

メカニズム指向ABM のためのプログラムの設計と実装

「メカニズム指向ABM シミュレーション研究 IV」

○岩倉洋平(香川短期大学)・荻林成章(千葉工業大学)・名倉賢(大和大学)・佐藤 浩(防衛大学校)

キーワード: メカニズム指向ABM, オブジェクト指向プログラミング, ボトムアップシミュレーション

1 はじめに

メカニズム指向ABM(荻林2024)の実現には, その方法論的思想を的確に反映したコンピュータプログラムの実装が不可欠となる。プログラムは, 単に計算を実行するツールではなく, 複雑な社会システムの構造とダイナミクスを表現し, メカニズム探求のための「実験装置」として機能しなければならない。

本稿では, メカニズム指向ABM の実践に求められるプログラム上の要件を明らかにし, その要件を満たすための具体的なプログラミングパラダイムと実装コンポーネントについて, Ogibayashi (2022)で用いられている実装例を用いて解説する。

2 メカニズム指向ABM のプログラム要件

メカニズム指向ABM は, その目的と方法論から, プログラム実装に対して以下のような特有の要件を課す。

(1) ボトムアップ原理の徹底

プログラムはマクロレベルでの変数間の関係性を事前に仮定せず, 個々のエージェントの振る舞いを記述し, その集積結果としてマクロな指標(例: GDP, 平均価格, 感染者数)が計算されるように設計されねばならない。

(2) エージェントの多様性と自律性の表現

プログラムは, これらのエージェントの異質性(Heterogeneity)や, 状況に応じて自律的に判断・行動する能力を柔軟に表現できる必要がある。

(3) 複雑な相互作用のモデル化

プログラムは, 様々な形態の相互作用を明記し, 実装できる必要がある。

(4) モデル構造の体系的探索の容易性

プログラムは, これらのモデル構成要素を容易に変更, 追加, 削除できるようなモジュール性と拡張性を持つ必要がある。

(5) 計算効率とスケーラビリティ

パラメータ探索や感度分析のために多数回のシミュレーションを実行する場合, 十分な計算効率と, モデル規模の拡大に対応できるスケーラビリティが求められる。

3 オブジェクト指向プログラミングの適用

上記のプログラム要件を満たす上で, オブジェクト指向プログラミング(object-oriented programming, OOP)は非常に有効なパラダイムである。OOP がメカニズム指向ABM に適している理由は以下の通りである。

(1) エージェントとオブジェクトの親和性

固有の属性を持ち自律的に行動するABM の「エージェント」の概念と自然に対応づけることができる。

(2) 多様性(ポリモーフィズム)の表現

ポリモーフィズムを利用することで, 共通のI/F(例:「行動する」「相互作用する」)に対し, 固有動作が実行可能となり, 柔軟性・拡張性が高まる。

(3) モジュール性と再利用性

クラス単位でプログラムが構成されるため, モデルの各構成要素(エージェントの種類, 相互作用ルール, 環境など)を独立したモジュールとして開発・管理しやすい。

4 実装における主要コンポーネント

メカニズム指向ABM をOOP で実装する際, 典型的には以下のような主要コンポーネント(クラス)が必要となる(Ogibayashi 2022)。社会現象におけるシミュレーションでは, これらの基本コンポーネントに加え, 特定の目的に応じたクラスが追加される。表1に主なクラスと役割, 図1に基本フローを記載する。

(1) エージェントクラス

モデルの中心。各クラスが対象とするシステムに応じて様々な種類が定義される(消費者, 生産者, 銀行, 感染者など)。属性(所持金, 在庫量, 感染状態など)と行動ルール(購入判断, 生産決定, 感染プロセスなど)をメソッドとして含む。

(2) シミュレーション管理クラス

時間ステップの進行, エージェントの行動順序の制御, シミュレーション全体の開始・進行・終了の管理と, 各種オブジェクトへのアクセスを管理する。

(3) パラメータ管理クラス

シミュレーションの初期条件や各種パラメータ(税率)エージェント数などを外部ファイル等から読み込み管理する。

(4) 乱数管理クラス

エージェントの初期属性のばらつきや, 確率的な意思決定, 相互作用のタイミングなどを制御するために, 質の高い乱数を生成・管理する。

(5) 参照管理クラス

各種オブジェクトのアドレスを返す関数を保有する(相互参照回避)。

(6) データ集計・出力クラス

シミュレーション中に生成されるミクロ・マクロのデータを収集し, 分析可能な形式でファイル等へ出力する。

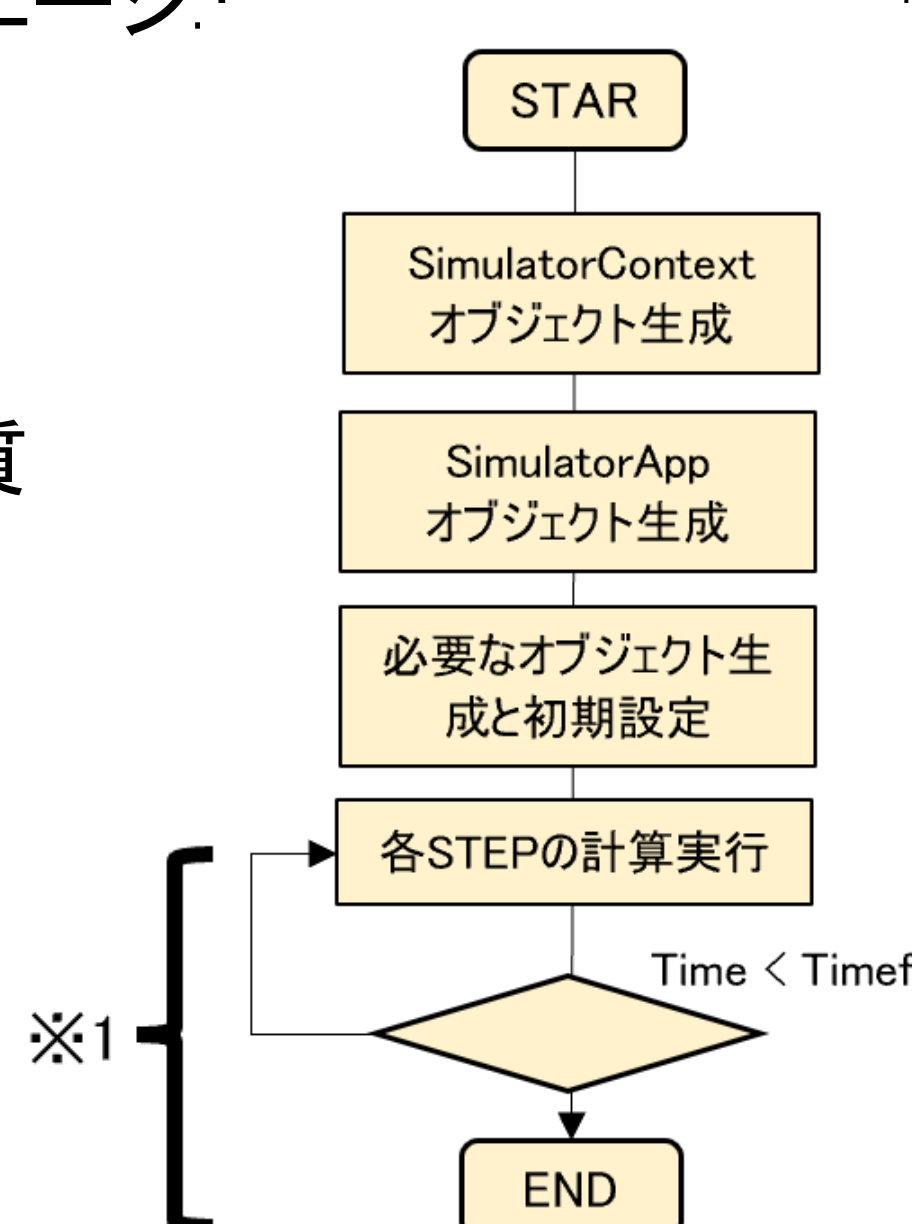


図1. 計算の基本フロー

5 経済システムへの応用

メカニズム指向ABMを経済現象の分析において, 特に資金や財のフローを正確に追跡することが不可欠である。

Ogibayashi (2022)では, この要請に応えるため, 現実世界の会計原則である「複式簿記」をABM フレームワーク内に実装している。具体的には, 以下のクラスが導入されている。図2に継承関係の例, 表2に財務に関わるクラス, 図3に経済システムの概念モデルを示す。経済システムへの応用の場合は, 図1(※1)にて「期首・期中・期末」処理を実施する。

(1) 会計計算クラス

各エージェント(消費者, 生産者, 銀行, 政府など)に属し, そのエージェントの資産, 負債, 資本, 収益, 費用といった財務状態を管理する。

(2) 会計データ集計クラス

各エージェントの会計データを集計し, GDPなどのマクロ経済指標を算出する。

(3) 仕訳伝票作成

エージェント間で取引(例: 商品の売買, 賃金の支払い, 借入)が発生するたびに, 会計原則に基づいた「仕訳伝票」が内部的に作成され, 関係するエージェントの会計計算クラスの各勘定科目の残高が更新される。

表2. 資金や財務に関わる主なクラスと役割

No	主要クラス	概要
(1)	AccountData	財務データや市場データを管理し, ファイルに出力する。
(2)	AccountantData	AccountantDataは, AccountData を利用して, より高レベルな財務処理やデータ管理を行う。
(3)	BookKeeper	各エージェントの財務データを管理する。取引発生時の仕訳帳更新, 期末に試算表を作成等, 財務状況を把握する。

※ Noは上記, 番号に対応する。

表1. 各種モデル共通の主なクラスと役割

No	主要クラス	概要
(2)・1	SimulatorApp	計算の全行程の管理。各オブジェクト生成と初期設定, 期首・期中・期末のような制御処理を各期繰り返す。
(2)・2	SimulatorContext	SimulatorConfig, Navigator等をどのクラスからも呼び出せるよう, 各オブジェクトへのアクセス管理を行う。
(3)	SimulatorConfig	設定管理。計算に必要なパラメータ変数, 入出力ファイルのパス等。
(4)	RandomGenerator	C++ 組み込み関数以外の各種乱数を生成する関数を保有。
(5)	各種Navigator	各種オブジェクトアドレスへ返す関数を保有(相互参照回避)。

※ Noは左記「主要コンポーネント」に対応する。

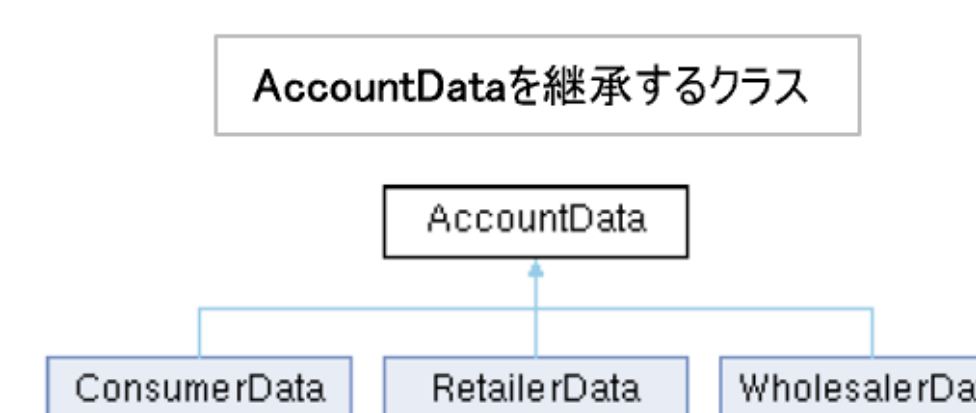
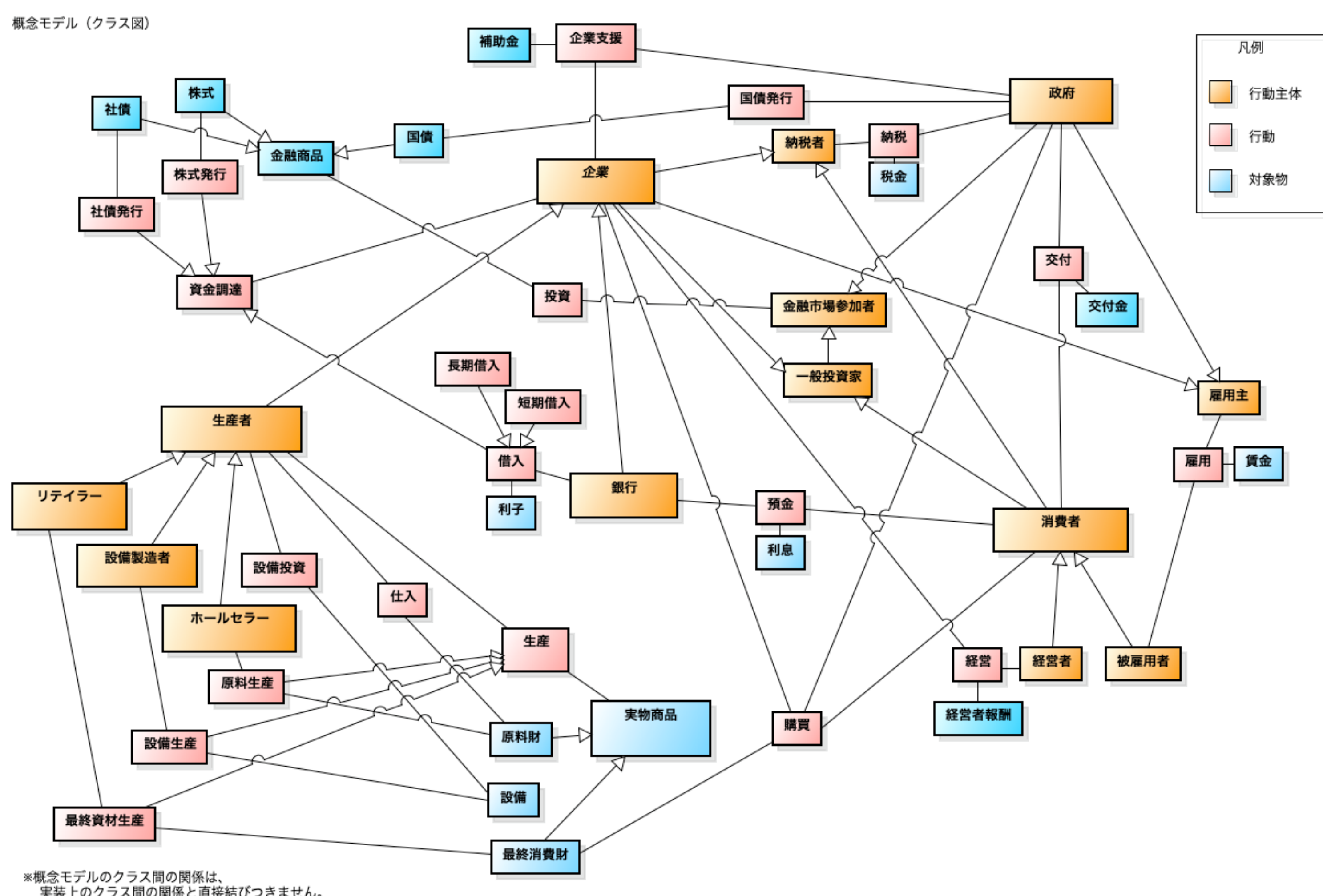


図2. 経済基本モデルにおける継承関係の例



※概念モデルのクラス間の関係は、実装上のクラス間の関係と直接結びつきません。

図3 経済システムの概念モデル(クラス図)

6 実装言語と開発環境

Ogibayashi (2022)では, 開発環境としてMicrosoft VisualStudio, 実装言語としてC++が用いられている。C++は, OOP をサポートし, 大規模なシミュレーションに必要な計算性能とメモリ管理の柔軟性を提供する点で適している。Visual Studio は, デバッグ機能などが充実した統合開発環境(IDE)である。

ただし, メカニズム指向ABM の原理自体は, 前述したメカニズム指向ABM のプログラム要件を効果的に満たせることが重要ある。

7 おわりに

本稿では, ボトムアップ原理, エージェントの多様性, モデル構造の探索容易性といった要件を満たす上で, オブジェクト指向プログラミングがいかに有効であるかを論じた。今後, メカニズム指向ABMのための再利用可能なライブラリやフレームワークが整備され, 様々な分野の研究者がより容易にこのアプローチを実践できるようになることが期待される。さらに, 近年発達の著しい生成AIによるプログラミング支援を用いる事で, メカニズム指向ABMの研究は, 社会科学における科学的知識の蓄積を加速させ, 複雑な社会問題へのより深い洞察をもたらすであろう。

参考文献

- 荻林成章 (2024)『メカニズム指向 ABM シミュレーション研究部会』, 日本シミュレーション & ゲーミング学会 2024 年秋季大会論文報告集
- Ogibayashi, S. (2022) Model structure of Agent-Based Artificial Economic System Responsible for Reproducing Fundamental Behavior of Goods Market. In Yuji Aruka (ed), Digital Designs for Money, Markets, and Social Dilemmas, pp. 89-141, Springer.

著者紹介 岩倉洋平 iwakura@kjc.ac.jp