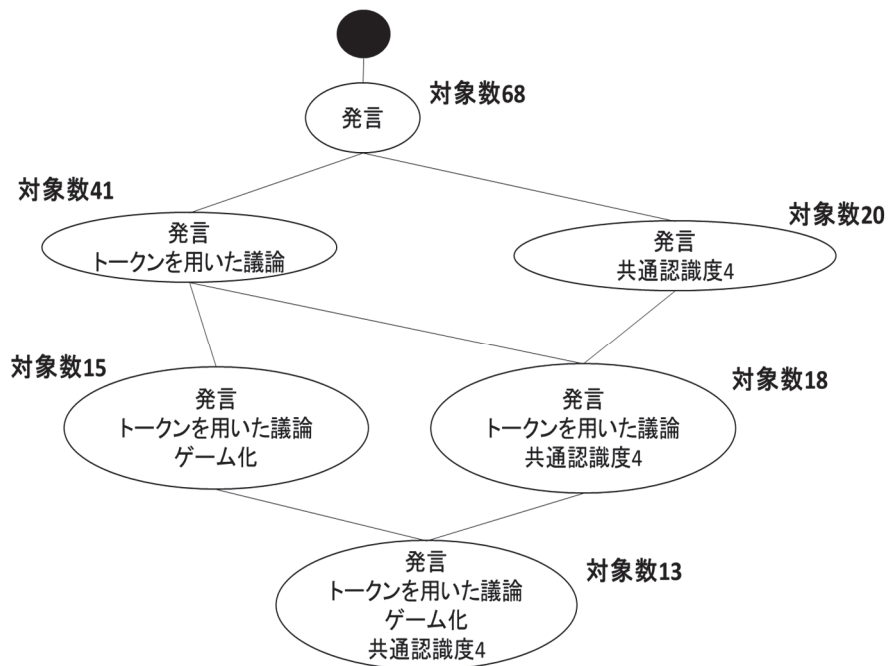


図 7-1 4 GBSP 群の形式概念分析により得られた Hasse 図 (重要な部分)

図 7-1 4 の Hasse 図から表 7-1 4 に示すアソシエーションルールと確信度が得られた。この表で、属性Xの行動が行われたり共通認識度が得られたりした場合に、属性Yの行動や共通認識度が得られる可能性が確信度である。

表 7-1 4 GBSP 群で得られたアソシエーションルールと確信度（重要な部分）

No.	属性Xを有する対象数	属性X	確信度	属性Yを有する対象数	属性Y
1	41	{トークンを用いた議論}	100%	41	{発言}
2	20	{共通認識度 4}	100%	20	{発言}
3	20	{発言} {共通認識度 4}	90%	18	{トークンを用いた議論}
4	15	{ゲーム化}	100%	15	{発言} {トークンを用いた議論}
5	15	{発言} {トークンを用いた議論} {ゲーム化}	87%	13	{共通認識度 4}

表 7-1 4 の No.5 のアソシエーションルールの結果に着目すると、{発言}、{トークンを用いた議論}、{ゲーム化}の行動が行われた対象（エッジ）のうち 87%が共通認識度 4 で共有されたことが分かる。このことから、ゲーム化してトークンを用いた議論を行うという、GBSP の最も基本的な行動が高い共通認識度の要因となっていることが分かった。

7.4.6. リッチピクチャ群の分析結果

リッチピクチャ群について、コンテキスト表から得られたすべての形式概念を含む Hasse 図を図 7-1 5 に示す。図 7-1 5 はノード数が多くて複雑になってしまったので、このうち、共有に強く関わりと考えられる{発言}、{ポストイットでの表現○}、{関係性の記載○}、{共通認識度 4}の属性を含む部分を抽出したものを図 7-1 6 に示す。さらに図 7-1 6 の Hasse 図から得られるアソシエーションルールと確信度を表 7-1 5 にまとめた。表 7-1 5 の No.8 のアソシエーションルールの結果に着目すると、{発言}、{ポストイットでの表現○}、{関係性の記載○}の行動が行われた対象のうち 4%だけが共通認識度 4 で共有されたこ

とが分かる。よって、GBSP の基本的な行動であるゲーム化やトークンを用いた議論に対して、リッチピクチャにおける基本的な行動であるポストイットでの表現や関係性の記載が、高い共通認識度にはつながらなかったことが分かる。

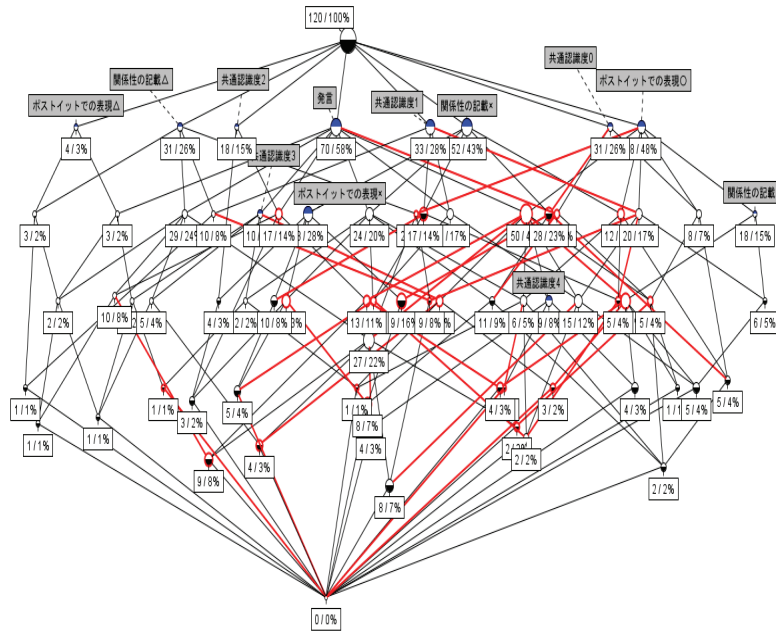


図 7-15 リッチピクチャ群の形式概念分析により得られた Hasse 図（全体）

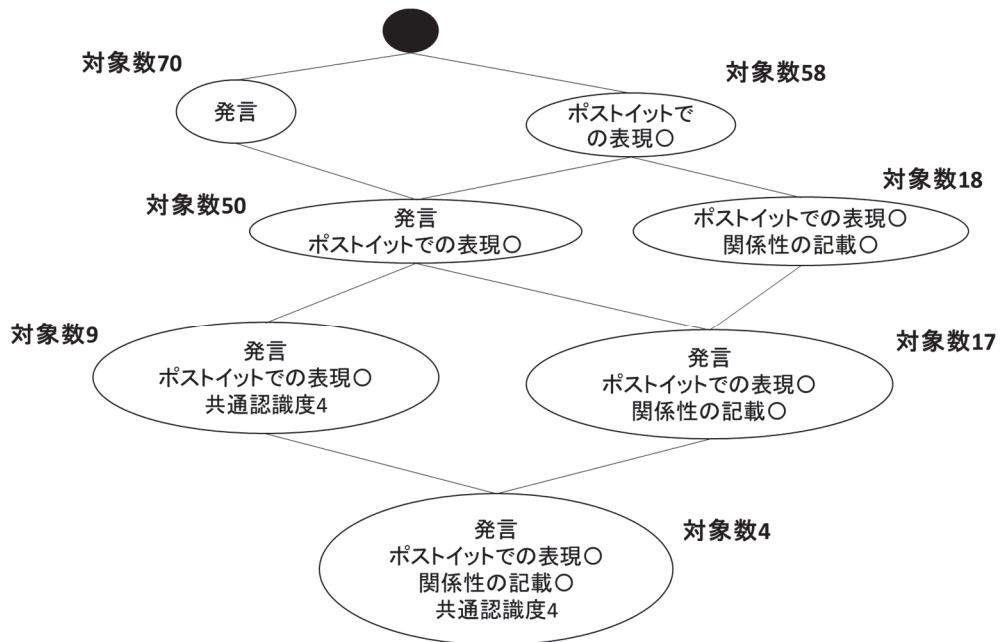


図 7-16 リッチピクチャ群の形式概念分析により得られた Hasse 図（重要な部分）

表 7-15 リッチピクチャ群で得られたアソシエーションルールと確信度（重要な部分）

No.	属性Xを有する対象数	属性X	確信度	属性Yを有する対象数	属性Y
1	58	{ポストイットでの表現○}	86%	50	{発言}
2	18	{関係性の記載○}	100%	18	{ポストイットでの表現○}
3	18	{ポストイットでの表現○} {関係性の記載○}	94	17	{発言}
4	50	{発言} {ポストイットでの表現○}	34	9	{共通認識度 4}
5	50	{発言} {ポストイットでの表現○}	18	17	{関係性の記載○}
6	9	{共通認識度 4}	100	9	{発言} {ポストイットでの表現○}
7	9	{発言} {ポストイットでの表現○} {共通認識度 4}	44	4	{関係性の記載○}
8	17	{発言} {ポストイットでの表現○} {関係性の記載○}	24	4	{共通認識度 4}

7.4.7. 形式概念分析による評価のまとめ

7.4節では、GBSP 群とリッチピクチャ群における問題状況の共有の様子を録画したビデオデータから抽出した行動と7.3.6節の結果として得られた共通認識度を属性とした形式概念分析を行うことで、GBSP における行動が共通認識度に与える影響について質的に評価した。その結果、対照手法であるリッチピクチャの基本的な行動である「ポストイットでの表現（問題状況における要素を明示かする行動）」や「関係性の記載」が高い共通認識度に与える影響が小さかった一方で、GBSP の基本的な行動である「ゲーム化」や「トークンを用いた議論」が高い共通認識度を生み出していることを導出することができた。

第8章 まとめと今後の展望

本章では、第1章から第8章の内容を総括し、本研究の取り組み内容とその成果についてまとめるとともに、残された課題や今後の展望について述べる。

8.1. 研究内容

本研究ではまず、複雑化する現代社会において登場した新しいパラダイムとしての複雑系科学について述べ、その基盤となる考え方である創発性について自然な形で取り扱うことのできるエージェントベースなアプローチによる社会シミュレーションの意義と現代社会および近未来社会における重要性について論じた。また、それを実行するための具体的な手法であるエージェントベースモデリングとマルチエージェントシミュレーションについて、基本的な概念と特徴、手順などを整理した。そして、それらの手法の有効性を高め、より幅広くインパクトの高い形で社会問題に対して応用していくにあたっての問題点・課題点を分析し、その解決につながるアプローチとして専門家とステークホルダの対話に基づく協働モデリングというアイデアを導出した。以上の議論に基づいて、本研究では下記に再掲する2つを目的と定めた。ここまでの議論は第1章に記載している。

【目的Ⅰ】

専門家とステークホルダが対話しながら協働でエージェントベースモデリングを行い、ステークホルダらが持つ暗黙知・経験知・専門知を活用しながらモデルを構築することのできる手法を確立する。これにより、幅広い分野や個別具体の事象、人と機能システムが相互作用を持つ状況などを対象とした場合において有用性・妥当性の高いモデルやシミュレーションを効果的に構築することを可能とし、エージェントベースモデリングの社会応用をより拡大することを目指す。

【目的Ⅱ】

提案する対話型モデリング手法を、様々な分野における実際のエージェントベースモデリングに適用し、マルチエージェントシミュレーションの構築を含む活動を実践することで、提案内容を洗練させるとともに、その有用性を示す。

【目的Ⅲ】

ステークホルダの知見を必要とする分野横断型の研究活動や教育活動において、提案する対話型モデリング手法を活用し、それぞれの活動における問題点を解決し、優れた成果を得る。

目的Ⅰ～目的Ⅲを達成するため、本研究では以下の(1)から(5)のようなことに取り組んだ。各項目には、取り組み内容に対応する本論文の章を付記する。

- (1) 専門家とステークホルダが対話しながら協働でエージェントベースモデリングを実践することを可能とする手法として、ゲームのデザインプロセスを応用した GBSP (Game-Based Situation Prototyping) を発案した (第2章)。
- (2) GBSP の具体的な方法論を構築し、その改善と洗練のために様々な活動において試行と実践を繰り返した (第3章～第6章)。
- (3) 組織におけるチーム形成、企業間のポジショニングに関する空間的競争、電力ネットワークにおける需給制御など、幅広い分野の研究活動に GBSP を適用してエージェントベースモデルと社会シミュレーションを構築した。また、それぞれの研究活動を進め、構築したモデルやシミュレーションが有用であることを示した (第3章～第5章)。
- (4) 複雑系科学に関する初学者向けの教育など、研究活動だけでなく教育活動の場においても GBSP を実践した (第6章)。
- (5) GBSP の有効性の一部を定量的に評価する方法を検討し、評価実験を実践した (第7章)。

8.2. 研究成果

以下では、前節でまとめた本研究における取り組みの結果として、どのような成果が得られたのかについて示す。

8.2.1. 対話型協働モデリング手法の構築

ゲームとエージェントベースモデルの類似性に注目することで、「ゲームをつくることを通して対話しながら協働でモデリングを行う」という手法を発案し、その実践方法を構築した。実践方法の構築にあたっては、ゲームデザインにおけるプロトタイピングの手法やゲームメカニクスにおける頻出パターンを整理し、それらをエージェントベースモデルが持つべき要素と対応づけることでエージェントベースモデリングを実施するのに適した手法となるよう工夫した。さらに、構築した手法を様々な分野の研究活動 (第3章～第5章) や教

育活動（第6章）に導入して実践することで、改善点を洗い出してプロセスや使用するツールセットを洗練させていった。その結果、最終的に得られた手法を GBSP（Game-Based Situation Prototyping）と名付け、独自の対話型協働モデリング手法として確立した。GBSP のコンセプトや構築した手法について、関連する学術講演会において発表したところ、優秀発表賞および優秀論文賞を受賞し、その成果が学術コミュニティにおいて認められた[46]。

8.2.2. 提案手法の有用性確認

提案手法である GBSP が実際に有効性であることを実践的に確認するため、様々な分野の研究活動（第3章～第5章）や教育活動（第6章）に適用し、協働モデリングを行った。

研究活動としては、組織におけるチーム形成、企業間のポジショニングに関する空間的競争、電力ネットワークにおける需給制御などの社会シミュレーションを軸とするものを扱った。組織におけるチーム形成のテーマでは、産学連携を担当する職員と異分野との協同研究経験を持つ研究者という形で、着目する現象におけるエージェントに相当するステークホルダがモデリングに参加した。企業間のポジショニングに関する空間的競争の研究では、マーケティング分野の研究者との協働モデリングを実践した。電力ネットワークにおける需給制御に関する研究では異分野の研究者ではなく、エージェントベースモデリングに専門性を持つ研究者同士となったが、専門性を持たない学生を含む複数人での協働モデリングを実践した。いずれの場合も、GBSP を用いることで専門性による隔たりを問題とせず、スムーズに共通認識を得ながら、効果的にモデリングを進めることができた。また、それぞれの研究活動で得られたモデルは研究としても意義の高いものであり、8.2.3節に示すように優れた研究成果を生み出すことができた。

また、初学者を対象とした複雑系科学に関する入門教育に GBSP を導入したところ、それを導入していなかった場合と比較して、初学者によるモデリングとシミュレーション構築がスムーズに進み、高い教育効果を得ることができた。特に、シミュレーション構築に入ってから、モデリングへの手戻りがなくなったことは大きな改善である。

ここまで挙げたように、実際の研究や教育の場において GBSP は十分に効果を発揮し、ステークホルダとの協働モデリングや議論、学びの質を高めることができた。これにより、提案手法が高い有効性を持つことを実践的に確かめることができた。

8.2.3. 提案手法を活用した研究の成果

提案手法である GBSP を活用して進めたいいくつかの研究活動は、それぞれ学術的にも意義の高い優れた成果を生み出すことができた。これらの研究成果も、本研究の重要な成果である。ここでは、それぞれの研究活動の成果の概要をまとめる。

学際研究チームの形成モデル

大学などの研究機関において学際的な研究チームが構築されるプロセスについてのエージェントベースモデルとシミュレーションを構築し、研究者への情報発信を担うコーディネータが取るべき戦略について考察した。GBSP を用いることで実務担当者や、学際的な研究チームに参加したことがある研究者と協働でモデリングを行い、コーディネータの情報発信の方法と研究者チームの成立のしやすさの関係性など、詳細な行動モデルを含めたシミュレーションを構築することができた。構築したシミュレーションを様々な条件下で実行し、学際的な研究チームの成立率を評価することで、コーディネータの実務に対する示唆を得ることができた。組織横断や分野横断で取り組まなければならない課題が増加している中、本研究のように学際的な研究チームの組織化について扱う研究の重要性は増していくと考えられ、本研究はそのモデルケースになりうる。

空間的競争モデル

小売業の立地競争や製造業者の製品差別化による競争を扱う空間的競争モデルの研究では、線分市場における最小差別化原理を導出したホテリングのモデルを源流として、様々なモデル化と分析が行われてきた。本研究では、ゲーム的状况における最適化問題を解析的に解くというアプローチで行われてきた従来の空間的競争モデルの研究に対して、エージェントベースなモデル化とシミュレーションを導入することで、より複雑で多様な状況・現象を扱うことに取り組んだ。GBSP を導入してマーケティングの分野の研究者と協働でモデリングを行うことで、ホテリングモデルを 2 次元のエージェントベースモデルの形式に拡張することができた。構築したシミュレーションは、特定の条件下において最大差別化原理が働くなどの既存研究の結果を再現するとともに、従来の研究では扱うことのできなかつたランダムな消費者分布における空間的競争の帰結を計算することが可能となった。これまでに、この研究に関連する成果を国際学術誌に発表している[1]。

分散型電力供給制御モデル

「リアルタイムプライシング」は電力価格をリアルタイムに変動させることで電力需要を制御する手法であり、供給量が不安定な自然エネルギー利用において有用であると考えられている。本研究では、制約付き最適化問題としての定式化と合意問題を応用した価格調整則による分散型のリアルタイムプライシングを提案している。これはネットワーク上で隣接するプレイヤー同士が情報を交換するプロセスを経て分散的な価格調整を実現するもので、これまでの研究では情報交換のタイミングに関する制限を考慮していなかった。しかし実システムにおいては通信や計算に一定の時間を要することから、情報交換や価格更新を一定の時間ごとでしか行うことができず、システムを不安定にさせる原因となる。本研究ではこの問題に対応し、需要供給量を予測しながら価格更新を行うことで不安定さを解消することを提案し、その具体的な手法とそれを評価するためのマルチエージェントシミュレータを構築した。この研究に関連する成果は、これまで国内外の学術誌に発表しており [80][109][110]、国際的に評価の高い学術誌への掲載も実現している [79]。

8.2.4. 提案手法を活用した教育の成果

初学者を対象とした複雑系科学に関する入門教育に GBSP を導入することで、高い教育効果が期待できる独自の教育プログラムを構築および実践することができた。授業の主な対象は複雑系科学やプログラミングに関する基礎知識を持たない初学者であり、限られた時間の中で初学者が主体的にマルチエージェントシミュレーションを構築できるようサポートすることが最大の課題であった。本稿で紹介したサマー小魚モデルを用いたチュートリアル、GBSP による協働モデリング、クラウド型システムモデリングツールを用いたダイアグラムモデリングなどは、その課題に対応した工夫の例である。これらの手法を導入することで、初学者が自分自身の力でシミュレーションを構築し、実践的に学ぶことのできる機会を創出することができた。この成果に関して、これまでに国際学会論文として発表している [108]。

8.2.5. 協働モデリング手法に関する評価手法の構築

第7章で示したように、協働モデリング手法としての GBSP の有効性を定量的に評価する試みとして、リッチピクチャを用いた状況共有との比較実験とその結果を用いた評価を行った。共有度と共有認識度という 2 つの指標を提案し、比較実験の結果に適用することで、GBSP が共有認識度において有効性が高いことを示した。さらに、形式概念分析の手法

を用いることで GBSP を用いた対話が知識の共有に対してどのような効果を持つのかを評価し、どのようなゲーム化の行動が共有認識度に寄与しているかを明らかにすることができた。この一連の実験と評価は、GBSP の効果の一部について有効性を評価したに過ぎないが、協働モデリングの手法に対する定量的な評価手法そのものを確立し、その実践を行ったという意義は大きい。協働モデリングの手法について定量的な直接的な評価を行うのは一般的に難しく、既存研究においては定性的な評価や、モデリングの成果を介した間接的な評価が行われる場合が多い。今回提案した共有度および共有認識度を用いた評価や、形式概念分析を用いて行動と指標の関係性を定量的に評価する手法は、GBSP の有効性評価にとどまらず、他の協働モデリングや対話手法、ワークショップ手法などの評価にも適用することができる。

8.2.6. 成果のまとめ

8.2.1 節から 8.2.5 節に示した成果は、次のようにまとめることができる。

- 専門家とステークホルダが対話しながら協働でエージェントベースモデリングを行うことのできる手法として、GBSP (Game-Based Situation Prototyping) を提案し、その具体的な手法を構築した。
- GBSP を様々な活動に適用して実践を行うことで、手法を改善するとともに、それが実問題への適用に対して高い有効性を持つことを示すことができた。また、有効性の一部については定量的にも評価することができた。
- GBSP を導入した様々な活動において優れた成果を出すことができた。研究活動への適用においては、それぞれ学術的意義の高い研究成果を生み出し、教育活動への適用においては、教育効果の高い独自の教育プログラムを構築することができた。

上記の成果は、それぞれ**目的 I**～**III**に対応しており、研究目的を達成することのできる十分な成果を生み出すことができた。

8.3. 今後の展望

8.3.1. ゲームからシミュレーションへの変換

本研究では専門家とステークホルダが対話しながら協働でモデリングを行うことのできる効果的な手法を構築した。一方で、モデルとして協働構築したゲームからマルチエージェントシミュレーションを構築するプロセスについては整理されたプロトコルやメソッドは存在せず、シミュレーション構築を担当する専門家次第である。そのため、シミュレーシ

ンの見やすさや有用性、得られたモデルとしてのゲームとの整合性などについては専門家の経験とスキルに依存してしまう。ステークホルダとの対話をエージェントベースモデルに変換するプロトコルとして機能する GBSP のように、GBSP の結果として得られたゲームをマルチエージェントシミュレーションに変換するプロトコルを構築することができれば、シミュレーション構築を担当する専門家の経験やスキルに依存することなく協働モデリングの結果からシミュレーションを得ることができる。そのようなプロトコルが実現すれば、GBSP と合わせて、ステークホルダとの対話に基づくモデリングとシミュレーションを専門性に依存せず誰でも効果的にできるようになり、エージェントベースな社会シミュレーションを活用することのできる幅を大きく広げることができると期待できる。

このようなプロトコルを実現する手段として、著者らは UML (Universal Modeling Language) またはそれを拡張したものを用いてゲームを記述し、シミュレーション構築へのインプットとするというアイデアを提案し、その具体化に取り組んでいる。UML はソフトウェア開発においてデータの構造や処理の流れを分かりやすく図示するために定められた記法であり、ソフトウェアを設計する際のモデル化の手法として広く普及している。マルチエージェントシミュレーションは一種のソフトウェアであると考えことができ、UML によってモデリングや設計が可能である。そこで、GBSP の結果として得られたゲームのルールやメカニクスを UML として記述する方法やパターンを整理しておけば、UML による記述を介することでゲームからシミュレーションへの変換が可能である。ゲームのルールやメカニクスも少数のメカニクスパターンの組み合わせで表現することができ、例えば Machination[104]などのゲームに特化したモデリング記法を用いれば、どのようなゲームもモデルとして記述することができる。同様に、ゲームを記述するための基本的な UML の記述パターンが明らかになれば、ゲームを一定の手続きで UML として記述することができ、ゲームとシミュレーションの変換を担うプロトコルを構築することが可能であると考えている。現在、このようなアイデアに基づいて、UML を用いてゲームを記述するプロセスの整理とゲームをシミュレーションに変換するプロトコルの構築に取り組んでいる。また、これまでに UML などのシステムモデルを、クラウド上でコミュニケーションしながら協働で構築することのできるツールの開発にも取り組んでおり[105][106]、上記のようなゲームとシミュレーションの変換プロセスに活用することができると考えている。

8.3.2. ステークホルダとの協働による社会デザイン

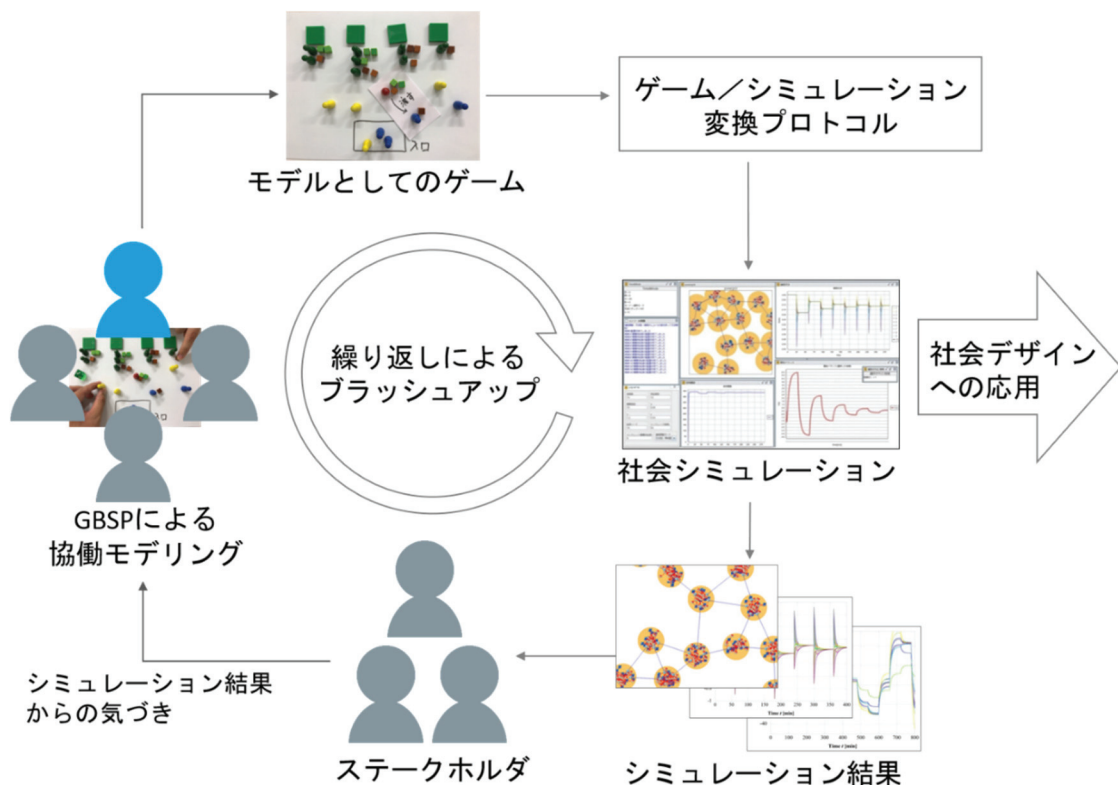


図 8-1 ステークホルダとの協働による社会デザインのプロセス

本研究の成果である GBSP を用いると、ステークホルダの経験値や暗黙知を効果的に引き出す形で協働でのモデリングが可能となり、妥当性および有用性の高い社会シミュレーションを素早く構築することが可能となる。さらに、8.3.1 節で示したようなゲームからシミュレーションへの変換プロトコルがあれば、協働モデリングの成果として得られたゲームを素早く正確にマルチエージェントシミュレーションに変換することができる。その結果として得られるマルチエージェントシミュレーションは、着目する社会現象に関する社会シミュレーションとなっており、様々な条件やシナリオのもとでのシミュレーションを行ってその結果を表示することができる。こうして得られた社会シミュレーションの結果をステークホルダに提示することで、ステークホルダは自身を持つ経験や知識と照らし合わせてシミュレーションの結果を評価し、専門家に対してフィードバックを行うことができる。ここで再び GBSP を用いることで、ステークホルダからのフィードバックを効果的に反映する形での協働モデリングを行い、モデルやシミュレーションをより現実近く有用性の高いものへとブラッシュアップしていくことができる。このようなサイクルを繰

り返して得ることのできる社会シミュレーションは、ステークホルダの経験や知見を的確に取り込むことで妥当性および有用性の高いものとなっており、より良い社会をデザインするための予測や最適化などに用いることができる。これは、ステークホルダと専門家が協働で社会デザインを行うことができる優れた方法であり、本研究が目指す姿である。図 8-1 にその概念図を示す。

第 1 章で論じたように現代社会は高度に複雑化し、様々な要因が絡み合う形で多くの問題が生じている。そのような問題を解決するには、一つの専門性だけでなく、様々な専門性や知見を組み合わせて取り組まなければならない。また、将来の社会の方向性として、Society5.0 などのテクノロジーによってシステム化された社会や、持続可能な開発目標 (SDGs) を達成できる社会などが求められている。Society5.0 のような社会をデザインするためには、人と機能システムの相互作用についてよく理解して予測しなければならない。SDGs を達成するためには、文化や地域ごとの特性に合わせて自然と技術と経済と生活が調和をとるような社会をデザインしなければならない。このようなとき、複雑な社会をボトムアップに自然な形で記述し、理解したり予測したりすることのできるエージェントベースアプローチな社会シミュレーションが有効である。さらに、単一の専門性だけでなく、異なる分野の研究者の知見や、諸問題に当事者として関わっている人たちなど、様々なステークホルダの知恵や経験が必要となってくる。本研究が目指す図 8-1 のようなステークホルダと協働で社会をモデリングし、シミュレーションによって理解と予測を行う方法は、まさにそのような社会デザインを実現するものであり、GBSP をはじめとする本研究の提案手法と、それを活用した研究成果は、その第一歩を刻むものであると言える。

参考文献

- [1] Peter Dicken, "Global Shift: Reshaping the Global Economic Map in the 21st Century," Sage, 4th edition, 2003
- [2] Julie Cidell, "Air transportation, airports, and the discourses and practices of globalization." *Urban Geography*, Vol.27, No.7, pp.651-663, 2006
- [3] David Block, "Globalization, transnational communication and the Internet." *International journal on multicultural societies* Vol.6, No.1, pp.13-28, 2004
- [4] Graeme Hutton and Maggie Fosdick. "The globalization of social media: Consumer relationships with brands evolve in the digital space." *Journal of advertising research* Vol.51, No.4, pp.564-570, 2011
- [5] Sudhir Dixit, et al. "Introduction to globalization of mobile and wireless communications: Today and in 2020." *Globalization of Mobile and Wireless Communications*. Springer, Dordrecht, pp.1-8, 2011
- [6] 山本尚利, 「近未来日本の事業環境予測」, 早稲田国際経営研究 48 pp.43-55, 2017
- [7] 井庭崇, 福原義久, 「複雑系入門: 知のフロンティアへの冒険」, NTT 出版, 1998
- [8] John H Holland, "Emergence: From chaos to order," OUP Oxford, 2000
- [9] 北村新三, 喜多一, 「創発システム」 計測と制御 Vol.40, No.1, pp.94-99, 2001
- [10] Stephen Wolfram, "A new kind of science," Vol. 5, Champaign, IL: Wolfram media, 2002
- [11] Brent Davis, and Dennis Sumara, "Complexity science and education: Reconceptualizing the teacher's role in learning." *Interchange* Vol.38, No.1, pp.53-67, 2007.
- [12] Allen Downey, "Think complexity: complexity science and computational modeling," O'Reilly Media, Inc., 2018.
- [13] Albert-László Barabási, "Network science," Cambridge university press, 2016.
- [14] Thietart, Raymond-Alain, and Bernard Forgues. "Chaos theory and organization," *Organization science*, Vol.6, No.1, pp.19-31,1995
- [15] Thomas Bäck, David B. Fogel, and Zbigniew Michalewicz, eds. "Evolutionary

- computation 1: Basic algorithms and operators,” CRC press, 2018.
- [16] Steven F Railsback, and Volker Grimm, “Agent-based and individual-based modeling: a practical introduction,” Princeton university press, 2019.
- [17] Alexis Drogoul, Jacques Ferber, and Cristophe Cambier, "Multi-agent simulation as a tool for analysing emergent processes in societies," Proceedings of Simulating Societies Symposium, 1994.
- [18] Nigel Gilbert, "Simulation: A new way of doing social science," American Behavioral Scientist Vol.42, No.10, pp.1485-1487, 1999
- [19] Ramzi Suleiman, Klaus G. Troitzsch, and G. Nigel Gilbert, eds. “Tools and techniques for social science simulation,” Springer Science & Business Media, 2012.
- [20] 出口弘, 「社会シミュレーション & サービスシステムが目指す世界」, 計測と制御 Vol.52, No.7, pp.563-567, 2013
- [21] Thomas C. Schelling, "Models of segregation," The American Economic Review Vol.59, No.2, pp.488-493, 1969
- [22] Michael D. Cohen, James G. March and Johan P. Olsen, "A garbage can model of organizational choice," Administrative science quarterly, Vol.17, No.1 pp.1-25, 1972
- [23] Douglas A. Samuelson and Charles M. Macal. "Agent-based simulation comes of age," OR MS TODAY Vol.33, No.4, pp.34, 2006
- [24] Nigel Gilbert and Pietro Terna, "How to build and use agent-based models in social science," Mind & Society Vol.1, No.1, pp.57-72, 2000
- [25] Gerd Wagner and Florin Tulba. "Agent-oriented modeling and agent-based simulation," International Conference on Conceptual Modeling, Springer, Berlin, Heidelberg, 2003.
- [26] Paul Davidsson, "Agent based social simulation: A computer science view," Journal of artificial societies and social simulation Vol.5, No.1, 2002
- [27] Patrick M. Boesch and Francesco Ciari. "Agent-based simulation of autonomous cars," 2015 American Control Conference (ACC). IEEE, 2015.
- [28] Giancarlo Fortino, et al. "Modeling and simulating Internet-of-Things systems: A hybrid agent-oriented approach," Computing in Science & Engineering Vol.19, No.5 pp.68-76, 2017

- [29] 科学技術基本計画, <https://www8.cao.go.jp/cstp/kihonkeikaku/index5.html>, 内閣府
(2019年12月10日閲覧)
- [30] 貝原俊也, 下原勝憲, 「System of systems コンセプトと超スマート社会」計測と制御
Vol.55, No.4, pp.288-290, 2016
- [31] 原辰次, 本多敏, 「超スマート社会におけるシステム科学技術概論」計測と制御 Vol.55,
No.4, pp.284-287, 2016
- [32] 瀧寛和, 「Society5.0 技術イノベーションに向けて」, 電気学会誌 Vol.137, No.5,
pp.275-275, 2017
- [33] R Axelrod and L. Tesfatsion, “Appendix AA guide for newcomers to agent-based
modeling in social sciences,” Handbook of Computational Economics, Vol.2, No.
Grant 0240853, pp.1647-1659.
- [34] Peer-Olaf Siebers and Uwe Aickelin. "Introduction to multi-agent simulation"
Encyclopedia of decision making and decision support technologies. IGI Global,
pp.554-564, 2008
- [35] 塩沢由典, 「ミクロマクロループについて」, 経済論叢, No.164, Vol.5, pp.1-77, 1999
- [36] Tobias Lorenz, "Abductive fallacies with agent-based modeling and system
dynamics," International Workshop on Epistemological Aspects of Computer
Simulation in the Social Sciences, Springer, Berlin, Heidelberg, 2006.
- [37] Jay W. Forrester, "System dynamics, systems thinking, and soft OR." System
dynamics review Vol.10, No.2-3, pp.245-256, 1994
- [38] John D. Sterman, "System dynamics modeling: tools for learning in a complex
world," California management review Vol.43, No.4, pp.8-25, 2001
- [39] Y. Bar-Yam, “General features of complex systems,” Encyclopedia of Life Support
Systems (EOLSS), UNESCO, EOLSS Publishers, September 2002
- [40] 高橋真吾, 「社会システムの研究動向 3—評価・分析手法 (1)—モデルの解像度と妥当
性評価」計測と制御 Vol.52, No.7, pp.582-587, 2013
- [41] J. W. Forrester, “Principles of Systems,” Productivity Pr, 1988.
- [42] 吉村忍, 西川紘史, 守安智, 「知的マルチエージェント交通流シミュレータ MATES の
開発」, シミュレーション, No.23, Vol.3, pp.228-237, 2004
- [43] 森俊勝, 「日本におけるマルチエージェントシミュレーション活用の動向」, 情報処理,

Vol.55, No.6, pp.585-590, 2014

- [44] Paul Davidsson, et al. "Applications of agent based simulation." International Workshop on Multi-Agent Systems and Agent-Based Simulation. Springer, Berlin, Heidelberg, 2006.
- [45] 三浦政司, 前波晴彦, 高橋真吾, 「Game-Based Situation Prototyping による協働型モデリングの実践」, 計測自動制御学会社会システム部会第 18 回研究会, p.23, 2019 年 3 月
- [46] 三浦政司, 前波晴彦, 「Game-Based Situation Prototyping による協働型モデリングの提案」, 計測自動制御学会社会システム部会第 12 回研究会, pp.40-43, 2017 年 3 月
- [47] 三浦政司, 前波晴彦, 「ゲーム設計とシステム設計の手法を応用した, エージェントベースモデリングのための対話プロセスの提案」, 第 10 回計測自動制御学会社会システム部会研究会論文集, pp.211-214, 2016 年 3 月
- [48] E. Adams and J. Dormans, "Game Mechanics: Advanced Game Design," New Riders Press, pp.63-68, 2012
- [49] Liraviasl, K. Khedri, et al. "A framework for modelling reconfigurable manufacturing systems using hybridized discrete-event and agent-based simulation," IFAC-PapersOnLine, Vol.48, No.3, pp.1490-1495, 2015
- [50] Pavel Vrba and Vladimír Marik. "Simulation in agent-based manufacturing control systems," 2005 IEEE International Conference on Systems, Man and Cybernetics, Vol. 2, 2005.
- [51] 鳥居大祐, 石田亨, 「参加型モデリングとマルチエージェントシミュレーション」, 電子情報通信学会技術研究報告, 人工知能と知識処理, Vol.133, No.104, pp.7-12, 2004
- [52] J. Gibson, "Introduction to Game Design, Prototyping, and Development," Addison-Wesley Professional, pp.125-140, 2014
- [53] 新井潔, 「ゲーミングシミュレーション」, オペレーションズ・リサーチ: 経営の科学, Vol.49, No.3, pp.143-147, 2004
- [54] 小幡範雄, 「環境コンフリクト実験ゲーム」, 技報堂出版, 1992
- [55] 菱山玲子, 「マルチエージェントシミュレーションにおけるゲーミングの利用」, 情報処理, Vol.55, No.6, pp.557-562, 2014
- [56] Jack AM Vennix, "Group model building," Chichester, 1996

- [57] S. Peck, "Group model building: facilitating team learning using system dynamics." *Journal of the Operational Research Society*, Vol.49, No.7, pp.766-767, 1998
- [58] Michel Etienne, "Companion modelling A participatory approach to support sustainable development" Versailles, Quae, 2011
- [59] François Bousquet, et al. "Multi-agent systems and role games: collective learning processes for ecosystem management," *Complexity and ecosystem management: the theory and practice of multi-agent systems*, Edward Elgar, Cheltenham, UK, pp.248-285, 2002
- [60] H. Maenami and M. Kudo, "Examining National Funding Programs for Science and Technology in Regional Areas in Japan", *Proceedings of AAAS2015*, 2014
- [61] 前波晴彦, 「『地域ニーズ即応型』を事例とした中小企業向け支援制度利活用の影響要因」, *産学連携学*, Vol.10, No.2, pp.14-23, 2014
- [62] H. Kawashima and H. Tomizawa, "Accuracy evaluation of Scopus Author ID based on the largest funding database in Japan," *Scientometrics*, Vol.103, No.3, pp.1061-1071, 2015
- [63] H. J. Falk-Krzesinski, "Guidance for team science leaders: Tools you can use," *The Academic Executive Brief*, Vol.2, No.1, pp.6-9, 2012
- [64] M. J. Nohd Nor, "The Malaysian experience: A new approach in managing multi-disciplinary research projects," *The Academic Executive Brief*, Vol.2, No.1, pp.10-12, 2012
- [65] S. C. Hsu, K. W. Weng, Q. Cui and W. Rand, "Understanding the complexity of project team member selection through agent-based modeling," *International Journal of Project Management*, Vol.34, No.1, 2016
- [66] 清水悟, 奥田隆史, 井手口哲夫, 田学軍, 「マルチエージェントアプローチによる組織的知識創造の効率性評価に関する研究-エージェントの表現に消費者価値観分析モデルを利用した」, *情報処理学会全国大会講演論文集 2012.1*, pp.191-193, 2012
- [67] 岡本和也, 高橋真吾, 大堀耕太郎, 山根昇平, 「サービス業務としての大学事務のエージェントベースモデル化とシナリオ分析」 *計測自動制御学会第3回社会システム部会研究会資料*, pp.17-22, 2013
- [68] Harold Hotelling, "Stability in Competition," *The Collected Economics Articles of*

- Harold Hotelling, Springer, pp.50–63, 1990
- [69] C. d'Aspremont, J. J. Gabszewicz and J-F. Thisse (1979), “On Hotelling's, “Stability in Competition”,” *Econometrica*, Vol.47, No.5, pp.1145-1150, 1979
- [70] J. M. Epstein and R. L. Axtell, “Growing Artificial Societies: Social Science from the Bottom Up,” Bradford Book, 1996
- [71] 白石秀壽, 三浦政司, 「空間的競争モデルの展望: マルチエージェントシミュレーションの可能性」, 地域学論集, Vol.14, No.1, pp.109-121, 2017
- [72] T. Tabuchi, “Two-stage Two-dimensional Spatial Competition between Two Firms,” *Regional Science and Urban Economics*, Vol.24, No.2, 207–227, 1994
- [73] R. J. Aumann, “Backward Induction and Common Knowledge of Rationality,” *Games and Economic Behavior*, Vol.8, No.1, pp.6-19, 1995
- [74] J. Momoh, “Smart Grid-Fundamentals of Design and Analysis,” Wiley-IEEE Press, 2012
- [75] E. Bitar, P. P. Khargonekar and K. Poolla, “Systems and control opportunities in the integration of renewable energy into the smart grid,” *IFAC Proceedings Volumes*, Vol.44, No.1, pp.4927-4932, 2011
- [76] W. Berger and F. C. Scheweppe, “Real time pricing to assist in load frequency control,” *IEEE Trans. on Power Systems*, Vol.4, No.3, pp.920-926, 1989
- [77] M. Roozbehani, M. Dahleh and S. Mitter “On the stability of wholesale electricity markets under real-time pricing,” *Proceedings of 49th IEEE Conference on Decision and Control*, pp.1911-1918, 2010
- [78] D. P. Bertekas, J. N. Tsitsiklis “Parallel and Distributed Computation,” Athena Scientific, 1997
- [79] Kazunori Sakurama and Masashi Miura, “Communication-based Decentralized Demand Response for Smart Microgrids” *IEEE Transaction on Industrial Electronics*, Vol.64, No.6, 2017
- [80] Kazunori Sakurama and Masashi Miura, “Distributed constraint optimization on networked multi-agent systems,” *Applied Mathematics and Computation*, No.292, pp.272-281, 2017
- [81] 渡部 慶二, 「むだ時間システムの制御」, コロナ社, 1993

- [82] R. Yin, P. Xu, M. A. Piette and S. Kiliccote, “Study on auto-DR and pre-cooling of commercial buildings with thermal mass in California,” *Energy Build*, Vol.42, No.7, pp.967-975, 2010
- [83] D. G. Luenberger and Y. Ye “Linear and Nonlinear Programming” Kluwer Academic Publishers, 2003
- [84] D. Friedman “Price Theory: An Intermediate Text,” South-Western Publishing Co., 1990
- [85] Masashi Miura, Yuta Tokunaga and Kazunori Sakurama “Graphical and Scalable Multi-Agent Simulator for Real-time Pricing in Electric Power Grid,” *Artificial Life and Robotics*, Vol.21, No.2, pp.181-187, 2016
- [86] 中川徹, 小柳義夫, 「最小二乗法による実験データ解析」, 東京大学出版会, pp.30-32, 1982
- [87] 山影進, 「人工社会構築指南」, 書籍工房早山, 2008
- [88] M. M. Waldrop, “COMPLEXITY: THE EMERGING SCIENCE AT THE EDGE OF ORDER AND CHAOS,” Simon & Schuster, 1992.
- [89] 山崎幸輔, 木村大希, 安田早希, 高田修汰, 「生死を賭けた籠城戦シミュレーション～鳥取飢え殺し～」, 第15回MASコンペ, 2015.
- [90] Masashi Miura, “Project Based Learning with Multi-Agent Simulation in Liberal Arts Education,” *Proceedings of the 3rd International Conference on Applied Computing & Information Technology*, pp.242-247, 2015.
- [91] 構造計画研究所, MAS コミュニティ, <http://mas.kke.co.jp/> (2017年5月1日閲覧)
- [92] 兼田敏之, 「*artisoc* で始める歩行者エージェントシミュレーションー原理・方法論から安全・賑わい空間のデザイン・マネジメントまでー」, 書籍工房早山, 2010.
- [93] 三浦政司, 櫻間一徳, 「マルチエージェントシミュレーションをはじめよう」, システム/制御/情報, vol.61, No.5, pp.169-174, 2017.
- [94] サメ小魚モデル配布ページ, <http://m-miura.jp/mas/shark-fish/> (2017年5月1日閲覧)
- [95] 鳥海不二夫, 山本仁志, 「マルチエージェントシミュレーションの基本設計」, 情報処理, Vol.55, No.6, pp.530-538, 2014.
- [96] BALUS 8 版, <http://app.balus.me> (2017年5月1日閲覧)
- [97] 三浦政司, 前波晴彦, 「Game-Based Situation Prototyping による協働型モデリングの

- 提案」, pp.40-43, 計測自動制御学会第 12 回社会システム部会研究会資料, 2017
- [98] Peter Checkland, "Soft systems methodology," Human systems management Vol.8, No.4, pp.273-289, 1989
- [99] Peter Checkland, "Soft systems methodology: a thirty year retrospective," Systems research and behavioral science, Vol.17, No.1, pp.11-58, 2000
- [100] 高橋真吾, 「システム学の基礎」, 培風館, 2007
- [101] P. J. Lewis, "Rich picture building in the soft systems methodology." European Journal of Information Systems Vol.1, No.5, pp.351-360, 1992.
- [102] 長田博泰, 「形式概念にもとづく質的研究」, 社会情報, 2004
- [103] 鈴木治, 室伏俊明, 「形式概念分析-入門・支援ソフト・応用」, 知能と情報, Vol.19, No.2, pp.103-142, 2007
- [104] Joris. DORMANS, "Machinations: Elemental feedback structures for game design," Proceedings of the GAMEON-NA Conference, pp. 33-40, 2009
- [105] 三浦政司, 南部陽介, 「協働型システムモデリングツールを用いたシステム設計に関する実践教育」, 平成 28 年度工学教育研究講演会講演論文集, pp.162-163, 2016 年 9 月
- [106] Yohsuke Nambu, Masashi Miura, Ryosuke Yoshizawa, Toshishige Hagihara, Shunsuke Kimura, Akira Yumiyama and Satoru Igarashi , "Development of Open Model-based Collaboration Tool and Application on Nano-satellite Project," Transactions of the Japan Society for Aeronautical and Space Sciences, Aerospace Technology, 2018
- [107] Masashi Miura and Hidetoshi Shiroishi, "Agent-Based Modeling and Simulation for Two-Dimensional Spatial Competition," Agent and Multi-agent Systems: Technology and Applications, Smart Innovation Systems and Technologies Book Series, pp.65-75, Springer, 2018
- [108] Masashi Miura, "STEM Project-Based Learning in Designing and Making Products for Developing Countries," the 12th International Dryland Development Conference, Poster Session, Aug 2016
- [109] 櫻間一徳, 三浦政司, 「スマートメーターの通信ネットワークを活用したデマンドレスポンス」, クリーンエネルギー, 日本工業出版株式会社, Vol.26, No.5, pp.31-34, 2017
- [110] 三浦政司, 櫻間一徳, 「需要供給予測を伴う分散型リアルタイムプライシング」, シ

システム制御情報学会論文誌, 第 31 号, 第 3 卷, pp.121-128, 2018

謝辞

本研究を遂行し、まとめるにあたって、実に多くの方々にお世話になりました。この場を借りて感謝の意を述べさせて頂きたいと思います。

まず、指導教官主査の西田信一郎先生は、私の研究活動や論文執筆がなかなか進まない中、根気強く助言や指導をして頂きました。思うように研究活動が進まず、どうしたら良いかわからなくて私が途方にくれている時でも、常に冷静な視点で、その時点を出発点と考えた場合の最適解に向けて有用なアドバイスを授けて下さるとともに、頑張りましょうと心強い声をかけて頂きました。多くのご面倒をかけてしまって申し訳ないという気持ちと同時に、最後までお付き合い頂いたことに対する大きな感謝の気持ちでいっぱいです。

また、急なお願いにも関わらず副査として多大なご指導とご協力を頂いた石井晃先生と小出隆夫先生にも同様の感謝を申し上げたいです。石井先生には学内や学会などでお会いするたびにお声をかけて頂き、鋭いアドバイスと共に励ましの言葉を頂きました。また、本論文の審査にあたっては、専門的なアドバイスだけでなく、文章の論調や整合性に関する広い視点での助言を多く頂きました。本当にありがとうございます。小出先生には副査としてご指導頂くだけでなく、私が勤務している鳥取大学工学部ものづくり教育実践センターのセンター長として、日々の仕事からキャリアの相談まで幅広くご支援頂きました。いつも優しく相談に乗って頂き、仕事の面でも研究の面でも大きな励ましをもらいました。本当にありがとうございました。

本研究の提案手法である GBSP に関して、構想の段階から様々なアドバイスを頂き、研究活動にご一緒して頂いた自然科学研究機構の前波晴彦先生および早稲田大学の高橋真吾先生にも大変お世話になりました。前波先生は鳥取大学に在籍していた頃から本研究に関する活動の多くにお付き合い頂き、ご専門である科学社会学の観点から多くの有用な助言を頂きました。また、第3章で記述した学際研究チームの形成モデルに関する研究では研究支援組織の実務を担う立場から GBSP に参加して頂き、たくさんの知見をご提供頂きました。高橋先生は、ゲーミフィケーション、システム論、エージェントベース社会シミュレーションなど本研究に深く関連する分野をご専門としており、その豊富な経験と知識から多くの指導・助言を頂きました。特に、第7章で記述した評価実験の実施や評価手法の構築にあたっては、実験フィールドの提供を含めて本当に大きなご協力を頂きました。早稲田大学

の鴨川セミナーハウスで高橋研究室の学生さんたちに協力を頂いた評価実験の取り組みは、私にとって忘れられない思い出となりました。評価実験の準備や分析において中心的な役割を担って手伝って頂いた高橋研究室大学院生の栗山誠太郎くんにも合わせて感謝の意を述べたいと思います。本当にありがとうございました。

GBSP を活用した各研究活動においては、鳥取大学の桑野将司先生と白石秀壽先生、そして京都大学の桜間一徳先生にたくさんのご協力を頂きました。桑野先生には学際研究チームの形成モデルに関する研究において、学際的な共同研究の経験を持つ研究者として GBSP にご参加頂き、研究者の振る舞いモデルについて多くの有用なご意見を頂きました。また、白石先生には空間的競争モデルに関する研究と一緒に取り組んで頂きました。白石先生とのディスカッションはモデリングや研究を進ませてくれるだけでなく、企業行動やマーケティングに関する学術的な考察など、私にとって初めてとなる事柄に触れる機会でもあり、新鮮な刺激と共に多くのことを学ばせて頂きました。桑野先生と白石先生は、研究のパートナーであるだけでなく、プライベートで食事やボードゲームを一緒に楽しむ仲でもあり、日々の生活に活力を与えてくれる友人でもありました。あらためてこの場を借りて感謝を申し上げるとともに、引き続き変わらぬ友情をお願いしたいです。そして桜間先生には分散型電力供給制御モデルに関する研究を通して、様々なご指導を頂きました。研究内容に関する議論や助言を頂くだけでなく、研究者としてあるべき姿や行動についても多くのアドバイスを頂きました。また、研究成果を学術論文にまとめる際には、粘り強く、隅々まで行き届いたご指導を頂きました。桜間先生から教えて頂いたことは、今後の私の研究活動にとっての規範と指針となると考えています。深く感謝致します。

本研究の一部は JST CREST (グラント番号 JPMJCR15K1) および中山隼雄科学技術文化財団からの研究助成を受けています。ここに記して、謝意を表したいと思います。

最後に、ここに本研究を進めるにあたってはここに挙げきれなかった多くの人々から、直接的または間接的にご指導、ご支援、ご協力を頂きました。あらためて、関わって下さった全ての皆様に感謝の言葉を申し上げます。本当にありがとうございました。

三浦 政司

付録

付録目次

付録 A：有効性評価実験に関する資料

共通情報	137
住民 A の役割情報	137
住民 B の役割情報	138
地方自治体 C の役割情報	139
専門家 D の役割情報	140
プレテスト	141
ポストテスト	142

付録 B：ソースコード

学際研究チームの形成モデルのソースコード	144
空間競争モデルのソースコード	158
分散型電力供給制御モデルのソースコード	170

付録 A : 有効性評価実験に関する資料

第 7 章で論じた有効性評価のための実験において用いた情報配布資料とペーパーテストを付録資料として掲載する。

共通情報

【共通情報】 グループ全員が知っている情報

地域 X は「住宅地」「市街地」「農地」「工場地」「未開発地」からなり、それぞれの土地は多数の地区に分かれています。「住宅地」は主に人々が暮らすスペース、「市街地」は人々が集まり、商店街や公共施設やその他の施設などが多数存在しているスペースです。「農地」は田畑のあるスペースで土地面積はあまり変動がないです。「未開発地」は山で覆われたスペースで自然であふれています。娯楽施設や飲食店や販売店などの「もの、店」の数も人口の増減に関わっています。

地域 X と都会間の人口の流出、流入を考えます。都会に流出するということは都会に移住するという事を表し、都会から流入するということは都会から地域 X に移住してくる事を表しています。

住民 A の役割情報

【役割情報】 ほとんどがグループ内であなたしか知らない情報

私は地域 X で生まれ、現在地域 X に住み続けて 18 年になるものです。現在大学の進学を考えているのですが、地域 X には大学がなく、大学へ進学するひとは近くの都会 Y に必ず進出しなければ行けません。また地域 X にくらべて都会 Y は私達若者にとって魅力的だと思います。というのも、都会 Y には「もの、店」が溢れており、また様々な雇用がありますが、現状地域 X には「もの、店」があまりなく、また雇用も充実していません。そのため、私達の歳で大学に行かない人も、そのような魅力に惹かれて都心へ進出しています。逆に言えば、地域 X にも「もの、店」が増えたり雇用の質が向上したりすれば、若者の流出は減ると思います。また、一度都会へ流出しても、地域 X へ帰ってくる人もいます。このような人達は、大学卒業の就職のタイミングが中心となっているため、地域 X の雇用の質が向上すれば増えると思う。

現在私は「住宅地」に住んでいます。「住宅地」は人が住むところですが、最近では「もの、店」が増加しつつあります。「もの、店」が豊富になった地区は「市街地」となっています。

地域 X は自然が豊かだと思います。住宅からでも見える「未開発地」の自然の風景は魅力的です。近年では「未開発地」も徐々にではありますが開発されてしまっています。このような「未開発地」の開発によって地域 X に自然の魅力が減ってしまうと思います。自然の多くは「未開発地」にあり、この自然の多さが地域 X の自然の魅力であると思います。

地域 X の中心部は市街地の集まっている部分であり、この「中心部の景観」は重要だと思う。「中心部の景観」を良くするには様々な要因は考えられるが、私は、「もの、店」が主な要因であると思う。「もの、店」が少ないと、シャッター街などが多くなり、地域の中心部が閑散となってしまう、そのような地域は魅力的でないと思います。

人口の増減を考える際、重要になってくるのが「出生率」ですが、この「出生率」を上げるのに重要なのは、「子育ての環境」だと思う。「子育ての環境」は、「インフラの質」が向上すると良くなると思います。また自然の魅力も関わってくると思います。

住民 B の役割情報

【役割情報】ほとんどがグループ内であなたしか知らない情報

私は商店街で自営業を行っているものです。地域 X に存在する業種は、「製造」「IT」「農業」「飲食、サービス」が中心となっていますが、職の選択肢は豊富とは言えません。業種ごとの雇用数が増せばその分だけ職の選択肢が増えるため、望ましいと思います。「もの、店」の量が増えたり、インフラの質が良くなったりすれば、若者の都会への流出は減り、また一度都会へ流出した人も地域 X に戻ってくると思います。

また、都会出身の人でもライフステージの変化のタイミングで地域 X に移住する人もいます。このような人々は、地域 X の自然の魅力や子育て環境の良さや中心地の景観の良さにひかれてやってきます。現在、地方自治体による子育てに関する補助金もでており、子育て環境は良くなっていると考えられます。

「もの、店」のニーズが出てきた場合、私たち商店街コミュニティは、いち早くそのようなニーズをくみ取り、新しく出店することを心がけています。また同様に外部の企業も「もの、店」のニーズに反応して出店します。「もの、店」の増加はあらたな「飲食、サービス」業の雇用を生み出すことができます。

地域 X の住民のほとんどは「住宅地」「市街地」に住んでいます。そのため、インフラは主に「住宅地」「市街地」に必要とされています。人口の多い地区ではインフラの

使用量が多いため、質が低下してしまいます。地域 X の中心部の景観はあまり優れていません。公園などの公共施設が少なく感じます。公共施設は「市街地」や「住宅地」に存在します

地域 X は都会より自然が豊かで、「未開発地」にはたくさんの自然が広がっており、魅力的です。

地方自治体 C の役割情報

【役割情報】 ほとんどがグループ内であなたしか知らない情報

地方自治体が外部の企業に対して補助金を出すことでその企業の営業所や事業所などを地域 X に進出させることを企業誘致と呼びます。企業誘致が可能な外部企業は「製造」「IT」の企業です。「製造」の企業誘致を行うと地域 X にあらたに工場が建ち、「製造」の業種の雇用数が増します。同様に「IT」企業の企業誘致を行うと市街地に企業の営業所が建ち「IT」の業種の雇用数が増します。

「工場地」の拡大が必要の際には「未開発地」を開拓することによって新たに「工場地」とします。同様に「住宅地」の拡大が必要の際には「未開発地」を開拓し新たに「住宅地」とし、これらの資金は自治体が負担します。「住宅地」に特殊なインフラの整備や区画整備を施すことで「市街地」とすることができます。

一度地域 X から都会などに流出した後に、また地域 X に流入してくる人を「U ターン者」と呼びます。また、地域 X 出身でないが地域 X に移住してくる人を「I ターン者」と呼びます。「U ターン者」「I ターン者」は 20~40 歳の人が行います。「U ターン者」「I ターン者」の数を増やすことが地域 X の人口を増加させることにつながります。

住民のほとんどが「住宅地」「市街地」に住んでいます。人口が増えると税収があがります。今まであげた補助金の他にも、子育ての補助金を住民に出すことで、子育て環境が良くなり、出生率があがると考えられます。また地域 X での消費されるお金は税収に関わってくるため、人口が税収に与える影響は「店、もの」が増加すると強くなるという調査結果がでています。

インフラに関する住民からの意見が多くでてきており、使用量の多いインフラに対しては資金を投入する必要がでてきます。

公園などの公共施設は私たちが管理しており、公共施設への投資を増やすことで公園などを増やし、地域 X の公共施設を増やすことができる。

専門家 D の役割情報

【役割情報】 ほとんどがグループ内であなたしか知らない情報

工場は「工場地」に建てられるが、「工場地」の地区ごとに許容できる工場数が決まっているため、「工場地」が過密になってくると、「工場地」を拡大する必要がでてきません。

「住宅地」において地区ごとで「もの、店」の数が増加すると、「住宅地」を「市街地」にする必要がでるため、新たに整備しなければならない。この際、住宅地が減るため同時に地方自治体が資金を投入し、「未開発地」を「住宅地」に開拓する必要がある。「住宅地」において地区ごとで許容できる人数が決まっており、ある程度人口が増加し、過密になると「未開発地」を「住宅地」に開拓する必要がでてきます。

一度「市街地」になった地区では住民はほとんど増えないため、「もの、店」の数は住民のニーズを満たした時点で一定になる傾向があります。一度「市街地」になった地区は人口も「もの、店」の数もほとんど変動しません。「市街地」の地区では、地区に対しての許容できる事業所や公共施設数は決まっています。そのため、許容数を超えた事業所や公共施設の設立はできません。また地域 X の住民のほとんどが「市街地」「住宅地」に住んでいます。

40 歳未満の人を増やすことが地域 X での出生数を上げることに繋がります。死亡数はほとんど一定です。

40 歳未満の人は流行に敏感に反応し「もの、店」が豊富な地域を好む傾向にあり、そのため 40 歳未満の人が地域 X に増加すると「住宅地」や「市街地」での「もの、店」のニーズが増加します。

地方自治体は「未開発地」の整備にも資金を投入します。「未開発地」といっても何もしなければ荒れてしまうからです。荒れてしまうと地域 X の自然の魅力が落ちてしまうと考えられます。

地区のインフラの質は、地方自治体の投入資金が増すと上がります。

地域 X には大学がなく、若者の大学への進学率は一定です。

プレテスト

- ① 分かる範囲でよいので、覚えていることを**詳細**に記載してください。
- ② 回答は文章で表現してください。図等は記入しないでください。
- ③ 自分の与えられた情報では答えられない問題もあります。

<例題>

問の例：Uターン者が重視している事にはどのようなことが考えられますか。

回答の例：インフラの質、雇用の豊富さ、「もの、店」の豊富さを重視している。

問1：

中心地の景観はどのような要因で良くなると考えられますか。

問2：

子育ての環境はどのような要因で良くなると考えられますか。

問3：

工場地を拡大することについて説明してください。

問4：

雇用が増える要因と、その要因によってどのような雇用が増えるか教えてください。

問5：

Iターン者が重要視していることがらにはどのような事ですか。

問6：

出生数が増えるにはどのような要因が考えられますか。

問7：

地域Xの自然の魅力に関して関係している事柄について説明してください。

ポストテスト

- ① 分かる範囲でよいので、覚えていることを**詳細**に記載してください。
- ② 回答は文章で表現してください。図等は記入しないでください。
- ③ 自分の与えられた情報では答えられない問題もあります。

<例題>

問の例：一度都会に進出した後、再び地域 X に戻ってくる人が重視している事にはどのようなことが考えられますか。

回答の例：インフラの質、雇用の豊富さ、「もの、店」の豊富さを重視している。

問 1：

中心地の景観はどのような要因で良くなると考えられますか。

問 2：

子育ての環境はどのような要因で良くなると考えられますか。

問 3：

工場地を拡大することについて説明してください。

問 4：

雇用が増える要因と、その要因によってどのような雇用が増えるか教えてください。

問 5：

I ターン者が重要視していることがらにはどのような事ですか。

問 6：

出生数が増えるにはどのような要因が考えられますか。

問 7 :

地域 X の自然の魅力に関して関係している事柄について説明してください。

問 8 :

若者が地域 X から都会へ流出してしまうのはどのような要因が考えられますか。

問 9 :

市街地の拡大について、関係している事柄について説明してください。

問 1 0 :

税収が増えるにはどのような事柄が考えられますか。

問 1 1 :

インフラの質はどのような事柄に影響を受けると考えられますか。

問 1 2 :

I・U ターン者とはそれぞれどのような人ですか。また何歳ほどの人ですか。

問 1 3 :

公共施設が増える要因はどのような事柄ですか。

問 1 4 :

地域 X に「もの、店」が増えるのはどのような人が原因ですか。それを踏まえたうえで「もの、店」が増える仕組みを説明してください。

付録 B : ソースコード

第 3 章から第 5 章において述べた各研究活動の成果として得られたマルチエージェントシミュレーションのソースコードを掲載する。

学際研究チーム形成モデルのソースコード

```
//=====
// artisoc
//=====
Version = 3.5;
//-----
// Agent Rule
//-----
#begin_rule UNIVERSE
Function distance_agt(agt1 as Agt, agt2 as Agt) as Double{
//2つのエージェントの間の距離を取得する関数

//空間を取得する (agt1 の空間を取得して距離計算に使う)
Dim sp as Space
Dim agt_type as AgtType
agt_type=SpecifyAgtType(agt1)
sp=GetRideSpace(agt_type)

//距離を計算する
dim distance as Double
distance=MeasureDistance(agt1.X, agt1.Y, agt2.X, agt2.Y, sp)

return(distance)
}
//-----
Function direction_agt(agt1 as Agt, agt2 as Agt) as Double{
//agt1 から見た agt2 の角度を取得する関数[degree]

//空間を取得する (agt1 の空間を取得して角度計算に使う)
Dim sp as Space
Dim agt_type as AgtType
agt_type=SpecifyAgtType(agt1)
sp=GetRideSpace(agt_type)

//角度を計算する
dim direction as Double
direction=GetDirection(agt1.X, agt1.Y, agt2.X, agt2.Y, sp)

return(direction)
}
//-----
Function AgtTypeComp(agt1 as Agt, agt2 as Agt) as Integer{
//agt1 と agt2 のタイプが一致するとき 1, 一致しないとき 0 を出力する

Dim agt_type1 as String
Dim agt_type2 as String
Dim answer as Integer

//それぞれのエージェントタイプを文字列として取得する
agt_type1=CStr(SpecifyAgtType(agt1))
agt_type2=CStr(SpecifyAgtType(agt2))

//文字列が一致していれば 1, 一致していなければ 0 を出力する
If StrComp(agt_type1,agt_type2)=0 then
    answer=1
Else
    answer=0
End if

return(answer)
}
```



```

}
//-----
Function AgtTypeComp2(agt1 as Agt, agt_type_str as String) as Integer {
//agt1 のタイプ名が agt_type_str と一致するとき 1, 一致しないとき 0 を出力する
//agt_type_str は"Universe.空間名.種別名"という形で記述する

Dim agt_type as String
Dim answer as Integer

//それぞれのエージェントタイプを文字列として取得する
agt_type=CStr(SpecifyAgtType(agt1))

//文字列が一致していれば 1, 一致していなければ 0 を出力する
If StrComp(agt_type,agt_type_str)=0 then
    answer=1
Else
    answer=0

End if

return(answer)
}

//-----
Function eRnd(e as Double) as Double {
//e~+e の範囲の値をランダム(一様分布)で返す

Dim output as Double

output = 2*e*(Rnd()-0.5)

return(output)
}

//-----
Function UniRnd(min as Double, max as Double) as Double {
//min~max の範囲の値をランダム(一様分布)で返す

Dim output as Double

output = min + Rnd() * (max - min)

return(output)
}

//-----
Function IntUniRnd(min as Integer, max as Integer) as Integer {
//min~max の範囲の整数値をランダム(一様分布)で返す

Dim min_temp as Double
Dim max_temp as Double
Dim rnd_temp as Double
Dim output as Integer

min_temp = Cdbl(min) - 0.5
max_temp = Cdbl(max) + 0.5
rnd_temp = min_temp + Rnd() * (max_temp - min_temp)

output = Round(rnd_temp)

return(output)

Univ_Init {

ClearConsoleScreen() //コンソールをクリア
SetRandomSeed(Universe.RandomSeed) //乱数シードをセット

//変数定義

```

```

Dim i as Integer
Dim j as Integer
Dim k as Integer

Dim a as Agt
Dim b as Agt
Dim c as Agt

Dim agtset_temp as AgtSet

//空間のサイズを取得
Universe.Institute_width=GetWidthSpace(Universe.Institute) //空間の幅(x)
Universe.Institute_height=GetHeightSpace(Universe.Institute) //空間の高さ(y)

//コントロールパネルの設定数値を型に合わせてキャストする
Universe.Num_Department=CInt(Universe.Num_Department)

//エージェントを生成////////////////////////////////////
//Personal_Theme エージェントを一つ生成
Universe.Personal_Theme = CreateAgt(Universe.Themes.Personal_Theme)

//Department エージェントを生成
Universe.All_Departments = CreateAgtMulti(Universe.Institute.Department, Universe.Num_Department)
i = 0
For each a in Universe.All_Departments
  a.Speciality = (i mod Universe.Num_Specialities)
  i = i + 1
Next a

//Coordinator エージェントを生成
Universe.All_Coordinators = CreateAgtMulti(Universe.Institute.Coordinator, Universe.Num_Coordinator)

i = 1 //Coordinator に UID を付与(1~Num_Coordinator)
For each a in Universe.All_Coordinators
  a.UID = i
  i = i + 1
Next a
DuplicateAgtSet(Universe.All_CDs, Universe.All_Coordinators) //ナゾエラー対策

//Group エージェントを生成
For each a in Universe.All_Departments
  a.Groups = CreateAgtMulti(Universe.Institute.Group, Universe.Num_Group)
  For each b in a.Groups
    b.Speciality = a.Speciality
  Next b
Next a
MakeAgtSet(Universe.All_Groups, Universe.Institute.Group)

//Researcher エージェントを生成
For each a in Universe.All_Groups
  ClearAgtSet(agtset_temp)
  ClearAgtSet(a.Researchers)
  Universe.Create_Temp_Speciality = a.Speciality //下で作る Researcher エージェントはグループ a の Speciality にかたよる
  agtset_temp = CreateAgtMulti(Universe.Institute.Researcher, Universe.Num_Researcher)
  DuplicateAgtSet(a.Researchers, agtset_temp)
Next a
MakeAgtSet(Universe.All_Researchers, Universe.Institute.Researcher)

//END//エージェントを生成////////////////////////////////////

//各エージェントに、上の階層のエージェントと、その中での ID (ID_in_XX) をセット
//各 Group に、自身が属する Department をセット
For each a in Universe.All_Departments
  i = 0
  For each b in a.Groups
    b.Department = a
    b.ID_in_Department = i
    i = i + 1
  next b
Next a

```

```

//各 Researcher に、自身が属する Group と Department をセット
For each a in Universe.All_Groups
  i = 0
  For each b in a.Researchers
    b.Group = a
    b.Department = a.Department
    b.ID_in_Group = i
    i = i+1
  next b
Next a
//各 Department に、自身に属する Researchers をセット
For each a in Universe.All_Departments
  ClearAgtSet(agtset_temp)
  For each b in a.Groups
    JoinAgtSet(agtset_temp, b.Researchers)
  next b
  ClearAgtSet(a.Researchers)
  DuplicateAgtSet(a.Researchers,agtset_temp)
Next a

DuplicateAgtSet(Universe.All_DPs, Universe.All_Departments) //ナゾエラー対策

//部局ごとに色をわけて表示////////////////////////////////////
//色の設定
Universe.DepartmentColors(1)=COLOR_CYAN//奇数番目の部局の色
Universe.DepartmentColors(0)=COLOR_BLUE//偶数番目の部局の色

Dim Department_w as Integer
Dim Department_h as Integer
Universe.Department_width = CInt(Universe.Institute_width/Universe.Num_Department)
Department_w = Universe.Department_width
Department_h = Universe.Department_height
//部局ごとに色を設定
For i=1 to Universe.Num_Department
  For j = (i-1) * Department_w to i * Department_w -1
    For k=0 to Department_h
      if (i mod 2) == 1 then
        Universe.Institute.Department_color(j,k, 0) = Universe.DepartmentColors(1)
      else
        Universe.Institute.Department_color(j,k, 0) = Universe.DepartmentColors(0)
      end if
      Universe.Institute.Department_alpha(j, k, 0) = 90
    Next k
  Next j
next i
//丸め誤差のために余った領域を塗りつぶす
For j = Universe.Num_Department * Department_w to Universe.Institute_width -1
  For k=0 to Department_h
    if (Universe.Num_Department mod 2) == 1 then
      Universe.Institute.Department_color(j, k, 0) = Universe.DepartmentColors(1)
    else
      Universe.Institute.Department_color(j, k, 0) = Universe.DepartmentColors(0)
    end if
    Universe.Institute.Department_alpha(j, k, 0) = 90
  Next k
Next j
//コーディネータエリアの色を白に設定
For j=0 to Universe.Institute_width-1
  For k=Department_h+1 to Universe.Institute_height-1
    Universe.Institute.Department_color(j, k, 0) = COLOR_WHITE
  Next k
Next j
//END//部局ごとに色をわけて表示////////////////////////////////////

//各エージェントに位置をセット
//Department に位置をセット※Department エージェントは表示されない
For each a in Universe.All_Departments
  a.X = a.ID * Universe.Department_width
  a.Y = 0
Next a
//Group に位置をセット※Group エージェントは表示されない

```

```

For each a in Universe.All_Groups
  b = a.Department
  a.X = b.X
  a.Y = (a.ID_in_Department + 1) * Universe.Department_height / (Universe.Num_Group + 1)
Next a
//Researcher に位置をセット
For each a in Universe.All_Researchers
  b = a.Group
  a.X = b.X + (a.ID_in_Group + 1) * Universe.Department_width / (Universe.Num_Researcher + 1)
  a.Y = b.Y
Next a

//研究テーマエリアの設定////////////////////////////////////
Dim Themes_width as Integer
Dim Themes_height as Integer

Themes_width=GetWidthSpace(Universe.Themes)
Themes_height=GetHeightSpace(Universe.Themes)

//締め切り線を設定
For i=0 to Themes_width -1
  For j = 0 to Themes_height -10 - 1
    Universe.Themes.BGColor(i, j, 0) = COLOR_WHITE
  Next j
  For j = Themes_height - 10 to Themes_height - 1
    Universe.Themes.BGColor(i, j, 0) = COLOR_RED
  Next j
next i
}

Univ_Step_Begin{

Dim i as Integer
Dim j as Integer
Dim k as Integer

Dim a as Agt
Dim b as Agt
Dim c as Agt
Dim d as Agt

Dim agtset_temp as AgtSet
Dim target_agtset as AgtSet

Dim check as Integer

Dim target_speciality as Integer
Dim signal_line_value as Integer
Dim max_researchers as Integer

//研究テーマの追加////////////////////////////////////

//初期 Theme の設定※初回ステップのみ実行
if GetCountStep() == 1 then
//Theme エージェントを生成
For i = 0 to 9
  Universe.LANE = i //レーン番号をセット
  CreateAgt(Universe.Themes.Theme)
next i
end if

//全ての Theme を取得
ClearAgtSet(Universe.All_Themes)
ClearAgtSet(Universe.Open_Themes)
MakeAgtSet(Universe.All_Themes, Universe.Themes.Theme)
For each a in Universe.All_Themes
  if a.Last_Display == False then
    AddAgt(Universe.Open_Themes, a)
  end if
Next a

```

```

//Themes 空間の各レーンをチェックして、空いているレーンに一定の確率で Theme を追加
//空いている = 今追加しても Theme の表示がかぶらない(一番上の Theme の y 座標が 45+45 より大きい)
For i=0 to 9
  check = 0
  For each a in Universe.All_Themes
    if a.Lane == i And a.y <= 90 then
      check = 1
    end if
  Next a
  if check == 0 then
    Universe.LANE = i
    if Rnd() * 100 < Universe.P_theme then
      CreateAgt(Universe.Themes.Theme)
    end if
  end if
next i

//END/研究テーマの追加////////////////////////////////////

//Coordinator に情報発信対象をセット////////////////////////////////////
//現在 Coordinator にセットされているターゲットをクリア
For each a in Universe.All_CDs
  ClearAgtSet(a.Cast_Target)
next a

For each a in Universe.Open_Themes
  b = a.Coordinator
  target_speciality = Universe.Num_Specialities //存在しない専門性番号を入れておく (専門性番号は Num_Specialities-1
  まで)
  For i=0 to a.Req_specialities_num - 1
    if a.Each_specialities_numbers(a.Req_specialities(i)) < a.Limit_numbers(a.Req_specialities(i)) then
      target_speciality = a.Req_specialities(i)
    end if
  next i

  if Universe.CastMode == "グループ" then

    max_researchers = 0
    For each c in Universe.All_Groups
      agtset_temp = c.Researchers
      k = 0 //target_speciality を持ち、State=0 な研究者を数えて k に積算していく
      For each d in agtset_temp
        For i=0 to a.Req_specialities_num - 1
          if a.Each_specialities_numbers(a.Req_specialities(i)) < a.Limit_numbers(a.Req_specialities(i)) then //定員に達していな
            い専門性を持った研究者を数え上げる
              //target_speciality = a.Req_specialities(i)
              if d.Speciality == a.Req_specialities(i) And d.State == 0 then
                //if d.Speciality == a.Req_specialities(i) then
                  k = k + 1
                end if
              end if
            end if
          next i
        next d
      //k が現在の max よりも大きかった場合、max を入れ替える
      if k >= max_researchers then
        max_researchers = k
        target_agtset = agtset_temp
      end if
    next c
    //Theme:a に由来するターゲットを Cast_Target に追加する
    //重複を許す(JoinAgtSet)
    JoinAgtSet(b.Cast_Target, target_agtset)

  elseif Universe.CastMode == "部局" then
    max_researchers = 0
    For each c in Universe.All_DPs
      agtset_temp = c.Researchers
      k = 0 //target_speciality を持つ研究者を数えて k に積算していく
      For each d in agtset_temp

```

```

For i=0 to a.Req_specialities_num - 1
  if a.Each_specialities_numbers(a.Req_specialities(i)) < a.Limit_numbers(a.Req_specialities(i)) then //定員に達していない専門性を持った研究者を数え上げる
    //target_speciality = a.Req_specialities(i)
    if d.Speciality == a.Req_specialities(i) And d.State == 0 then
      //if d.Speciality == a.Req_specialities(i) then
        k = k + 1
      end if
    end if
  next i
next d
//k が現在の max よりも大きかった場合、max を入れ替える
if k >= max_researchers then
  max_researchers = k
  target_agtset = agtset_temp
end if
next c
//Theme:a に由来するターゲットを Cast_Target に追加する
//重複を許す(JoinAgtSet)
JoinAgtSet(b.Cast_Target, target_agtset)

else
//キャストモードが「全体」のときは全ての Researcher を対象にする
JoinAgtSet(b.Cast_Target, Universe.All_Researchers)
end if

//信号強度によって信号線の色を変える
signal_line_value = 128 * CountAgtSet(b.Cast_Target) / CountAgtSet(Universe.All_Researchers)
b.Signal_Line_Color = RGB(signal_line_value, signal_line_value, signal_line_value)

Next a

//END/Coordinator に情報発信対象をセット//////////////////////////////////////
}

Univ_Step_End{

Dim i as Integer
Dim j as Integer
Dim k as Integer

Dim a as Agt
Dim b as Agt
Dim c as Agt

Dim agtset_temp as AgtSet

//全ての Theme を取得
ClearAgtSet(Universe.All_Themes)
ClearAgtSet(Universe.Open_Themes)
MakeAgtSet(Universe.All_Themes, Universe.Themes.Theme)
For each a in Universe.All_Themes
  if a.Last_Display == False then
    AddAgt(Universe.Open_Themes, a)
  end if
Next a

//必要専門性それぞれの充足数の計算
For each a in Universe.Open_Themes
  For i = 0 to Universe.Num_Specialities - 1
    a.Each_specialities_numbers(i) = 0 //全て 0 にセット
  next i
  For each b in a.Researchers
    a.Each_specialities_numbers(b.Speciality) = a.Each_specialities_numbers(b.Speciality) + 1
  Next b
Next a

//申請が成立したテーマの処理
For each a in Universe.Open_Themes
  //必要専門性が揃っているかチェック
  k = 0 // k >= Req_specialities_num で OK

```

```

For i = 0 to a.Req_specialities_num - 1
  if a.Each_specialities_numbers(a.Req_specialities(i)) > 0 then
    //それぞれの必要専門性の充足数が0より大きい時、kをインクリメントする
    k = k + 1
  end if
next i
//必要研究者数と必要専門性が揃っていればOK
if CountAgtSet(a.Researchers) >= a.Req_number And k >= a.Req_specialities_num then
  a.Color = COLOR_BLUE
  a.Last_Display = True
  Universe.Matched_Themes_Numbers = Universe.Matched_Themes_Numbers + 1
  For each b in a.Researchers
    b.Learning = b.Learning + 1 //成功経験
    b.State = 2 //Stateを「2:研究取り組み期間」に更新
    b.Color = COLOR_WHITE //状態2のときは白色になる
    b.Period_count = a.Period //研究取り組み期間をセット
  Next b
  //コーディネータの担当テーマから該当テーマをはずす
  b = a.Coordinator
  DelAgtSet2( b.Themes, a)
end if
Next a

//締め切りを過ぎたテーマの処理
For each a in Universe.Open_Themes
  if a.Now_step > a.Deadline_step And a.Last_Display == False then
    a.Color = COLOR_RED
    a.Last_Display = True
    For each b in a.Researchers
      b.Learning = b.Learning - 1 //失敗経験
      b.Work_On = Universe.Personal_Theme //個人テーマにもどる
      b.Color = @GetSpecialityColor(b.Speciality) //個人テーマにもどる
      b.State = 0 //Stateを0:個人テーマに更新
    Next b
    //コーディネータの担当テーマから該当テーマをはずす
    b = a.Coordinator
    DelAgtSet2( b.Themes, a)
  end if
Next a

//Last_Countを過ぎたテーマを除去
For each a in Universe.All_Themes
  if a.Last_Count >= 1 then
    For each b in a.Req_specialities_Indicators
      DelAgt(b)
    Next b
    For each b in a.Researcher_Indicators
      DelAgt(b)
    Next b
    DelAgt(a.Req_number_Indicator)
    DelAgt(a.Coordinator_Indicator)
    DelAgt(a)
  end if
Next a

//アウトプット
//研究チーム成立率
Universe.Team_Building_Rate = Universe.Matched_Themes_Numbers / Universe.All_Themes_Numbers

//状態分布
Dim state0 as Integer
Dim state1 as Integer
Dim state2 as Integer
Dim N as Double

state0 = 0
state1 = 0
state2 = 0
N = CountAgtSet(Universe.All_Researchers)
For each a in Universe.All_Researchers
  if a.State == 0 then

```

```

state0 = state0 + 1
elseif a.State == 1 then
state1 = state1 + 1
elseif a.State == 2 then
state2 = state2 + 1
end if
Next a
Universe.State_0_count = state0 / N
Universe.State_1_count = state1 / N
Universe.State_2_count = state2 / N

//学習効果の出力////////////////////////////////////
Dim learning_max as Integer
Dim learning_min as Integer
Dim learning_total as Integer

learning_max = Universe.Institute.Researcher(0).Learning
learning_min = Universe.Institute.Researcher(0).Learning
learning_total = 0

For each a in Universe.All_Researchers
learning_total = learning_total + a.Learning
if a.Learning > learning_max then
learning_max = a.Learning
end if
if a.Learning < learning_min then
learning_min = a.Learning
end if
Next a

Universe.Learning_AVE = learning_total / CountAgtSet(Universe.All_Researchers)
Universe.Learning_MAX = learning_max
Universe.Learning_MIN = learning_min
//END・学習効果の出力////////////////////////////////////

//ファイル書き込み
Dim fileline as String
if Universe.File_Output == "する" then
fileline = GetCountStep() & "," & Universe.Team_Building_Rate & "," & Universe.All_Themes_Numbers & "," &
Universe.Matched_Themes_Numbers & "," & Universe.Learning_MAX & "," & Universe.Learning_AVE & "," &
Universe.Learning_MIN & "," & Universe.State_0_count & "," & Universe.State_1_count & "," & Universe.State_2_count
WriteLnFile(1, fileline)

end if
}

Univ_Finish{

PrintLn("最終チーム成立率 : " & Universe.Team_Building_Rate)

//ファイルを閉じる
if Universe.File_Output == "する" then
CloseFile(1)
end if

}
#end_rule UNIVERSE

//-----
// Agent Rule
//-----
#begin_rule UNIVERSE.INSTITUTE.COORDINATOR
Agt_Init{

My.SignalCapacity = 100
//位置をセット : 等間隔に一行に並ぶ
My.X = (My.ID+1) * Universe.Institute_width / (Universe.Num_Coordinator+1)
My.Y = Universe.Department_height + (Universe.Institute_height - Universe.Department_height) / 2

```



```

My.Num_Themes = 0
}

Agt_Step{
}
#end_rule UNIVERSE.INSTITUTE.COORDINATOR

//-----
// Agent Rule
//-----
#begin_rule UNIVERSE.INSTITUTE.RESEARCHER
Agt_Init{
Dim p as Double

//初期設定
My.Learning = 0
My.State = 0 //0:個人テーマ, 1:学際テーマにアブライ, 2:研究取り組み期間

//専門性を付加
//このバージョンでは, 001,111,123,030 のいずれかをランダムに付加(今回は真ん中のケタのみでマッチング判断)
p = Rnd()

if p < Universe.Distribution_p then
  My.Speciality = @IntUniRnd(0, Universe.Num_Specialities - 1)
else
  My.Speciality = Universe.Create_Temp_Speciality
end if

My.Color = @GetSpecialityColor(My.Speciality)

//取り組んでいるテーマを個人テーマにセット
My.Work_On = Universe.Personal_Theme
}

Agt_Step{

Dim agtset_temp as Agtset
Dim a as Agt
Dim b as Agt
Dim total_score as Double
Dim p as Double
Dim total_p as Double

Dim intention as Double
Dim feasibility as Double
Dim rest_num as Integer
Dim information as Double

//テーマ遷移処理////////////////////////////////////
if My.State == 0 then //現在個人テーマに取り組んでいるときのみ遷移する

//遷移先テーマの候補を挙げる
ClearAgtSet(agtset_temp)
//すでに定員が埋まっているときは候補からはずす(定員が埋まっていないもののみ候補にあげる)
For Each a in Universe.Open_Themes
  if CountAgtSet(a.Researchers) < a.Req_number And a.Each_specialities_numbers(My.Speciality) <
a.Limit_numbers(My.Speciality) then
    AddAgt(agtset_temp, a)
  end if
Next a
AddAgt(agtset_temp, Universe.Personal_Theme) //個人テーマを候補に加える

//各テーマ候補を評価する (Score をつける)

For Each a in agtset_temp
  if @AgtTypeComp2(a, "Universe.Themes.Personal_Theme") == 1 then
    //個人テーマのスコア
    a.Score = 30
  end if
Next a
}
}

```

```

else
//個人志向（個人の特性と経験から算出される）
intention = 0.5 + 0.1 * My.Learning
if intention > 1 then
intention = 1
elseif intention < 0.1 then
intention = 0.1
Else
end if

//実現可能性
rest_num = a.Req_number - CountAgtSet(a.Researchers)
if rest_num > 0 then
feasibility = 1/rest_num
end if

//コーディネータからの情報
b = a.Coordinator
if GetAgtEntry(b.Cast_Target, My) != -1 then
information = b.SignalCapacity / CountAgtSet(b.Cast_Target)
else
information = 0
end if

//スコアの算出
a.Score = a.Benefit * intention * feasibility * (1+information)

end if
Next a

//各テーマ候補への遷移確率を計算する////////
//（計算結果は各テーマの変数 P に格納）
//合計スコアを計算する
total_score = 0
For Each a in agtset_temp
total_score = total_score + a.Score
Next a
//正規化した確率を各テーマ候補の P に格納する
For Each a in agtset_temp
a.P = a.Score / total_score
Next a
//END/各テーマ候補への遷移確率を計算する////////

//遷移確率に従って取り組むテーマを遷移させる
//テーマ候補を遷移確率が高い順に並べ替える
SortAgtSet(agtset_temp,"P",False)
p = Rnd() //p に 0~1 の乱数をセット
total_p = 0
For Each a in agtset_temp
if total_p <= p And p < (total_p+a.P) then
My.Work_On = a
if @AgtTypeComp2(a, "Universe.Themes.Theme") == 1 then
AddAgt(a.Researchers, My)

//Researcher_Indicator を生成
b = CreateAgt(Universe.Themes.Researcher_Indicator)
b.Theme = a
AddAgt(a.Researcher_Indicators, b)
b.Color = @GetSpecialityColor(My.Speciality)
b.X = a.X - 45 + 90/(a.Req_number+1) * CountAgtSet(a.Researcher_Indicators)
b.Y = a.Y

//Institute 空間での色をグレイに変える
My.Color = RGB(128,128,128)

//State を「1:学際テーマにアブライ」に更新する
My.State = 1

end if
end if
total_p = total_p+ a.P
Next a

```

```

elseif My.State == 2 then //State が「2:研究取り組み期間」のときの処理
  My.Period_count = My.Period_count - 1
  if My.Period_count <= 0 then//My.Period_count が 0 になったら State を「0:個人テーマ」に戻す
    My.Work_On = Universe.Personal_Theme //個人テーマをセット
    My.Color = @GetSpecialityColor(My.Speciality) //個人テーマにもどる(表示色)
    My.State = 0
  end if
end if
//END・テーマ遷移処理////////////////////////////////////
}
#end_rule UNIVERSE.INSTITUTE.RESEARCHER

//-----
// Agent Rule
//-----
#begin_rule UNIVERSE.THEMES.THEME
Agt_Init{
Dim i as Integer
Dim j as Integer
Dim k as Integer
Dim p as Integer
Dim a as Agt
Dim b as Agt
Dim agtset_temp as AgtSet

//初期設定
My.Color = RGB(224,255,255)
My.Last_Display = False
My.Last_Count = 0
Universe.All_Themes_Numbers = Universe.All_Themes_Numbers + 1

//利得を割当
My.Benefit = 0.3 + Rnd() * (0.8-0.3)

//レーン番号をセット
My.Lane = Universe.LANE

//サイズと初期 y 座標を設定
My.Size = 90
My.X = 50 + Universe.LANE * 100
My.Y = 45

//締め切りを設定
My.Deadline_step = 8
My.Now_step = 0

//研究期間を設定
My.Period = 48

//必要人数と必要専門性種類数を生成
My.Req_number = @IntUniRnd(2,6)
My.Req_specialities_num = @IntUniRnd(1,2)

//必要専門性を生成////////////////////////////////////
//specialities 配列の中身をクリア
For i = 0 to GetArraySize(My.Req_specialities) - 1
  My.Req_specialities(i) = 0
next i

My.Req_specialities(0) = @IntUniRnd(0, Universe.Num_Specialities - 1)
For i = 1 to My.Req_specialities_num - 1
  k = 1 //最初のループを回すために k を 0 でない値にしておく
  Do While k != 0
    p = @IntUniRnd(0, Universe.Num_Specialities - 1)
    My.Req_specialities(i) = p
    k = 0
  For j = 0 to i-1
    if My.Req_specialities(j) == p then
      k = k + 1 //重複があった場合、k を 0 でない値にする
    end if
  Next j
Next i

```

```

    end if
  next j
Loop
next i

//それぞれの専門性の定員数をセット
For i = 0 to Universe.Num_Specialities - 1
  My.Limit_numbers(i) = 0 //全て 0 にセット
next i
if My.Req_specialities_num > 1 then //必要専門性が複数するとき
  For i = 0 to My.Req_specialities_num - 1
    My.Limit_numbers(My.Req_specialities(i)) = My.Req_number - 1 //必要専門性については定員 My.Req_number - 1 をセ
    ット
  next i
else
  My.Limit_numbers(My.Req_specialities(0)) = My.Req_number //必要専門性については定員 My.Req_number をセット
end if

//END/必要専門性を生成////////////////////////////////////

//Coordinator を配置////////////////////////////////////
//各 Coordinator が担当しているテーマの数を数え上げる -> Num_Themes
For Each a in Universe.All_CDs
  a.Num_Themes = CountAgtSet(a.Themes)
Next a
//All Coordinators を Num_Themes でソートしたものを agtset_temp に格納
ClearAgtSet(agtset_temp)
DuplicateAgtSet(agtset_temp, Universe.All_CDs)
SortAgtSet(agtset_temp, "Num_Themes", True)
//Num_Themes が最も小さい Coordinator をこの Theme の担当として割り振る
a = GetAgt(agtset_temp, 0)
AddAgt(a.Themes, My) //Coordinator の Themes にこの Theme を追加
My.Coordinator = a //この Theme の Coordinator に、割り振られた Coordinator を追加
//END/Coordinator を配置////////////////////////////////////

//Indicator を表示////////////////////////////////////
//Req_number_Indicator を生成して自身の Indicator にひも付け
a = CreateAgt(Universe.Themes.Req_number_Indicator)
a.Theme = My
My.Req_number_Indicator = a
a.Req_number = My.Req_number
a.X = My.X - 25
a.Y = My.Y - 25

//Req_specialities_Indicator を生成して自身の Indicators 集合にひも付け
ClearAgtSet(agtset_temp)
agtset_temp = CreateAgtMulti(Universe.Themes.Req_specialities_Indicator, My.Req_specialities_num)
i=0
For Each a in agtset_temp
  a.Theme = My
  a.Req_speciality = My.Req_specialities(i)
  a.X = My.X + 45/(My.Req_specialities_num+1) * (i+1)
  a.Y = My.Y - 25
  a.Color = @GetSpecialityColor(a.Req_speciality)
  i=i+1
Next a
DuplicateAgtSet(My.Req_specialities_Indicators, agtset_temp)

//Coordinator_Indicator を生成して自身の Indicator にひも付け
a = CreateAgt(Universe.Themes.Coordinator_Indicator)
a.Theme = My
My.Coordinator_Indicator = a
a.UID = My.Coordinator.UID
a.X = My.X - 25
a.Y = My.Y + 25
//ここまで Indicator を表示////////////////////////////////////
}

Agt_Step{

```

```
Dim a as Agt
//締め切りに向かって進んでいく////////////////////////////////////
if My.Last_Display == False then

    My.Y = 45 + My.Now_step * 480 / My.Deadline_step
    My.Now_step = My.Now_step + 1

    //Indicator も連動して進んでいく
    My.Req_number_Indicator.Y = My.Y - 25
    For Each a in My.Req_specialities_Indicators
        a.Y = My.Y - 25
    next a
    For Each a in My.Researcher_Indicators
        a.Y = My.Y
    next a
    My.Coordinator_Indicator.Y = My.Y + 25
    //END/締め切りに向かって進んでいく////////////////////////////////////
else

    //ラスト表示期間中
    My.Last_Count = My.Last_Count + 1
end if
}
#end rule UNIVERSE.THEMES.THEME
```

空間的競争モデルのソースコード

```
//=====/  
// artisoc /  
//=====/  
Version = 4.0;  
//-----  
// Agent Rule  
//-----  
#begin_rule UNIVERSE  
Univ_Init{  
  //初期設定  
  ClearConsoleScreen()  
  SetRandomSeed(Universe.randomSeed)  
  
  //距離の単位を設定 (Universe.distanceUnit=1 となる)  
  Universe.distanceUnit = (GetWidthSpace(Universe.map)^2 + GetHeightSpace(Universe.map)^2)^0.5 //空間の対角線を 1 に  
  設定  
  
  //価格競争の打ち切り処理に関するパラメータ  
  Universe.breakShFirst = 0.2  
  Universe.breakShSecond = 30  
  Universe.upperCountForPriceCompFirst = 10  
  Universe.upperCountForPriceCompSecond = Universe.breakShSecond * 100  
  
  //デバッグ用 state 変数  
  Universe.stateMNT = "Univ_Init"  
  
  //消費者を設置  
  SetCustomers()  
  
  //探索ポイントを配置  
  SetSearchPoints()  
  
  //店舗の配置  
  SetShops()  
  
  //ラウンド 0 の結果を更新  
  AssignCustomers()  
  For each one in Universe.allShops  
    one.optProfit = one.profit  
  Next one  
  
  //ラウンド 0 の history を生成  
  MakeHistory(Universe.shop0)  
  MakeHistory(Universe.shop1)  
}  
  
Univ_Step_Begin{  
  
  //Turn=0,1 が済んだら Round を+1 する  
  //Univ_Init での処理が Turn=0  
  If Universe.Turn == 0 then  
    Universe.Round = Universe.Round + 1  
  End if  
  
}  
  
Univ_Step_End{  
  //Turn を切り替える  
  Universe.Turn = 1 - Universe.Turn  
}  
  
Univ_Finish{  
}  
  
Sub SetCustomers(){  
  //消費者を作成, 設置する  
  Dim spaceWidth as Integer  
  Dim spaceHeight as Integer
```

```

Dim i as Integer
Dim j as Integer
Dim one as Agt
Dim one2 as Agt
Dim ones as AgtSet
Dim N as Integer
Dim Nw as Integer
Dim Nh as Integer
Dim d as Double
Dim a as Double
Dim c as Integer
Dim xi as Double
Dim yi as Double

```

```

spaceWidth = GetWidthSpace(Universe.map)
spaceHeight = GetHeightSpace(Universe.map)

```

```

If Universe.customerDistribution == "一様ランダム" then

```

```

    For i=0 to Universe.customerNum - 1 Step 1
        one = CreateAgt(Universe.map.customer)
        one.X = Rnd() * spaceWidth
        one.Y = Rnd() * spaceHeight
    Next i

```

```

Elseif Universe.customerDistribution == "一様等間隔" then

```

```

    d = CInt((spaceWidth * spaceHeight / Universe.customerNum)^0.5)
    Nw = CInt(spaceWidth / d)
    Nh = CInt(spaceHeight / d)
    N = Nw * Nh
    Universe.customerNum2 = N

```

```

    For i=1 to Nh Step 1
        For j=1 to Nw Step 1
            one = CreateAgt(Universe.map.customer)
            one.X = d * j - 0.5 * (d * (Nw+1) - spaceWidth)
            one.Y = d * i - 0.5 * (d * (Nh+1) - spaceHeight)
        Next j
    Next i

```

```

Elseif Universe.customerDistribution == "線上等間隔" then
//Y 方向中線に等間隔で並ぶ

```

```

    d = 0
    If Universe.customerNum > 1 then
        d = GetWidthSpace(Universe.map) / (Universe.customerNum - 1)
    End if

```

```

    For i=0 to Universe.customerNum - 1
        one = CreateAgt(Universe.map.customer)
        one.X = i * d
        one.Y = GetHeightSpace(Universe.map) / 2

```

```

//エージェント数=1 のときの対応 : このときのみ one.X=i*d(=0)ではない

```

```

    If Universe.customerNum == 1 then
        one.X = GetWidthSpace(Universe.map) / 2
    End if

```

```

    Next i

```

```

Elseif Universe.customerDistribution == "2次元正規分布" then

```

```

//Para1 で  $\sigma_x$ , Para2 で  $\sigma_y$  を指定する

```

```

//Para=0 で  $\sigma=0$ , Para=1 で  $\sigma_x=w$ ,  $\sigma_y=h$ 

```

```

    For i=0 to Universe.customerNum - 1
        one = CreateAgt(Universe.map.customer)
        //x 座標の決定
        xi = -1.0

```

```

        Do While (xi < 0 OR GetWidthSpace(Universe.map) < xi)

```

```

            xi = NormInv(Rnd(), GetWidthSpace(Universe.map)/2, GetWidthSpace(Universe.map) * Universe.cDistPara1) //3  $\sigma$ 

```

で端っこ

```

Loop
one.X = xi
//y 座標の決定
xi = -1.0
Do While (xi < 0 OR GetHeightSpace(Universe.map) < xi)
xi = NormInv(Rnd(), GetHeightSpace(Universe.map)/2, GetHeightSpace(Universe.map)* Universe.cDistPara2) //3
σ で端っこ
Loop
one.Y = xi
Next i

Elseif Universe.customerDistribution == "クラスター" then
//Para3 でクラスターの数を指定する(小数点以下切り捨て)
//Para1 で σ の値を指定する (Para1 で σ =w)
N = Int(Universe.cDistPara3)

For i = 0 to N - 1
one = CreateAgt(Universe.map.clusterPoint)
one.X = Rnd() * spaceWidth
one.Y = Rnd() * spaceHeight
Next i
MakeAgtSet(ones, Universe.map.clusterPoint)

For i=0 to Universe.customerNum - 1
one = CreateAgt(Universe.map.customer)
//x 座標の決定
c = 0
Do While (c == 0)
one2 = GetAgt(ones, Int(Rnd()*N))
xi = NormInv(Rnd(), one2.X, spaceWidth * Universe.cDistPara1)
yi = NormInv(Rnd(), one2.Y, spaceWidth * Universe.cDistPara1)
if 0<=xi And xi < spaceWidth And 0 <= yi And yi <= spaceHeight then
c = 1
Else
c = 0
End if

Loop
one.X = xi
one.Y = yi
Next i

Elseif Universe.customerDistribution == "局在" then
//Para1 で σ_x を指定する Para=1 で σ_x=w

For i=0 to Universe.customerNum - 1
one = CreateAgt(Universe.map.customer)
//x 座標の決定
xi = -1.0
Do While (xi < 0 OR GetWidthSpace(Universe.map) < xi)
xi = NormInv(Rnd(), 0, spaceWidth * Universe.cDistPara1)
if xi < 0 then
xi = xi + SpaceWidth
End if
Loop
one.X = xi
//y 座標の決定
one.Y = Rnd() * spaceHeight
Next i

Else

DisplayErrorMsg("SetCustomers/Universe.customerDistribution")

End if

MakeAgtSet(Universe.allCustomers, Universe.map.customer)
}

Sub SetSearchPoints(){
//探索点を作成, 設置する

```



```

Dim i as Integer
Dim j as Integer
Dim one as Agt
Dim dX as Double
Dim dY as Double

dX = 0
dY = 0
if Universe.searchPointNumX > 1 then
    dX = GetWidthSpace(Universe.map) / (Universe.searchPointNumX - 1)
End if
if Universe.searchPointNumY > 1 then
    dY = GetHeightSpace(Universe.map) / (Universe.searchPointNumY - 1)
End if

For i=0 to Universe.searchPointNumY - 1
    For j=0 to Universe.searchPointNumX - 1
        one = CreateAgt(Universe.map.searchPoint)
        one.X = j * dX
        one.Y = i * dY

        // 1行, 1列対応
        If Universe.searchPointNumX == 1 then
            one.X = GetWidthSpace(Universe.map) / 2
        End if
        If Universe.searchPointNumY == 1 then
            one.Y = GetHeightSpace(Universe.map) / 2
        End if

    Next j
Next i

MakeAgtSet(Universe.allSearchPoints, Universe.map.searchPoint)
Universe.searchPointNum = CountAgtSet(Universe.allSearchPoints)
}

Sub DisplayErrorMsg(msg as String){
//エラーメッセージを表示してシミュレーションを終了する
//msg は可能な限り, エラー箇所を特定できるような形で記述する

    PrintLn("エラー : " & msg)
    PrintLn("シミュレーションを終了します. ")
    ExitSimulation()
}

Sub SetShops(){
//店舗を生成して設置する

    Dim x as Double
    Dim y as Double
    Dim sp as Agt
    Dim spSet as AgtSet
    Dim one as Agt
    Dim one2 as Agt

    ClearAgtSet(spSet)
    DuplicateAgtSet(spSet, Universe.allSearchPoints)

    If Universe.shopInitLocation == "ランダム" then

        //shop0 の設置
        x = Rnd() * GetWidthSpace(Universe.map)
        y = Rnd() * GetHeightSpace(Universe.map)
        sp = GetNearestAgt(x, y, spSet, Universe.map)
        DelAgtSet2(spSet, sp)
        Universe.shop0 = CreateShop(sp, Universe.initPrice, 0, COLOR_BLUE)

        //shop1 の設置
        x = Rnd() * GetWidthSpace(Universe.map)
        y = Rnd() * GetHeightSpace(Universe.map)
        sp = GetNearestAgt(x, y, spSet, Universe.map)

```

```

Universe.shop1 = CreateShop(sp, Universe.initPrice, 1, COLOR_RED)

Elseif Universe.shopInitLocation == "固定" then

    //shop0 の設置
    x = 0.40 * GetWidthSpace(Universe.map)
    y = 0.50 * GetHeightSpace(Universe.map)
    sp = GetNearestAgt(x, y, spSet, Universe.map)
    DelAgtSet2(spSet, sp)
    Universe.shop0 = CreateShop(sp, Universe.initPrice, 0, COLOR_BLUE)

    //shop1 の設置
    x = 0.60 * GetWidthSpace(Universe.map)
    y = 0.50 * GetHeightSpace(Universe.map)
    sp = GetNearestAgt(x, y, spSet, Universe.map)
    Universe.shop1 = CreateShop(sp, Universe.initPrice, 1, COLOR_RED)

Else

    DisplayErrorMsg("SetShops/Universe.shopInitLocation")

End if

MakeAgtSet(Universe.allShops, Universe.map.shop)
}

Function CreateShop(sp as Agt, price as Double, group as Integer, color as Integer) as Agt{
    //店舗を生成する

    Dim one as Agt

    one = CreateAgt(Universe.map.shop)

    one.X = sp.X
    one.Y = sp.Y
    one.UID = group //このバージョンでは UID=group
    one.group = group
    one.price = price
    one.color = color
    one.searchPoint = sp
    one.optPrice = price
    one.optSP = sp

    return(one)
}

Function GetNearestAgt(x as Double, y as Double, agts as Agtset, map as Space) as Agt{
    //agts の中から, (x,y)に最も近いエージェントを選ぶ

    Dim one as Agt
    Dim nearestAgt as Agt
    Dim distance as Double
    Dim nearestDistance as Double

    nearestAgt = GetAgt(agts, 0)
    nearestDistance = MeasureDistance(x, y, nearestAgt.X, nearestAgt.Y, map) //初期値として agts の 0 番目エージェントまでの距離を代入
    For each one in agts
        distance = MeasureDistance(x, y, one.X, one.Y, map)
        if distance < nearestDistance then //現在の NearestDistance より近いやつが出てきたら, NearestDistance と nearestAgt を更新する
            nearestAgt = one
            nearestDistance = distance
        End if
    Next one

    return( nearestAgt)
}

Sub AssignCustomers() {
    //現在の shop の位置と価格を用いて, 各 customer に shop を割り振る(customer.selectGroup を更新する)

```

```

//割り振った後に shop.customerNum と shop.profit を更新する

Dim one As Agt
Dim selectedShop as Agt

//shop の customerNum を初期化
For each one in Universe.allShops
    one.customerNum = 0
Next one

//全ての customer に対してそれぞれ shop を割り振る&shop.customerNum を加算
For each one in Universe.allCustomers
    selectedShop = GetPreferShop(one)
    one.selectGroup = selectedShop.group
    selectedShop.customerNum = selectedShop.customerNum + 1
Next one

//shop.profit を更新する
For each one in Universe.allShops
    one.profit = one.price * one.customerNum
    if one.profit < 0 then
        DisplayErrorMsg("AssignCustomers/profit_<_0/state: " & Universe.stateMNT)
    End if
Next one
}

Function GetPreferShop(customer as Agt) as Agt{
//customer が選ぶ shop を決める
//このバージョンでは 2 店舗(shop0, shop1)の競争のみ有効

    Dim cost0 as Double
    Dim cost1 as Double
    Dim preferShop as Agt

    cost0 = CalcCost(customer, Universe.shop0)
    cost1 = CalcCost(customer, Universe.shop1)

    if cost0 < cost1 then
        preferShop = Universe.shop0
    ElseIf cost0 > cost1 then
        preferShop = Universe.shop1
    Else //cost が等しいとき
        if (Universe.divideFlag Mod 2) == 0 then
            preferShop = Universe.shop0
        Else
            preferShop = Universe.shop1
        End if
        Universe.divideFlag = Universe.divideFlag + 1
    End if

    return(preferShop)
}

Function CalcCost(customer as Agt, shop as Agt) as Double{
//設定した評価関数に従って評価をした値を返す関数
//引数 customer から見た引数 shop の評価を返す
//shop.price には、現在の価格を設定しておく

    Dim cost as double
    Dim distance as double

    distance = MeasureDistance(customer.X, customer.Y, shop.X, shop.Y, Universe.map) / Universe.distanceUnit

    if Universe.costFunctionMode == "p+tX" then
        cost = shop.price + Universe.t * distance
    elseif Universe.costFunctionMode == "p+tX^2" then
        cost = shop.price + Universe.t * distance^2
    else
        DisplayErrorMsg("CalcCost/Universe.costFunctionMode")
    End if
}

```

```

    return(cost)
}

Sub MakeHistory(shop as Agt){
//shop の現状態に対応した history エージェントをつくる

    Dim one as Agt

    one = CreateAgt(Universe.map.history)
    one.Round = Universe.Round
    one.group = shop.group
    one.color = shop.color
    one.X = shop.X
    one.Y = shop.Y
    one.price = shop.price
    one.searchPoint = shop.searchPoint
    one.customerNum = shop.customerNum
    one.profit = shop.profit

    //2 つ前の ID の history から矢印を引く
    If Universe.Round > 0 then
        ClearAgtSet(one.previous)
        AddAgt(one.previous, Universe.map.history(one.ID - 2))
    End if
}

Sub PriceComp(Me as Agt, Opp as Agt){
//価格競争を行い、その結果を Me と Opp の decisionPrice に格納
//Me:先行, Opp:後攻
    Dim breakFlag as Integer
    Dim count as Integer
    Dim decPriceMe as Double
    Dim decPriceOpp as Double
    Dim diff as Integer
    Dim diffPre as Integer
    Dim dP as Double

    //価格競争フェーズ 1
    Universe.stateMNT = "PriceComp/フェーズ 1"
    breakFlag = 0
    count = 0
    Me.decisionPrice = Me.optPrice
    Opp.decisionPrice = Opp.optPrice
    Do While breakFlag == 0
        decPriceMe = DecidePriceFirst(Me, Opp, Opp.decisionPrice) //Opp の価格が decisionPrice のとき, Me の価格を粗く
        (First)最適化する
        decPriceOpp = DecidePriceFirst(Opp, Me, decPriceMe)

        //変化が特定の範囲内におさまれば While を終了する
        If ( (abs(decPriceMe - Me.decisionPrice) / Me.decisionPrice) < Universe.breakShFirst) And ( (abs(decPriceOpp -
        Opp.decisionPrice) / Opp.decisionPrice) < Universe.breakShFirst) And count != 0 then
            breakFlag = 1
        End if
        if count > (Universe.upperCountForPriceCompFirst - 1) then
            breakFlag = 1
        End if

        //decisionPrice を更新
        Me.decisionPrice = decPriceMe
        Opp.decisionPrice = decPriceOpp
        count = count + 1
        //PrintLn(decPriceMe & " / " & decPriceOpp)

    Loop

    /*
    Me.price = Me.decisionPrice
    Opp.price = Opp.decisionPrice
    AssignCustomers()
    if Me.profit == 0 then

```

```

    Me.decisionPrice = Opp.decisionPrice
End if
if Opp.profit == 0 then
    Opp.decisionPrice = Me.decisionPrice
End if
*/

//価格競争フェーズ2 で使う価格ステップを算出
dP = (Me.decisionPrice + Opp.decisionPrice) / Universe.breakShSecond
if dP <= 0 then
    DisplayErrorMsg("PriceComp/dP_<= 0")
End if

//価格競争フェーズ2
Universe.stateMNT = "PriceComp/フェーズ2"
breakFlag = 0
diffPre = 0
count = 0
Do While breakFlag == 0
    diff = DecidePriceSecond(Me, Me.decisionPrice, dP, Opp, Opp.decisionPrice) //Opp の価格が decisionPrice のとき, Me
    の価格を細かく (Second)最適化する.結果は Me.decisionPrice に格納され, 戻り値は diff
    diff = DecidePriceSecond(Opp, Opp.decisionPrice, dP, Me, Me.decisionPrice)
    //Opp の diff が 0 または符号反転したところで終了
    If (diff == 0) OR (diff + diffPre) == 0 then
        breakFlag = 1
    End if
    if count > (Universe.upperCountForPriceCompSecond - 1) then
        DisplayErrorMsg("PriceComp フェーズ2/カウントオーバー")
    End if

    //diffPre を保存
    diffPre = diff
    count = count + 1
Loop
}

Function DecidePriceFirst(Me as Agt, Opp as Agt, OppPrice as Double) as Double{
//Opp の価格が OppPrice のとき, Me の価格を粗く最適化する
//戻り値は最適な Me の価格

Dim one as Agt
Dim profit as Double
Dim optProfitTemp as Double
Dim optPriceTemp as Double

SetResPrice(Me, Opp, OppPrice) //Opp の価格を固定して, Me に対する留保価格を各 customer にセット

//各 Customer の resPrice を取り出して, Me の価格を resPrice にしたときの profit を算出し, optProfitTemp と
optPriceTemp を更新する
optProfitTemp = -1 //暫定最高利益の初期値を設定. 取り得る値よりも低く設定.
For each one in Universe.allCustomers
    //従来の方法で算出
    /*Me.price = one.resPrice
    Opp.price = OppPrice
    AssignCustomers()
    profit = Me.profit
    */

    profit = CalcProfitWithResPrice(Me, one.resPrice) //高速手法
    If profit > optProfitTemp then
        optProfitTemp = profit
        optPriceTemp = one.resPrice
    End if
Next one

//profit が負になるときにエラーメッセージ
if optProfitTemp < 0 then
    DisplayErrorMsg("DecidePriceFirst/optProfitTemp_< 0")
End if

```

```

//戻り値をセット
return(optPriceTemp)
}

Function CalcProfitWithResPrice(Me as Agt, resPrice as Double) as Double{
//resPriceをつかった高速 assign によって Me の Profit を計算する
//Me, Opp の位置, Opp の価格は固定
//必ず SetResPrice()の後に実行しなければならない
//shop.price は使わない
Dim one as Agt
Dim n as Integer
Dim profit as Double

If resPrice <= 0 then
profit = 0
Else
n = 0
For each one in Universe.allCustomers
If resPrice <= one.resPrice then
n = n + 1
End if
Next one
profit = n * resPrice
if profit < 0 then //profit が負になるときにエラー
DisplayErrorMsg("CalcProfitWithResPrice/profit_<_0")
End if
End if
return(profit)
}

Function DecidePriceSecond(Me as Agt, MePrice as Double, dP as Double, Opp as Agt, OppPrice as Double) as Integer{
//Opp の価格が OppPrice のとき, Me の価格を細かく最適化する. 結果は Me.decisionPrice に格納され, 戻り値は変化
の符号

Dim optProfitTemp as Double
Dim optPriceTemp as Double
Dim diff as Integer

//diff=0 の場合を試行
Me.price = MePrice
Opp.price = OppPrice
AssignCustomers()
optProfitTemp = Me.profit
optPriceTemp = Me.price
diff = 0

//diff=-1 の場合を試行
Me.price = MePrice - dP
if Me.price < 0 then //MePrice < dP のときは Me.price を 0 にセット
Me.price = 0
End if
Opp.price = OppPrice
AssignCustomers()
If Me.profit > optProfitTemp then //profit が良くなった場合は optProfitTemp と diff を更新
optProfitTemp = Me.profit
optPriceTemp = Me.price
diff = -1
End if

//diff=+1 の場合を試行
Me.price = MePrice + dP
Opp.price = OppPrice
AssignCustomers()
If Me.profit > optProfitTemp then //profit が良くなった場合は optProfitTemp と diff を更新
optProfitTemp = Me.profit
optPriceTemp = Me.price
diff = 1
End if

//結果をセット
Me.decisionPrice = optPriceTemp

```

```

//戻り値をセット
return(diff)
}

Sub SetResPrice(Me as Agt, Opp as Agt, OppPrice as Double){
//全ての customer エージェントに、店舗 Me に対する resPrice をセットする
//このとき、Me の位置、Opp の位置、Opp の価格が固定されているとする

Dim one as Agt
Dim distanceMe as Double //顧客 one から Me までの距離
Dim distanceOpp as Double //顧客 one から Opp までの距離
Dim resPrice as Double

For each one in Universe.allCustomers

distanceMe = MeasureDistance(one.X, one.Y, Me.X, Me.Y, Universe.map) / Universe.distanceUnit
distanceOpp = MeasureDistance(one.X, one.Y, Opp.X, Opp.Y, Universe.map) / Universe.distanceUnit

If Universe.costFunctionMode == "p+tX" then
resPrice = Opp.price + Universe.t * (distanceOpp - distanceMe)
if resPrice < 0 then
one.resPrice = 0
Else
one.resPrice = resPrice
End if
Elseif Universe.costFunctionMode == "p+tX^2" then
resPrice = Opp.price + Universe.t * (distanceOpp^2 - distanceMe^2)
if resPrice < 0 then
one.resPrice = 0
Else
one.resPrice = resPrice
End if
Else
DisplayErrorMsg("SetResPrice/Universe.costFunctionMode")
End if

Next one
}

//Function RoundResPrice(x as Double) as Integer{
//x が非負なら小数点以下を切り下げる。また、x が負なら 0 を返す
// Dim y as Integer
// y = Int(x) + 1
// return(y)
//}

#end_rule UNIVERSE

//-----
// Agent Rule
//-----
#begin_rule UNIVERSE.MAP.SHOP
Agt_Init{
}

Agt_Step{

Dim Opp as Agt //自分ではない方の shop
Dim spSet as AgtSet //このターンで位置探索をする sp のリスト
Dim sp as Agt

//暫定最適値
Dim optSpTemp as Agt
Dim optPriceTemp as Double
Dim optProfitTemp as Double
Dim optOppPriceTemp as Double

```

```

Dim optOppProfitTemp as Double
Dim spSetCount as Integer

If My.UID == Universe.Turn then

    //自分ではない方の shop を特定
    Opp = Universe.map.shop(1 - My.ID)

    //探索候補 spSet から, Opp が乗っている sp を除く
    ClearAgtSet(spSet)
    DuplicateAgtSet(spSet, Universe.allSearchPoints)
    DelAgtSet2(spSet, Opp.searchPoint)
    spSetCount = CountAgtSet(spSet)

    //暫定最適値=現在の値
    @AssignCustomers()
    optProfitTemp = My.profit
    optSpTemp = My.searchPoint
    optPriceTemp = My.price
    optOppPriceTemp = Opp.price
    optOppProfitTemp = Opp.profit

    //位置候補を順にまわす
    ClearConsoleScreen()
    For each sp in spSet
        //My を sp の位置に移動する(仮移動)
        My.X = sp.X
        My.Y = sp.Y

        //価格競争
        //My と Opp の位置を固定して, 価格を決定する
        //価格競争の結果は My と Opp の decisionPrice に格納
        @PriceComp(My, Opp)

        My.price = My.decisionPrice
        Opp.price = Opp.decisionPrice

        Universe.stateMNT = "価格設定終了後"
        @AssignCustomers()

        if My.profit == 0 or Opp.profit == 0 then
            My.profit = -1.0
        End if

        //sp における価格競争の結果, 利益が暫定最高利益を超えた場合に暫定最高利益とその位置を更新する
        If My.profit > optProfitTemp then
            optSpTemp = sp
            optPriceTemp = My.price
            optProfitTemp = My.profit
            optOppPriceTemp = Opp.price
            optOppProfitTemp = Opp.profit
        End if

        //進捗率をコンソールに表示
        PrintLn( Int(sp.ID * 100 / spSetCount) & "%")
    Next one

    //optXXX を更新する
    My.optSP = optSpTemp
    My.optPrice = optPriceTemp
    My.optProfit = optProfitTemp
    Opp.optPrice = optOppPriceTemp
    Opp.optProfit = optOppProfitTemp

    //位置と価格を更新する
    My.X = optSpTemp.X
    My.Y = optSpTemp.Y
    My.searchPoint = My.optSP
    My.price = optPriceTemp
    Opp.price = optOppPriceTemp

    //Turn 終了時のシェア更新と履歴保存

```



```
Universe.stateMNT = "ターン終了時"  
@AssignCustomers()  
@MakeHistory(My)  
  
End if //If My.UID == Universe.Turn の END  
  
}  
#end_rule UNIVERSE.MAP.SHOP  
  
//-----  
// Agent Rule  
//-----  
#begin_rule UNIVERSE.MAP.CUSTOMER  
Agt_Init{  
  
}  
  
Agt_Step{  
  //shopを更新する (線の表示のため)  
  //これらが実行される前に最新の状態で AssignCustomers が実行されている必要がある  
  ClearAgtSet(My.shop)  
  AddAgt(My.shop, Universe.map.shop(My.selectGroup))  
}  
#end_rule UNIVERSE.MAP.CUSTOMER
```

分散型電力供給制御モデルのソースコード

```
//=====/  
// artisoc /  
//=====/  
Version = 3.5;  
//-----  
// Agent Rule  
//-----  
#begin_rule UNIVERSE  
Function distance_agt(agt1 as Agt, agt2 as Agt) as Double{  
// 2つのエージェントの間の距離を取得する関数  
  
//空間を取得する (agt1 の空間を取得して距離計算に使う)  
Dim sp as Space  
Dim agt_type as AgtType  
agt_type=SpecifyAgtType(agt1)  
sp=GetRideSpace(agt_type)  
  
//距離を計算する  
dim distance as Double  
distance=MeasureDistance(agt1.X, agt1.Y, agt2.X, agt2.Y, sp)  
  
return(distance)  
  
}  
  
//-----  
Function direction_agt(agt1 as Agt, agt2 as Agt) as Double{  
//agt1 から見た agt2 の角度を取得する関数[degree]  
  
//空間を取得する (agt1 の空間を取得して角度計算に使う)  
Dim sp as Space  
Dim agt_type as AgtType  
agt_type=SpecifyAgtType(agt1)  
sp=GetRideSpace(agt_type)  
  
//角度を計算する  
dim direction as Double  
direction=GetDirection(agt1.X, agt1.Y, agt2.X, agt2.Y, sp)  
  
return(direction)  
  
}  
  
//-----  
Function AgtTypeComp(agt1 as Agt, agt2 as Agt) as Integer{  
//agt1 と agt2 のタイプが一致するとき 1, 一致しないとき 0 を出力する  
  
Dim agt_type1 as String  
Dim agt_type2 as String  
Dim answer as Integer  
  
//それぞれのエージェントタイプを文字列として取得する  
agt_type1=CStr(SpecifyAgtType(agt1))  
agt_type2=CStr(SpecifyAgtType(agt2))  
  
//文字列が一致していれば 1, 一致していなければ 0 を出力する  
If StrComp(agt_type1,agt_type2)=0 then  
answer=1  
Else  
answer=0  
  
End if  
  
return(answer)  
  
}  
  
//-----  
Function AgtTypeComp2(agt1 as Agt, agt_type str as String) as Integer{
```

```

//agt1 のタイプ名が agt_type_str と一致するとき 1, 一致しないとき 0 を出力する
//agt_type_str は"Universe.空間名.種別名"という形で記述する

Dim agt_type as String
Dim answer as Integer

//それぞれのエージェントタイプを文字列として取得する
agt_type=CStr(SpecifyAgtType(agt1))

//文字列が一致していれば 1, 一致していなければ 0 を出力する
If StrComp(agt_type,agt_type_str)=0 then
    answer=1
Else
    answer=0

End if

return(answer)
}

//-----
Function eRnd(e as Double) as Double{
//-e?+e の範囲の値をランダム(一様分布)で返す

Dim output as Double

output = 2*e*(Rnd()-0.5)

return(output)
}

//-----
Function UniRnd(min as Double, max as Double) as Double{
//min?max の範囲の値をランダム(一様分布)で返す

Dim output as Double

output = min + Rnd() * (max - min)

return(output)
}

//-----
Function IntUniRnd(min as Integer, max as Integer) as Integer{
//min?max の範囲の整数値をランダム(一様分布)で返す

Dim min_temp as Double
Dim max_temp as Double
Dim rnd_temp as Double
Dim output as Integer

min_temp = CDbI(min) - 0.5
max_temp = CDbI(max) + 0.5
rnd_temp = min_temp + Rnd() * (max_temp - min_temp)

output = Round(rnd_temp)

return(output)
}

Univ_Init{

ClearConsoleScreen()//コンソールをクリア
SetRandomSeed(Universe.RandomSeed)

Universe.nego_step_number=100

```

```

//-----
//テンポラリー変数, インクリメント変数の宣言
//-----

//インクリメント変数
Dim i as Integer

//-----
//パラメータの設定&事前処理
//-----
Dim calc_time_min as Integer
Dim temp_array(1440) as Double
Dim temp_array2(240) as Double

for i=0 to GetArraySize(temp_array)-1
    temp_array(i)=0
next i

for i=0 to GetArraySize(temp_array2)-1
    temp_array2(i)=0
next i

Universe.DATA_time_min=temp_array
Universe.e_total=temp_array
Universe.DATA_time_hour=temp_array2

//空間のサイズを取得
Universe.space_width=GetWidthSpace(Universe.powergrid) //空間の幅(x)
Universe.space_height=GetHeightSpace(Universe.powergrid) //空間の高さ(y)

//エージェント数の設定値を整数にキャストする
Universe.area_num=CInt(Universe.area_num)
Universe.home_num=CInt(Universe.home_num)
Universe.solar_num=CInt(Universe.solar_num)

//-----
//少ない地域数・エージェント数設定時の処理
//-----
/*
if Universe.area_num <= 0 then
    PrintLn("地域数が不適当です(1以上の整数を入力して下さい)")
    ExitSimulation()
elseif Universe.area_num == 1 then
    Universe.clicked_area = 0
end if

if (Universe.area_num*Universe.home_num <= 4) Or (Universe.area_num*Universe.solar_num <= 4) then
    PrintLn("エージェント数が少なすぎます。")
    PrintLn("全体の供給家数と需要家数がそれぞれ5以上となるように設定して下さい")
    ExitSimulation()
end if
*/

//-----
//エージェントの生成と配置
//-----

//area エージェントの生成と配置
For i=1 to Universe.area_num
    CreateAgt(Universe.powergrid.area)
next i

//ファイル書き込み準備
Dim filename as String
Dim condition as String
Dim label_lamda as String
Dim label_G as String
Dim label_xw as String
Dim label_theta as String
Dim mode_string as String

```

```

if Universe.File_Output == "する" then

    condition = "地域数=" & Universe.area_num & "," & "供給家数=" & Universe.solar_num & "," & "グループ数=" &
    Universe.home_num & "," & "α=" & Universe.alpha & "," & "β=" & Universe.beta_d & "," & "ε=" & Universe.gamma & ","
    & "乱数シード=" & Universe.RandomSeed

    filename = "DATA/lamda.dat"
    label_lamda
    "s,lamda0,lamda1,lamda2,lamda3,lamda4,lamda5,lamda6,lamda7,lamda8,lamda9,lamda10,lamda11,lamda12,lamda13"
    OpenFile(filename, 1, 2) //上書きモードでオープン
    WriteLnFile(1, condition) //計算条件を書き込む
    WriteLnFile(1, label_lamda) //データラベルを書き込む
    CloseFile(1)
    OpenFile(filename, 1, 3) //追記モードでオープン

    filename = "DATA/G.dat"
    label_G = "hour,G0,G1,G2,G3,G4,G5,G6,G7,G8,G9,G10,G11,G12,G13"
    OpenFile(filename, 2, 2) //上書きモードでオープン
    WriteLnFile(2, condition) //計算条件を書き込む
    WriteLnFile(2, label_G) //データラベルを書き込む
    CloseFile(2)
    OpenFile(filename, 2, 3) //追記モードでオープン

    filename = "DATA/xw.dat"
    label_xw = "hour,home_x,home_w,solar_x,solar_w,large_x,large_w,trans_x,trans_w_high,trans_w_low"
    OpenFile(filename, 3, 2) //上書きモードでオープン
    WriteLnFile(3, condition) //計算条件を書き込む
    WriteLnFile(3, label_xw) //データラベルを書き込む
    CloseFile(3)
    OpenFile(filename, 3, 3) //追記モードでオープン

end if

}

Univ_Step_Begin{

//-----
//テンポラリー変数, インクリメント変数の宣言
//-----
Dim r_max as Double
Dim terminal_position as Double
Dim r as Double
Dim theta as Double
Dim temp_x as Double
Dim temp_y as Double
Dim area_set as AgtSet
Dim temp_agt as Agt
Dim temp_agt2 as Agt
Dim temp_agt3 as Agt
Dim temp_agt4 as Agt
Dim temp_agt5 as Agt
Dim player_agtset as AgtSet
Dim temp_agtset as AgtSet
Dim num_temp as Integer
Dim check_counter as Integer
//インクリメント変数
Dim i as Integer
Dim j as Integer
Dim k as Integer //flag
Dim ct as Integer
Dim loop_ct as Integer
Dim one as Agt
Dim one2 as Agt
Dim one3 as Agt
//ファイル書き込み(theta)準備
Dim filename as String
Dim condition as String
Dim label_theta as String

//-----

```

```

//パラメータの設定&事前処理
//-----
Universe.min_range_between_area=30
r_max=10 //地域の中に需要供給エージェントを配置するときの範囲（地域中心から半径 r_max 以内）
terminal_position=2/3

//-----
//エージェントの生成と配置
//エージェントの生成と配置は初回ステップのみ実行する
//-----

if Universe.count== 0 Then //初回ステップのみ

//-----
//地域エージェントを配置する
//-----
MakeAgtSet(area_set, Universe.powergrid.area) //地域エージェントを area_set に格納
RandomPutAgtset(area_set) //地域エージェントをランダムに配置(とりあえず配置)

//地域同士が重なる場合や孤立する場合を修正する
loop_ct=0 //ループカウンタ
Do While 1
  check_counter=0
  //地域同士が重なる場合は check_counter を+1 して該当する地域を再ランダム配置
  //地域間の距離が min_range_between_area 以下の場合、重なっていると判断する
  For each one in area_set
    MakeOneAgtSetAroundPosition(temp_agtset,Universe.powergrid, one.X, one.Y, one.Layer,
Universe.min_range_between_area,Universe.powergrid.area)
    DelAgtSet2(temp_agtset, one)
    ct=CountAgtSet(temp_agtset)
    if ct>0 then //再ランダム配置
      check_counter=check_counter+1
      one.X=Universe.space_width*Rnd()
      one.Y=Universe.space_height*Rnd()
    End if
  next one
  //孤立地域がある場合は check_counter を+1 して該当する地域を再ランダム配
  For each one in area_set
    MakeOneAgtSetAroundPosition(temp_agtset,Universe.powergrid, one.X, one.Y, one.Layer,
Universe.link_range_area,Universe.powergrid.area)
    DelAgtSet2(temp_agtset, one)
    ct=CountAgtSet(temp_agtset)
    if ct==0 then //再ランダム配置
      check_counter=check_counter+1
      one.X=Universe.space_width*Rnd()
      one.Y=Universe.space_height*Rnd()
    End if
  next one
  //重なり地域も孤立地域もない場合(check_counter=0 のとき)Break する
  if check_counter==0 then
    PrintLn("地域の配置を完了しました")
    Break
  End if
  //再配置の試行回数が一定数以上になった場合も Break する(重複、孤立地域は諦める)
  loop_ct=loop_ct+1
  if loop_ct>=30000 then
    PrintLn("地域の配置に失敗しました(孤立地域があります)")
    Break
  End if
Loop

//-----
//地域エージェント間にリンク(送電線)を生成する
//-----
ClearAgtSet(area_set)
MakeAgtSet(area_set, Universe.powergrid.area)

//地域エージェントに serial を付与する(serial=1,2,...)
i=1 //インクリメント変数の初期化
For each one in area_set
  one.serial=i
  i=i+1

```

```

next one

//隣接する地域を link エージェントセットに格納
For each one in area_set
  ClearAgtset (one.link)
  MakeOneAgtSetAroundPosition(one.link, Universe.powergrid, one.X, one.Y, one.Layer, Universe.link_range_area,
  Universe.powergrid.area) //自分のまわりの視野内にあるエージェント取得
  DelAgtSet2(one.link, one) //link から自分自身を削除
next one

//-----
//隣接する地域の間送電エージェント(trans)を生成して trans_in_set(AgtSet)または trans_out_set(AgtSet)に格納
//serial が大きい地域から小さい地域へ流れがあるときを正とする
//<地域 A>→<送電家 Z>→<地域 B>のように電力の流れがあるとすると
//・地域 A の serial(Int)が地域 B の serial より大きい
//・送電家 Z は地域 A の trans_out_set(AgtSet), 地域 B の trans_in_set(AgtSet)に入る
//・送電家 Z の from_player(Agt)は地域 A, 送電家 Z の to_player(Agt)は地域 B
//-----

For each one in area_set //地域エージェントそれぞれに(one)
  For each one2 in one.link //各地域エージェント(one)の隣接地域それぞれに(one2) (one:地域 A, one2:地域 B)
    if one.serial>one2.serial then //serial が大きいほうから小さい方に向けて処理をすれば十分 (2回処理する必要はない)
      //-----
      //送電エージェントを生成して配置する
      //-----
      temp_agt=CreateAgt(Universe.powergrid.trans) //送電エージェントを生成

      //20160802Simulation
      if (one.ID == 2 And one2.ID == 8) OR (one.ID == 8 And one2.ID == 2) then
        Universe.op_trans = temp_agt
      end if

      AddAgt(one.trans_out_set, temp_agt) //送電エージェントを serial が大きい方のエージェント(one)の trans_out_set に
      格納
      AddAgt(one2.trans_in_set, temp_agt) //送電エージェントを serial が小さい方のエージェント(one)の trans_in_set に
      格納
      temp_agt.from_player=one //送電エージェントから見て上流側の地域を from_player に格納
      temp_agt.to_player=one2 //送電エージェントから見て下流側の地域を to_player に格納
      //送電エージェントを配置. ただし空間出力には表示しない(送電線のみ表示)
      temp_agt.X=(one.X+one2.X)/2
      temp_agt.Y=(one.Y+one2.Y)/2

      //-----
      //送電ターミナルを生成して配置する
      //地域 A 側のターミナルは type="from"(Str)で, 送電家 Z の from_terminal(Agt)に相当
      //地域 B 側のターミナルは type="to"(Str)で, 送電家の to_terminal(Agt)に相当
      //ターミナルはそれぞれの地域の trans_terminals(AgtSet)に格納
      //ターミナルの trans(Agt)にそれぞれ紐づく送電家を格納
      //-----
      r=r_max*terminal_position
      //地域 A 側ターミナルの配置
      temp_agt2=CreateAgt(Universe.powergrid.trans_terminal) //地域 A 側ターミナルを生成
      temp_agt2.trans=temp_agt //地域 A 側ターミナルの trans に送電家 Z を格納
      temp_agt2.area=one //地域 A 側ターミナルの地域を指定(地域 A)
      temp_agt.from_terminal=temp_agt2 //送電家 Z の from_terminal に地域 A 側ターミナルを格納
      theta=@direction_agt(one, one2)
      temp_agt2.X=one.X+r*cos(theta/180*PI())
      temp_agt2.Y=one.Y+r*sin(theta/180*PI())
      AddAgt(one.trans_terminals, temp_agt2) //地域 A の trans_terminals に格納
      //地域 B 側ターミナルの配置
      temp_agt3=CreateAgt(Universe.powergrid.trans_terminal) //地域 B 側ターミナルを生成
      temp_agt3.trans=temp_agt //地域 B 側ターミナルの trans に送電家 Z を格納
      temp_agt3.area=one2 //地域 B 側ターミナルの地域を指定(地域 A)
      temp_agt.to_terminal=temp_agt3 //送電家 Z の to_terminal に地域 B 側ターミナルを格納
      theta=@direction_agt(one2, one)
      temp_agt3.X=one2.X+r*cos(theta/180*PI())
      temp_agt3.Y=one2.Y+r*sin(theta/180*PI())
      AddAgt(one2.trans_terminals, temp_agt3) //地域 B の trans_terminals に格納
      //ターミナル間に線を引く(ベアリング)
      AddAgt(temp_agt2.pair, temp_agt3)

```

```

    AddAgt(temp_agt3.pair, temp_agt2)
  end if
next one2
next one

//-----
//各地域に需要供給家エージェントを配置
//各種エージェントをランダムに配置し、隣接するものを link(AgtSet)に入れる(価格調整情報網)
//-----
ClearAgtSet(Universe.theta_output_agts)
For each one in area_set //各地域ごとに
  ClearAgtSet(one.members) //members を初期化
  ClearAgtSet(player_agtset) //player_agtset を初期化
  ClearAgtSet(temp_agtset) //temp_agtset を初期化

  //home エージェントの配置
  num_temp=IntUniRnd(Universe.home_num, Universe.home_num_max) //この地域の home 数を min~max の間でランダムに決定
  SetPlayers("home", num_temp, r_max, one, Universe.space_width, Universe.space_height)
  one.home_num = num_temp //area エージェントに home 数の情報を入れる

  //solar エージェントの配置
  num_temp=IntUniRnd(Universe.solar_num, Universe.solar_num_max) //この地域の solar 数を min~max の間でランダムに決定
  SetPlayers("solar", num_temp, r_max, one, Universe.space_width, Universe.space_height)
  one.solar_num = num_temp //area エージェントに solar 数の情報を入れる

  //player_agtset に需要供給家エージェントをリストアップ
  For each one2 in one.members
    AddAgt(player_agtset, one2)
  next one2
  For each one2 in one.trans_terminals
    AddAgt(player_agtset, one2)
  next one2

  //player に UID(地域ごと)を振る
  i=0
  For each one2 in player_agtset
    one2.UID = i
    i= i+1
  next one2

  //孤立エージェントを修正する
  loop_ct=0 //ループカウンタをリセット
  Do While 1
    check_counter=0
    //孤立エージェントがある場合は check_counter を+1 して該当するエージェントを再ランダム配置
    ClearAgtset(temp_agtset)
    For each one2 in player_agtset
      MakeAllAgtSetAroundPosition(temp_agtset,Universe.powergrid, one2.X, one2.Y, one2.Layer, Universe.link_range_player)
      DelAgtSet2(temp_agtset, one2) //自分自身を除去
      DelAgtSet(temp_agtset, area_set) //エリアエージェントを除去

      ct=CountAgtSet(temp_agtset)
      if ct=0 then //再ランダム配置
        check_counter=check_counter+1
        r=r_max*Rnd()
        theta=2*PI()*Rnd()

        temp_x=one.X+r*cos(theta)
        temp_y=one.Y+r*sin(theta)

        //空間外に配置されるのを防ぐ
        if temp_x < 0 then
          temp_x = 0
        end if
        if temp_x >Universe.space_width then
          temp_x = Universe.space_width
        end if
        if temp_y < 0 then
          temp_y = 0
        end if

```



```

if temp_y > Universe.space_height then
    temp_y = Universe.space_height
end if

one2.X=temp_x
one2.Y=temp_y
End if
next one2
//孤立エージェントがない場合(check_counter=0 のとき)Break する
if check_counter=0 then
    PrintLn("地域" & CStr(one.serial) & "の需要供給家の配置を完了しました")
    Break
End if
//再配置の試行回数が一定数以上になった場合も Break する(重複, 孤立地域は諦める)
loop_ct=loop_ct+1
if loop_ct>=10000 then
    PrintLn("地域" & CStr(one.serial) & "の需要供給家の配置に失敗しました(孤立エージェントがあります)")
    Break
End if
Loop

//隣接する需要供給家を link に格納する(赤線)
For each one2 in player_agtset
    ClearAgtset(one2.link)
    MakeAllAgtSetAroundPosition(one2.link, Universe.powergrid, one2.X, one2.Y, one2.Layer, Universe.link_range_player)
    DelAgtSet2(one2.link, one2) //自分自身を除去
    DelAgtSet(one2.link, area_set) //エリアエージェントを除去
    DelAgtSet(one2.link, one.trans_out_set) //トランスエージェントを除去
    DelAgtSet(one2.link, one.trans_in_set) //トランスエージェントを除去
    SortAgtSet(one2.link, "UID", True) //link 集合の中のエージェントを UID でソート
next one2

next one //各地域ごとに

if Universe.File_Output == "する" then

    condition = "地域数=" & Universe.area_num & "," & "供給家数=" & Universe.solar_num & "," & "グループ数=" &
    Universe.home_num & "," & "α=" & Universe.alpha & "," & "β=" & Universe.beta_d & "," & "ε=" & Universe.gamma & ","
    & "乱数シー ド=" & Universe.RandomSeed

    filename = "DATA/theta.dat"

    label_theta = "step"
    SortAgtSet(Universe.theta_output_agts, "UID", True)
    For i = 0 to (Universe.powergrid.area(Universe.File_Output_ID).home_num +
    Universe.powergrid.area(Universe.File_Output_ID).solar_num - 1)
        one = GetAgt(Universe.theta_output_agts, i)
        if AgtTypeComp2(one, "Universe.powergrid.home") == 1 then
            if one.large == 0 then
                label_theta = label_theta & ",h" & one.UID
            else
                label_theta = label_theta & ",L" & one.UID
            end if
        Else
            label_theta = label_theta & ",s" & one.UID
        End if
    next i

    OpenFile(filename, 4, 2) //上書きモードでオープン
    WriteLnFile(4, condition) //計算条件を書き込む
    WriteLnFile(4, label_theta) //データラベルを書き込む
    CloseFile(4)
    OpenFile(filename, 4, 3) //追記モードでオープン

end if

End if //初回ステップのみ各エージェントの生成配置

Dim clicked_x as Double
Dim clicked_y as Double
Dim clicked_map as String

```

```

//-----
//インタラクティブ表示
//-----
//マップをクリックすると地域の情報をコンソールに表示する
if IsMouseClickedOnMap() then
  GetClickedMapPosition(clicked_x, clicked_y, clicked_map)

  ClearAgtSet(area_set)
  MakeAgtSet(area_set, Universe.powergrid.area)

  For Each one in area_set
    If MeasureDistance(one.X, one.Y, clicked_x, clicked_y, Universe.powergrid) < 15 Then
      Universe.clicked_area=one.ID
    End If
  Next one

end if
}

//-----
//独自定義関数
//-----
//SetPlayers(需要供給エージェントタイプ(文字列), 生成配置するエージェント数, 配置するエリア半径, 配置する地域エージェント)
//需要供給エージェントを生成して地域内のランダムな位置に配置する関数
//戻り値: 生成したエージェントから構成される AgtSet
Function SetPlayers(type as String, n as Integer, r_max as Double, area as Agt, x_max as Double, y_max as Double) as AgtSet
{
  Dim agt_type as AgtType
  Dim temp_agt as Agt
  Dim r as Double
  Dim theta as Double
  Dim i as Integer
  Dim k as Integer //large のセットのため
  Dim Comp as Integer
  Dim Comp2 as Integer
  Dim output as AgtSet
  Dim temp_x as Double
  Dim temp_y as Double
  Dim temp_D3 as Double
  Dim ave_D3 as Double

  //フルバスのエージェントタイプを取得
  type="Universe.powergrid." & type
  agt_type=CAgtType(type)
  Comp = StrComp(type, "Universe.powergrid.home")
  Comp2 = StrComp(type, "Universe.powergrid.solar")

  k=0
  ave_D3 = 0
  temp_D3 = 0
  //n コのエージェント生成して配置
  For i=1 to n

    temp_agt =CreateAgt(agt_type)
    temp_D3 = temp_agt.D3

    //home_rep の登録
    if (Comp == 0) And (k == 1) then
      temp_agt.C3 = Universe.k * 1.0
      temp_agt.C4 = Universe.k * 1.0
      temp_agt.D1 = Universe.E * 0.2
      temp_agt.D2 = Universe.E * 0.1
      temp_agt.D3 = Universe.E
      temp_D3 = temp_agt.D3
      temp_agt.x_i = -temp_agt.D3
      k = k+1
      PrintLn("地域" & area.ID & "の home_rep をセット・ID : " & temp_agt.ID)
    if area.ID == Universe.File_Output_ID then

```

```

    Universe.op_home = temp_agt
  end if
end if

//large Supplier の登録
if (Comp == 0) And (k == 0) then
  temp_agt.large = 1
  k = k+1
  PrintLn("地域" & area.ID & "の Large Supplier をセット・ID : " & temp_agt.ID)
  temp_D3 = 0
  if area.ID == Universe.File_Output_ID then
    Universe.op_large = temp_agt
  end if
end if

//solar_rep の登録
if (Comp2 == 0) And (k == 0) then
  temp_agt.C3 = Universe.k * 1
  temp_agt.C4 = Universe.k * 1
  temp_agt.D1 = Universe.E * 0.3
  temp_agt.x_i = 0
  k = k+1
  PrintLn("地域" & area.ID & "の solar_rep をセット・ID : " & temp_agt.ID)
  if area.ID == Universe.File_Output_ID then
    Universe.op_solar = temp_agt
  end if
end if

//位置決め(ランダム)
r=r_max*Rnd()
theta=2*PI()*Rnd()

temp_x=area.X+r*cos(theta)
temp_y=area.Y+r*sin(theta)

//空間外に配置されるのを防ぐ
if temp_x < 0 then
  temp_x = 0
end if
if temp_x > x_max then
  temp_x = x_max
end if
if temp_y < 0 then
  temp_y = 0
end if
if temp_y > y_max then
  temp_y = y_max
end if

temp_agt.X=temp_x
temp_agt.Y=temp_y
//該当地域の members に登録
AddAgt(area.members, temp_agt)
if area.ID == Universe.File_Output_ID then
  AddAgt(Universe.theta_output_agts, temp_agt)
end if

//戻り値 AgtSet に登録
AddAgt(output, temp_agt)
//地域を登録
temp_agt.area=area

ave_D3 = ave_D3 + temp_D3
next i

ave_D3 = ave_D3 / (n-1)
if Comp == 0 then
  area.ave_D3 = ave_D3
end if

return(output)
} //SetPlayers の終了

```

```

Univ_Step_End{
//-----
//変数の宣言
//-----
Dim temp_agtset as AgtSet
Dim temp_agtset2 as AgtSet
Dim area_agtset as AgtSet
Dim player_agtset as AgtSet //home,solar,trans
Dim negotiator_agtset as AgtSet //home,solar,trans_terminal
Dim count_min as Integer
Dim count_second as Integer
Dim max_agt as Agt
Dim min_agt as Agt
Dim i as Integer
Dim file_hour as Double
Dim one as Agt

//-----
//パラメータの設定&事前処理
//-----
count_min=Universe.count*0.25 //離散時間を調整
count_second=Universe.count*15

ClearAgtset (area_agtset)
ClearAgtset(player_agtset)
ClearAgtset (negotiator_agtset )
ClearAgtset (temp_agtset)
ClearAgtset (temp_agtset2)

//全ての地域を取得
MakeAgtSet(area_agtset , Universe.powergrid.area)

//全てのプレイヤーを取得
MakeAgtSet(player_agtset, Universe.powergrid.home) //home を追加
MakeAgtSet(temp_agtset, Universe.powergrid.solar)
JoinAgtSet(player_agtset, temp_agtset) //solar を追加
MakeAgtSet(temp_agtset, Universe.powergrid.trans)
JoinAgtSet(player_agtset, temp_agtset) //trans を追加
PurifyAgtSet(player_agtset, player_agtset) //重複を削除

//全ての交渉プレイヤーを取得
MakeAgtSet(negotiator_agtset, Universe.powergrid.home) //home を追加
MakeAgtSet(temp_agtset, Universe.powergrid.solar)
JoinAgtSet(negotiator_agtset, temp_agtset) //solar を追加
MakeAgtSet(temp_agtset, Universe.powergrid.trans_terminal)
JoinAgtSet(negotiator_agtset, temp_agtset) //trans_terminal を追加
PurifyAgtSet(negotiator_agtset, negotiator_agtset) //重複を削除

//-----
//通常モード処理
//-----

Dim fileline as String
Dim G_output as Double
Dim lamda_output as Double
Dim temp_area as Agt

//ファイルアウトプット用
Dim temp_members as AgtSet
Dim temp_member as Agt
Dim home_x as Double
Dim home_w as Double
Dim solar_x as Double
Dim solar_w as Double
Dim wind_x as Double
Dim wind_w as Double
Dim large_x as Double
Dim large_w as Double
Dim trans_x as Double
Dim trans_w_h as Double

```

```

Dim trans_w_1 as Double
Dim lamda0 as Double
Dim lamda1 as Double
Dim lamda2 as Double
Dim lamda3 as Double
Dim lamda4 as Double
Dim lamda5 as Double
Dim lamda6 as Double
Dim lamda7 as Double
Dim lamda8 as Double
Dim lamda9 as Double
Dim lamda10 as Double
Dim lamda11 as Double
Dim lamda12 as Double
Dim lamda13 as Double
Dim aaa as Agt

if Universe.nego_mode==false then
//-----
//状態量の更新
//-----
//停滞量の総量を更新
Universe.e_total=0
For each one in area_agtset
  Universe.e_total=Universe.e_total+one.G_i
next one

//目的関数を更新
Universe.U_total=0
For each one in player_agtset
  Universe.U_total=Universe.U_total+one.U
next one

//ラムダを更新
For each one in area_agtset
  one.lamda=one.lamda_next
next one

//x_iを更新
For each one in player_agtset
  one.x_i=one.x_i_next
next one

//-----
//時刻の取得
//-----
Universe.time_second=count_second mod 60
Universe.time_min=count_min mod 60
Universe.time_day=count_min ¥ 1440
Universe.time_hour=(count_min mod 1440) ¥ 60
Universe.time_hour_conti = CDbl(Universe.time_hour) + CDbl(Universe.time_min) / 60.0
Universe.mode="通常モード"

Universe.DATA_time_min(Universe.count)=count_min
Universe.DATA_e_total(Universe.count)=Universe.e_total
Universe.DATA_time_hour(Universe.s)=Universe.s

//negoモードへの切り替え
if Universe.time_min == 0 And Universe.time_second == 0 And Universe.step_total !=0 then
  Universe.ct_nego=0
  Universe.nego_mode=true

end if

if Universe.File_Output == "する" then
  file_hour = Universe.count / (4 * 60)
  //-----G.datの出力-----
  fileline = file_hour & "," & Universe.powergrid.area(0).G_i & "," & Universe.powergrid.area(1).G_i & "," &
  Universe.powergrid.area(2).G_i & "," & Universe.powergrid.area(3).G_i & "," & Universe.powergrid.area(4).G_i & "," &

```

```

Universe.powergrid.area(5).G_i & "," & Universe.powergrid.area(6).G_i & "," & Universe.powergrid.area(7).G_i & "," &
Universe.powergrid.area(8).G_i & "," & Universe.powergrid.area(9).G_i & "," & Universe.powergrid.area(10).G_i & "," &
Universe.powergrid.area(11).G_i & "," & Universe.powergrid.area(12).G_i & "," & Universe.powergrid.area(13).G_i
WriteLnFile(2, fileline)

//-----xw.dat の出力-----
one= Universe.op_home
home_x = one.x_i
home_w = - one.C1

one = Universe.op_solar
solar_x = one.x_i
solar_w = one.C1

one = Universe.op_large
large_x = one.x_i
large_w = one.C1

one = Universe.op_trans
trans_x = one.x_i
trans_w_h = one.C1
trans_w_l = - one.C1
fileline = file_hour & "," & home_x & "," & home_w & "," & solar_x & "," & solar_w & "," & large_x & "," & large_w & ","
& trans_x & "," & trans_w_h & "," & trans_w_l
WriteLnFile(3, fileline)

//-----lamda.dat の出力-----
temp_members = Universe.powergrid.area(0).members
temp_member = GetAgt(temp_members, 0)
lamda0 = temp_member.theta

temp_members = Universe.powergrid.area(1).members
temp_member = GetAgt(temp_members, 0)
lamda1 = temp_member.theta

temp_members = Universe.powergrid.area(2).members
temp_member = GetAgt(temp_members, 0)
lamda2 = temp_member.theta

temp_members = Universe.powergrid.area(3).members
temp_member = GetAgt(temp_members, 0)
lamda3 = temp_member.theta

temp_members = Universe.powergrid.area(4).members
temp_member = GetAgt(temp_members, 0)
lamda4 = temp_member.theta

temp_members = Universe.powergrid.area(5).members
temp_member = GetAgt(temp_members, 0)
lamda5 = temp_member.theta

temp_members = Universe.powergrid.area(6).members
temp_member = GetAgt(temp_members, 0)
lamda6 = temp_member.theta

temp_members = Universe.powergrid.area(7).members
temp_member = GetAgt(temp_members, 0)
lamda7 = temp_member.theta

temp_members = Universe.powergrid.area(8).members
temp_member = GetAgt(temp_members, 0)
lamda8 = temp_member.theta

temp_members = Universe.powergrid.area(9).members
temp_member = GetAgt(temp_members, 0)
lamda9 = temp_member.theta

temp_members = Universe.powergrid.area(10).members
temp_member = GetAgt(temp_members, 0)
lamda10 = temp_member.theta

temp_members = Universe.powergrid.area(11).members

```

```

temp_member = GetAgt(temp_members, 0)
lamda11 = temp_member.theta

temp_members = Universe.powergrid.area(12).members
temp_member = GetAgt(temp_members, 0)
lamda12 = temp_member.theta

temp_members = Universe.powergrid.area(13).members
temp_member = GetAgt(temp_members, 0)
lamda13 = temp_member.theta

fileline = file_hour & "," & lamda0 & "," & lamda1 & "," & lamda2 & "," & lamda3 & "," & lamda4 & "," & lamda5 & "," &
lamda6 & "," & lamda7 & "," & lamda8 & "," & lamda9 & "," & lamda10 & "," & lamda11 & "," & lamda12 & "," & lamda13
WriteLnFile(1, fileline)
//-----lamda.dat の出力-----

end if // Universe.File_Output == "する"

//-----
//時間の更新
//-----
Universe.count=Universe.count+1

end if

//-----
//交渉モード処理
//-----
if Universe.nego_mode==true then
  Universe.mode="交渉モード"
  Universe.ct_nego=Universe.ct_nego+1
  if Universe.ct_nego>Universe.nego_step_number then

    Universe.nego_mode=false
    Universe.s=Universe.s+1

    //Universe.DATA_xhat_displayed(0)=0
    //For i=1 to Universe.s
    // Universe.DATA_xhat_displayed(i) = Universe.powergrid.home(Universe.clicked_home).xhat_DATA(i)
    //next i
    //
    //Universe.DATA_x_s_displayed(0)=0
    //For i=1 to Universe.s
    // Universe.DATA_x_s_displayed(i) = Universe.powergrid.home(Universe.clicked_home).x_s_DATA(i)
    //next i

  end if

  //thetaを更新
  For each one in negotiator_agtset
    one.theta=one.theta_next
  next one

  For i = 0 To 5 Step 1
    Universe.output(i)=Universe.powergrid.home(i).theta
  Next i
  For i = 0 To 5 Step 1
    Universe.output(i+5)=Universe.powergrid.solar(i).theta
  Next i

end if // if Universe.nego_mode==true の end

if Universe.G_Graph_display == "する" then
  For i=0 to Universe.count //G グラフの出力
    Universe.DATA_G_displayed(i) = Universe.powergrid.area(Universe.clicked_area).G_DATA(i)
  next i
end if

```

```

//-----theta.dat の出力
if Universe.File_Output == "する" then

    fileline = GetCountStep()
    SortAgtSet(Universe.theta_output_agts, "UID", True)
    For i = 0 to ( Universe.powergrid.area(Universe.File_Output_ID).home_num +
    Universe.powergrid.area(Universe.File_Output_ID).solar_num - 1 )
        one = GetAgt(Universe.theta_output_agts, i)
        fileline = fileline & "," & one.theta
    next i
    WriteLnFile(4, fileline)

end if

Universe.step_total=Universe.step_total+1
}

Univ_Finish{
if Universe.File_Output == "する" then
    CloseFile(1)
    CloseFile(2)
    CloseFile(3)
    CloseFile(4)
end if
}
#end_rule UNIVERSE

//-----
// Agent Rule
//-----
#begin_rule UNIVERSE.POWERGRID.TRANS
Agt_Init{
My.C1=3.0//+@eRnd(0.2)
My.C2= Universe.k
My.C3= Universe.k//+@eRnd(0.2)
My.x_i=0
//My.C4=0.2
//My.D1=1.0
//My.D2=0.5
//My.D3=1.0
//dG=-1
}

Agt_Step{

//-----
//通常モード処理
//-----
if Universe.nego_mode==false then
//U, dU の計算
//My.U = -My.C3*(My.x_i-My.C1)^2+My.C2
//My.dU = -2*My.C3*(My.x_i-My.C1)

if -My.C1<=My.x_i And My.x_i<My.C1 then
    My.U= My.C2
    My.dU= 0
Else
    if My.x_i< -My.C1 then
        My.U=-My.C3*(My.x_i+My.C1)^2+My.C2
        My.dU=-2*My.C3*(My.x_i+My.C1)
    else
        My.U=-My.C3*(My.x_i-My.C1)^2+My.C2
        My.dU=-2*My.C3*(My.x_i-My.C1)
    end if
end if
end if

```



```

My.x_i_next=My.x_i + Universe.alpha * (My.dU - My.from_terminal.lamda+My.to_terminal.lamda )
end if

//-----
//交渉モード処理
//-----
if Universe.nego_mode==true then

end if

}
#end_rule UNIVERSE.POWERGRID.TRANS

//-----
// Agent Rule
//-----
#begin_rule UNIVERSE.POWERGRID.TRANS_TERMINAL
Agt_Init{
My.lamda=0
My.x_i=0
My.x_s=0
My.gamma=Universe.gamma
My.u_x=0
My.u_l=0
My.v_l=0
My.v_xl=0
My.a_i=0
My.b_i=0
My.lamda=0
My.x_i=0
My.x_s=0
My.x_est_0=0
}

Agt_Step{
Dim one as Agt
Dim temp_agt1 as Agt
Dim temp_agt2 as Agt
Dim sigma as Double
Dim init as Double
Dim i as Integer
Dim j as Integer

If My.type=="from" then
My.x_i = - My.trans.x_i
else
My.x_i = My.trans.x_i
end if

//-----
//交渉モード処理
//-----
if Universe.nego_mode==true then

if Universe.ct_nego==1 then

init=My.x_i

My.theta=My.lamda - Universe.beta_d*init
My.theta_next=My.theta
My.x_est=init

else

sigma=0
for each one in My.link

```

```

        sigma=sigma+My.theta-one.theta
    next one
    My.theta_next=My.theta-Universe.gamma*sigma

    if Universe.ct_nego==Universe.nego_step_number then
        My.lamda=My.theta_next
        My.x_s=My.x_i
    end if

end if

end if
}
#end_rule UNIVERSE.POWERGRID.TRANS_TERMINAL

-----
// Agent Rule
-----
#begin_rule UNIVERSE.POWERGRID.HOME
Agt_Init{
My.C1=1.0//+@eRnd(0.2)
My.C2=1.0

My.C3=Universe.k / 5//+@eRnd(0.2)
My.C4=Universe.k / 5

My.C3_1=Universe.k / 10.0 //Large の
My.C4_1=Universe.k / 10.0

My.D4 = 0.2

My.D3 = Universe.E * (1 + @eRnd(0.5)) //r
My.D1 = My.D3 * My.D4 //p
My.D2 = My.D3 * My.D4 * 0.5 //q

//My.dG=-1
My.gamma=Universe.gamma
My.u_x=0
My.u_1=0
My.v_1=0
My.v_xl=0
My.a_i=0
My.b_i=0
My.lamda=0
My.x_s=0
My.x_est_0=0
My.xhat = 0
My.large = 0

My.x_i = - My.D3 //*(1 + @eRnd(0.5)) //初期誤差
}

Agt_Step{
Dim one as Agt
Dim temp_agt1 as Agt
Dim temp_agt2 as Agt
Dim sigma as Double
Dim init as Double
Dim i as Integer
Dim j as Integer

My.sv_lamda=My.area.lamda

-----
//通常モード処理
-----
if Universe.nego_mode==false then

```

```

if My.large == 0 then
//効用の時間変化を計算
if Universe.time_hour < 6 then
My.C1= My.D3 - My.D2 * Sin(PI() * Universe.time_hour_conti / 6)
Else
My.C1= My.D3 + My.D1 * Sin(PI() * (Universe.time_hour_conti -6) / 18)
end if

//U, dU の計算
if My.x_i > -My.C1 then
My.U = -My.C3*(My.x_i+My.C1)^2+My.C2
My.dU = -2*My.C3*(My.x_i+My.C1)
Else
My.U = -(My.C4)*(My.x_i+My.C1)^2+My.C2
My.dU = -2*(My.C4)*(My.x_i+My.C1)
end if
My.x_i_next=My.x_i + Universe.alpha * (My.dU + My.lamda )

ElseIf My.large == 1 then
if Universe.count == 0 then
if My.area.ID == 2 then
My.area.ave_D3 = My.area.ave_D3 * 0.7
End if
My.x_i = (My.area.home_num-1) * (My.area.ave_D3 )/- My.D2 * 0.5 * Sin(PI() * 1.0 / 6)  /** 1.1 //+ @eRnd(0.7)
end if

//効用の時間変化を計算
if Universe.time_hour < 6 then
My.C1= (My.area.home_num-1) * (My.area.ave_D3 - My.D2 * 0.5 * Sin(PI() * Universe.time_hour_conti / 6))
Else
My.C1= (My.area.home_num-1) * (My.area.ave_D3 + My.D1 * 0.5 * Sin(PI() * (Universe.time_hour_conti - 6) / 18))
end if

//U, dU の計算
if My.x_i < My.C1 then
My.U = -My.C3_1*( My.x_i-My.C1)^2+My.C2
My.dU = -2*My.C3_1*(My.x_i - My.C1)
Else
My.U = -My.C4_1*(My.x_i-My.C1)^2+My.C2
My.dU = -2*My.C4_1*(My.x_i-My.C1)
end if
My.x_i_next=My.x_i + Universe.alpha * 2 * (My.dU + My.lamda )
end if //if My.large の end if

//状態量の更新

end if //if Universe.nego_mode==false の end if

//-----
//交渉モード処理
//-----
if Universe.nego_mode==true then

if Universe.ct_nego==1 then

init=My.x_i

My.x_s_DATA(Universe.s)=My.x_i

My.theta=My.lamda - Universe.beta_d*init
My.theta_next = My.theta

else
sigma=0
for each one in My.link
sigma=sigma+My.theta-one.theta
next one
My.theta_next=My.theta-Universe.gamma*sigma

if Universe.ct_nego==Universe.nego_step_number then
My.lamda=My.theta_next

```

```

    My.x_s=My.x_i
    end if

end if

end if

}
#end_rule UNIVERSE.POWERGRID.HOME

//-----
// Agent Rule
//-----
#begin_rule UNIVERSE.POWERGRID.SOLAR
Agt_Init{
My.C1=0 //+@eRnd(0.2)
My.C2=1.0

My.C3=Universe.k / 10//+@eRnd(0.1)
My.C4=Universe.k

My.D1=Universe.E * 0.05 * (1 + @eRnd(0.3)) //太陽光発電の p

My.D2=1.0 //風力発電の p
My.D3=1.0 //風力発電の q

My.dG=1
My.gamma=Universe.gamma
My.u_x=0
My.u_l=0
My.v_l=0
My.v_xl=0
My.a_i=0
My.b_i=0
My.lamda=0

My.x_s=0
My.x_est_0=0

My.x_i= 0 //+ @eRnd(0.5)
}

Agt_Step{
Dim one as Agt
Dim temp_agt1 as Agt
Dim temp_agt2 as Agt
Dim sigma as Double
Dim init as Double
Dim i as Integer
Dim j as Integer

//-----
//通常モード処理
//-----
if Universe.nego_mode==false then

if 6<= Universe.time_hour And Universe.time_hour < 18 then
My.C1= My.D1 * Sin(PI() * ( Universe.time_hour_conti - 6) / 12)
else
My.C1= 0
end if

//U, dU の計算
if My.x_i<My.C1 then
My.U = -My.C3*(My.x_i-My.C1)^2+My.C2
My.dU = -2*My.C3*(My.x_i-My.C1)
Else

```

```

My.U = -My.C4*(My.x_i-My.C1)^2+My.C2
My.dU = -2*My.C4*(My.x_i-My.C1)
end if

My.x_i_next =My.x_i + Universe.alpha * (My.dU + My.lamda)
end if

//-----
//交渉モード処理
//-----
if Universe.nego_mode==true then

if Universe.ct_nego==1 then

init=My.x_i

My.theta=My.lamda - Universe.beta_d*init
My.theta_next=My.theta

else

sigma=0
for each one in My.link
sigma=sigma+My.theta-one.theta
next one
My.theta_next=My.theta-Universe.gamma*sigma

if Universe.ct_nego==Universe.nego_step_number then
My.lamda=My.theta_next
My.x_s=My.x_i
end if

end if

end if

}
#end_rule UNIVERSE.POWERGRID.SOLAR

//-----
// Agent Rule
//-----
#begin_rule UNIVERSE.POWERGRID.AREA
Agt_Init{
My.lamda=0
My.lamda_next=0
My.G_i=0
}

Agt_Step{
Dim one as Agt
Dim temp_agtset as Agtset

//-----
//通常モード処理
//-----
if Universe.nego_mode==false then
My.G_i = 0
//ClearAgtSet(temp_agtset)
//DuplicateAgtSet(temp_agtset ,My.members)
//DelAgtSet (temp_agtset, My.trans_terminals)
For each one in My.members
My.G_i = My.G_i + one.x_i
next one
For each one in My.trans_terminals
My.G_i = My.G_i + one.x_i
next one

/*

```

```

For each one in My.trans_out_set
  My.G_i = My.G_i - one.x_i
next one
For each one in My.trans_in_set
  My.G_i = My.G_i + one.x_i
next one
*/

if Universe.time_min == 0 And Universe.time_second == 0 And Universe.time_hour != 0 then
  My.lamda_next = My.lamda - Universe.beta * My.G_i
end if

My.p=My.lamda
My.G_DATA(Universe.count)=My.G_i
end if

if Universe.nego_mode==true then

if Universe.ct_nego==1 then
  My.init_ave = 0
  //ClearAgtSet(temp_agtset)
  //DuplicateAgtSet(temp_agtset ,My.members)

  For each one in My.members
    My.init_ave = My.init_ave+one.lamda-Universe.beta_d*one.x_i
  next one
  For each one in My.trans_terminals
    My.init_ave = My.init_ave+one.lamda-Universe.beta_d*one.x_i
  next one

  My.init_ave=My.init_ave/(CountAgtSet(My.members)+CountAgtSet(My.trans_terminals))
end if

end if

}
#end_rule UNIVERSE.POWERGRID.AREA

```