

# モデリング言語による社会シミュレーション構築環境

青山 希<sup>†1</sup> 松澤 芳昭<sup>†2</sup>  
井庭 崇<sup>†3</sup> 大岩 元<sup>†4</sup>

本論文では、社会シミュレーションのプログラムを、モデリング言語だけで作成するための記法と、モデル化のプロセス、そしてそれらを支援するツールを提案する。提案する環境では、モデルの静的な構造と、動的な振る舞いを UML で記述し、その中で行われる具体的な処理を、独自に定義したビジュアルなアクション記述言語である「Action Block Language」(ABL) で記述する。このシミュレーション構築環境を用いると、モデリング言語でモデルを記述するだけで、あとは支援ツールがそのソースコードを自動生成し、シミュレーションの実行が可能となる。その結果、モデル作成者はソースコードを記述する必要がなくなり、社会現象をモデル化する作業に集中できるようになる。さらに、提案する環境の有効性を明らかにするため、既存モデルの再現実験を行った。

## Building Social Simulation with Modeling Language

NOZOMU AOYAMA,<sup>†1</sup> YOSHIAKI MATSUZAWA,<sup>†2</sup> TAKASHI IBA<sup>†3</sup>  
and HAJIME OHIWA <sup>†4</sup>

In this paper, we propose a visual modeling language, a modeling process, and tools for building social simulation with modeling language. A modeler can develop the programs of social simulation with “Unified Modeling Language” (UML) and “Action Block Language” (ABL), which is proposed as visual modeling language for describing the agent’s actions. Therefore, we do not have to take care of the implementation, because the tools will translate from models, which is described with modeling language, to the program codes. In this paper, we experimented the performance of the proposed language and tools, applying them to an existing model “Format Competition of Video Cassette Recorder”. As a result, the performance of the proposed language and tools is justified in the application.

### 1. はじめに

マルチエージェント・モデルによるコンピュータ・シミュレーションは、複雑な社会現象を理解するための道具として、さまざまな分野で利用されている<sup>1)</sup>。しかし、シミュレーションをプログラミング言語で実装するためには、文法や技術的な側面など、モデルに直接関係のない事柄を意識する必要があり、モデリングとは別のスキルが必要になる<sup>2)</sup>。そのため、プログラミングに馴染みのない社会学者は、分析手法としてシミュレーションを用いることが非常に困難である

という問題がある。また、モデリングと実装の、抽象度の異なる二つの作業を行き来しなければならなくなるため、モデリングの段階まで立ち戻ってモデルを推敲することが難しいという問題もある。

このような背景を踏まえ、本論文では、社会シミュレーションのプログラムを、モデリング言語だけで作成するための環境を提案する。これには、社会をモデル化するために利用する記法の定義と、それを利用してモデルを記述するためのプロセス、そしてその支援ツールが含まれている。これらを用いることで、モデル作成者は、社会現象をモデリング言語で記述し、記述したモデルから実行可能なソースコードを自動生成することができる。その結果、プログラミング言語による実装を意識せず、より抽象度の高い記述だけでシミュレーションのプログラムを作成することができるようになる(図1)。

以下では、まずモデリング言語でシミュレーションを構築するための、記法とプロセスについて述べる。その後、提案環境を用いて、マルチエージェントによ

<sup>†1</sup> 慶應義塾大学 SFC 研究所

Keio Research Institute at SFC

<sup>†2</sup> 慶應義塾大学 政策・メディア研究科

Graduate School of Media and Governance, Keio University

<sup>†3</sup> 慶應義塾大学 総合政策学部

Faculty of Policy Management, Keio University

<sup>†4</sup> 慶應義塾大学 環境情報学部

Faculty of Environmental Information, Keio University

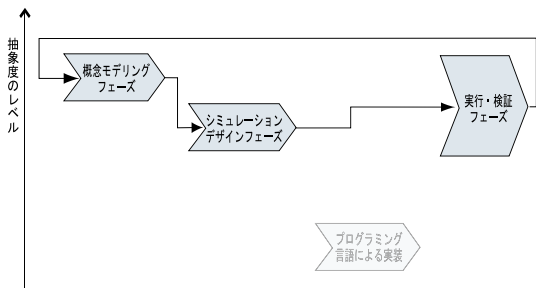


図 1 本論文で提案するシミュレーション構築のプロセスと抽象度  
Fig. 1 Proposed building process and the level of abstraction

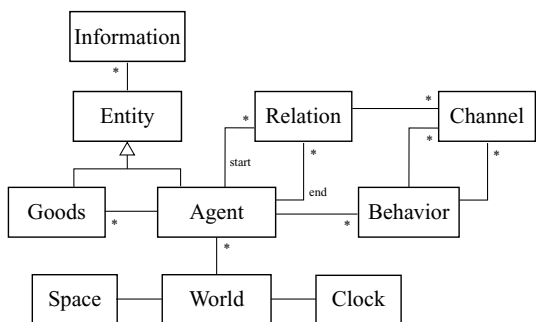


図 2 PlatBox 基礎モデルのクラス図 (先行研究<sup>3)</sup> より)  
Fig. 2 Class diagram of PlatBox Foundation Model

る人工市場モデルを再現することで、その有用性を評価する。

## 2. 社会シミュレーション構築のためのモデリング言語

本論文が提案する手法では、抽象度の異なるモデルを作成するために、2種類のモデリング言語を使用する。本節ではまず、モデルを作成するための語彙となる「PlatBox 基礎モデル」について述べ、次にモデル記述のための記法について述べる。

### 2.1 モデル作成のための語彙

本論文で提案する手法は、モデルを記述するための語彙として、「PlatBox 基礎モデル」(PlatBox Foundation Model: 以下、基礎モデル)を使用する。基礎モデルは、エージェントベースアプローチによって社会・経済をモデル化する際によく現れる要素や構造を、モデル化のための枠組みとして抽出したものである(図2)。

### 2.2 概念モデルを作成するためのモデリング言語

本論文で提案する手法では、モデルの静的な構造と動的な振る舞いを概念モデルとして記述する。これらはUML(統一モデリング言語)<sup>4)</sup>に定義された記法を用いて表現することができる。

### 2.3 実行可能なモデルを作成するためのモデリング言語

本論文で提案する手法では、モデリング言語で記述した図からプログラミング言語のソースコードを生成する。そのためには、概念モデルをどのように実現するのかを詳細に定義した、一段抽象度の低いモデルを記述する必要がある。このモデルのことをここでは、「実行可能なモデル」という。

実行可能なモデルの記述においては、静的な構造とBehaviorの状態遷移を、UMLのクラス図とステートチャート図で表現する。Behaviorが状態遷移するとき起きるアクションの記述は、独自に定義した社会シミュレーションのためのビジュアルなアクション記述言語「Action Block Language」(ABL)で表現する。ABLについての詳細は次節で述べる。

### 2.4 社会シミュレーションのためのアクション記述言語「Action Block Language」(ABL)

アクション記述言語「Action Block Language」(ABL)は、社会シミュレーションの中で実行される処理(アクション)を、基礎モデルの語彙で記述するためのビジュアルな記述言語である。

#### 2.4.1 アクションパーツ

ABLには、アクション記述に頻出する処理のまとまりを定義した「アクションパーツ」が用意されており、それらには基礎モデルの語彙で記述された名前がつけられている。モデル作成者は、「このAgentが持つGoodsを全て取り出す」というような処理(アクション)の目的を考え、対応するアクションパーツを選択してアクションを記述することができる。

ABLには、基礎モデルに定義されているすべての操作に対応する100個以上のアクションパーツが用意されている。これらのアクションパーツと制御構造を組み合わせることで、基礎モデルに基づいて作成されたモデルのアクションは、すべてABLで記述することができる。

#### 2.4.2 ABLによるアクションの記述

ABLを用いてアクションを記述するには、まずアクションとして定義された処理を、基礎モデルの語彙で記述できる程度まで詳細化する必要がある。ここでは、商取引でよく登場する「お金を支払う」というアクションの例を考えてみることにしよう。最初に、

PlatBox 基礎モデルは、これまで「Boxed Economy 基礎モデル」という名称で提案してきたが<sup>3)</sup>、Boxed Economy Project の名称変更に伴い現在の名称に変更された。

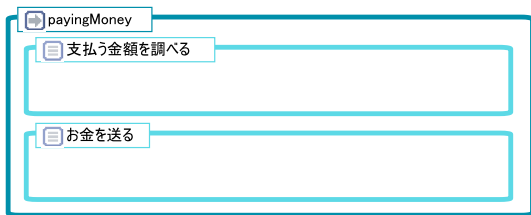


図 3 (1) アクションの処理を ABL で詳細化した例  
Fig. 3 An Example of The Detailed Action

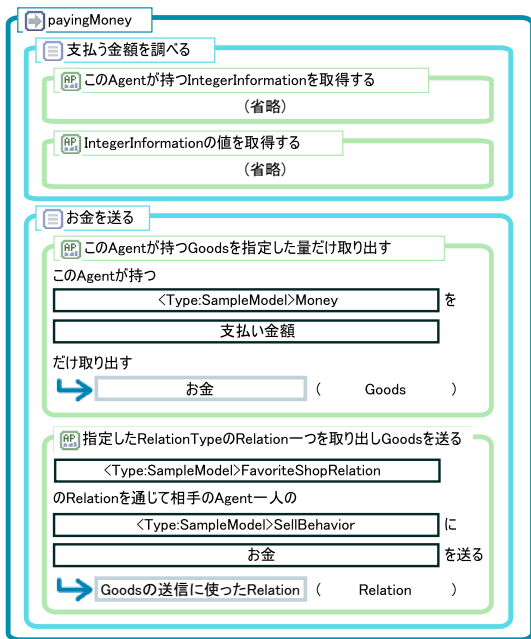


図 4 (2) アクションパーツの引数を指定したアクションの例  
Fig. 4 An Example of The Action selected The Parameter

「payingMoney」というアクションを詳細化して、「支払う金額を調べる」と「お金を送る」という2つの処理のまとまりに分けて記述する(図3)。この詳細化の作業の中で、必要ならば条件分岐や繰り返しなどの制御構造も使用することができる。

次に、詳細化したアクションの中の処理を、アクションパーツを組み合わせて記述する作業を行う。例では、「お金を支払う」という処理を、「このAgentが持つGoodsを指定した量だけ取り出す」と「指定したRelationTypeのRelation一つを取り出してGoodsを送る」という2つのアクションパーツを組み合わせて記述する(図4)。

### 3. 社会シミュレーション構築のプロセス

ここでは、これまで提案してきたモデリング言語を用いてモデル作成を行うための包括的なプロセスを提

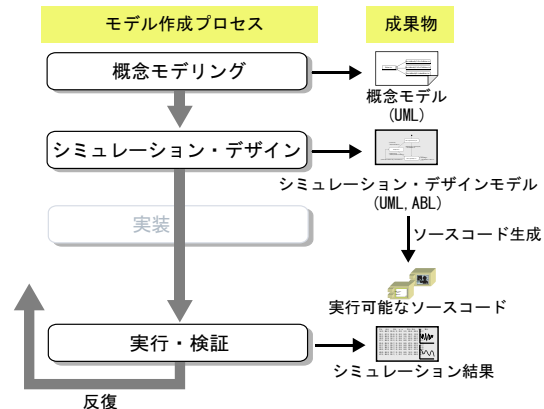


図 5 モデル作成プロセスの全体像  
Fig. 5 Development Process Overview

案する<sup>5)</sup>。

提案するプロセスには「概念モデリングフェーズ」、「シミュレーションデザインフェーズ」、「実行・検証フェーズ」の3つのフェーズが定義されている。さらに、それぞれのフェーズには成果物として必要なモデルが複数定義されている。モデル作成者はこのプロセスに従って、必要なモデルを作成していくことで、実行可能な社会シミュレーションのプログラムを得ることができる(図5)。以下では、それぞれのフェーズについて詳しく説明する。

#### 3.1 概念モデリングフェーズ

概念モデリングフェーズは、社会現象を分析し、概念モデルとして記述するフェーズである。モデル作成者は、モデル化しようとしている対象が「どのようなものであるか」を洗い出し、Agent, Relation, Informationなどの基礎モデルの要素の組み合わせとして記述する。

このフェーズでは、モデル作成者は「概念の静的な構造の記述」、「Agentのアクティビティの記述」、「Agent間の相互作用の記述」の3種類の概念モデルを、反復しながら作成する。

#### 3.2 シミュレーション・デザインフェーズ

シミュレーション・デザインフェーズは、概念モデリングフェーズで作成した概念モデルを、実行可能なソフトウェアコンポーネントとして、「どのように実現するか」を記述するフェーズである。このフェーズでは、モデルから実行可能なソースコードを生成するために、概念モデルの実現方法を詳細に定義する必要がある。

このフェーズでは、実行可能なソースコードを生成するために、以下の3種類のモデルとシミュレーションの初期設定を記述する。モデル作成者は、「シミュ

レーションの静的な構造の設計」,「Behaviorの状態遷移の設計」,「Behaviorが状態遷移する際のアクションの設計」の3種類のモデル作成を反復して行い,最後に「シミュレーションの初期設定」を行う。

### 3.3 実行・検証フェーズ

実行・検証フェーズは,モデルから生成したシミュレーションのソースコードを,PlatBox Simulator<sup>6)</sup>上で実行しその正当性を検証するフェーズである。PlatBox Simulatorは,基礎モデルに基づいて記述されたプログラムを実行するためのプラットフォームであり,グラフやファイル出力などの方法でシミュレーションの結果を得ることができる。

## 4. 提案する手法の評価

### 4.1 提案する手法による既存モデルの再現

提案する手法の有用性を評価するため,既存モデルの再現を行った。テーマとして取り上げたのは「家庭用VCR(Video Cassette Recorder)の規格競争シミュレーション」<sup>7)</sup>のモデルである。このモデルは,規格競争の典型といわれる家庭用VCRの市場を,消費者の相互作用としてモデル化し,その振舞いを分析するためのマルチエージェントモデルである。

モデルを再現した結果,本論文で提案するモデル作成環境を利用することで,ソースコードを記述せずにモデリング言語でシミュレーションを構築することができた。具体的にはシミュレーションを構成する16個のJavaファイル,3176行のソースコードを,モデル図から自動で生成することができた。モデル図の内訳は,クラス図8枚,ステートチャート図7枚,ABLの図7枚である。

### 4.2 他の適用事例

前節で示した適用事例以外にも,本論文で提案するシミュレーション構築環境は,以下のような事例への適用が行われている。

- 繰り返し囚人のジレンマモデル
- 進化経済学的な産業変動モデル
- 外国為替市場モデル
- バスケット通貨制度モデル

これら以外にも大学における研究,教育の場で利用されてきており,提案するシミュレーション構築環境を利用することで,基礎モデルで表現可能なモデルならば,ソースコードを記述せずにシミュレーションが行えることが明らかになってきている。

## 5. おわりに

本論文では,社会シミュレーションをモデリング言

語で構築するための,記法,プロセス,支援ツールの提案を行い,その評価として提案する環境を使用した既存モデルの再現を行った。その結果,プロセスにしたがってモデルを作成し,モデルから実行可能なプログラムを生成することで,モデリング言語だけで家庭用VCRの規格競争シミュレーションを構築することができた。これらの成果が,健全な社会研究コミュニティの発展に寄与することを期待する。

謝辞 慶應義塾大学 SFC の PlatBox Project のメンバーおよび,その前身である Boxed Economy Project のメンバーに感謝したい。特に支援ツールのプロトタイプ開発に携わった浅加浩太郎さん,仕様策定に協力してくれた津屋隆之介さん,開発に携わった武田林太郎さんに感謝したい。また,支援ツールの試用に協力してくれた慶應義塾大学井庭研究室のメンバーにも感謝の意を述べたい。

## 参考文献

- 1) 山影進, 服部正太: コンピュータの中の人工社会: マルチエージェントシミュレーションモデルと複雑系, 共立出版 (2002).
- 2) Stefasson, B.: Simulating Economic Agents in Swarm, *Economic Simulations in Swarm: Agent-Based Modelling and Object Oriented Programming* (Luna, F. and Stefasson, B.(eds.)), Kluwer Academic Publishers (2000).
- 3) 井庭崇, 中鉢欣秀, 松澤芳昭, 海保研, 武藤佳恭: Boxed Economy Foundation Model: 社会・経済のエージェントベースモデリングのためのフレームワーク, 情報処理学会論文誌: 数理モデル化と応用, Vol. 44, No. SIG14 (TOM9), pp. 20-30 (2003).
- 4) Object Management Group: *OMG Unified Modeling Language Specification*, Object Management Group (2000). 『UML仕様書』, OMG Japan SIG 翻訳委員会 UML 作業部会 (訳), アスキー, 2001.
- 5) Iba, T., Matsuzawa, Y. and Aoyama, N.: From Conceptual Models to Simulation Models: Model Driven Development of Agent-Based Simulations., *9th Workshop on Economics and Heterogeneous Interacting Agents*. (2004).
- 6) Iba, T.: A Framework and Tools for Modeling and Simulating Societies as Evolutionary Complex Systems., *2nd. International Conference of the European Social Simulation Association*. (2004).
- 7) 井庭崇, 竹中平蔵, 武藤佳恭: 人工市場アプローチによる家庭用VTRの規格競争シミュレーション, 情報処理学会論文誌: 数理モデル化と応用, Vol.42, No. SIG14 (TOM5), pp. 73-89 (2001).