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

井庭 崇[†1,†2] 中鉢欣秀[†3] 松澤芳昭[†3]
海保研[†3] 武藤佳恭[†4]

本論文では，エージェントベースアプローチによって，社会・経済のモデルを作成するためのフレームワークを提案する．このフレームワークによって，「複雑系」（構成要素の振舞いのルールが状況によって動的に変化するシステム）のモデルを記述し，シミュレートすることが可能となる．提案フレームワークは，概念モデル・フレームワークとシミュレーションモデル・フレームワークの 2 つで構成され，モデル化からシミュレーションまでの一貫した支援を行う．提案フレームワークの特徴は，エージェント間の相互作用を，財（および情報）のやりとりとして明示化する点にある．また，エージェントの行動を，エージェントとは別のモデル要素として外部化し，オブジェクトコンポジションによって付加する点にも特徴がある．本論文は，社会・経済システムをオブジェクト指向でモデル化するための 1 つのパターンを提示するだけでなく，モデルを作成・実行する環境もあわせて提供することで，シミュレーションを行う仕組みも実現する．

Boxed Economy Foundation Model: Framework for Agent-based Modeling of Economies and Societies

Takashi Iba,[†1,†2] Yoshihide Chubachi,[†3] Yoshiaki Matsuzawa,[†3]
Ken Kaiho[†3] and Yoshiyasu Takefuji[†4]

In this paper, we propose a framework of agent-based models for economies and societies. This framework allows us to describe and simulate the complex system where the rules of the behavior for each elements change dynamically through out the simulation. The proposed framework is based on a set of two models. One is for building the conceptual models and the other for executing the simulation. Two models together are able to process consistently from modeling to execution. One of characteristics in the proposed framework is that interactions between agents are clearly declared as the exchange of goods (with information). Also, a behavior is defined as a different object from each agents in the model to realize flexible design by using the object-composition technique. This paper not only provides a pattern of modeling the socio-economic system by using object-oriented methodology, but also is able to analyze the state of the system using simulation by providing the environment to build the model and execute the simulation.

1. はじめに

本論文では，エージェントベースアプローチによって，社会・経済のモデルを作成するためのフレームワークを提案する．エージェントベースアプローチとは，社会・経済を多数の自律的主体（エージェント）のミクロ的な相互作用からなるシステムとしてモデル化するものである．エージェントベースアプローチでは，単にミクロレベルの相互作用を組み込むというのではなく，ミクロレベルとマクロレベルが不可分であるという点が強調されることが多く，このようなシステムは「複雑系」と呼ばれている[1]．「複雑系」という用語のここでの定義は，「構成要素の振舞いのルールが状況によって動的に変化するシステム」というものである[2]．

このような複雑系のモデルを厳密に記述するためには，振舞いのルールを固定化している従来の力学系を超えて，「相空間の次元，ルールなどの点で『開いた

†1 千葉商科大学政策情報学部
Faculty of Policy Informatics, Chiba University of Commerce
†2 フジタ未来経営研究所
Fujita Institute of Future Management Research
†3 慶應義塾大学 SFC 研究所
Keio Research Institute at SFC, Keio University
†4 慶應義塾大学環境情報学部
Faculty of Environmental Information, Keio University

力学系』という道具立てが必要になる」[3] といわれている．しかし現在のところ，そのような方法は考案されておらず，研究の進展を困難なものにしている．

このような背景をふまえて，本論文では，社会・経済を複雑系として「擬似的」に記述するための方法を提案する．それは，動的に変化しうるモデルの構造を定義し，「モデル・フレームワーク」として用意するという方法である．これにより，「振舞いのルールに従って状態が変化するモデル」だけでなく，「状態の変化によって振舞いのルールが変化するモデル」も表現できるようになる．

以下では，モデル・フレームワークとはどのようなものかを整理した後に，社会・経済のモデル・フレームワークを提案する．その後，適用事例を取り上げて，提案フレームワークの特徴を検討し，最後に，先行研究との差異について考察を行う．

2. モデル・フレームワーク

本論文では，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抽出し，「モデル・フレームワーク」として定義する．ここでは，抽象度の違いにより，「概念モデル・フレームワーク」と「シミュレーションモデル・フレームワーク」という 2 種類のフレームワークを考える．これらのフレームワークは一貫性を有しており，概念モデルの作成からシミュレーションモデルの実装までをシームレスに行うための支援をする（**図 1**）．

2.1 概念モデル・フレームワーク

概念モデル・フレームワークは，対象についての概念モデルを作成する際に，共通して登場する要素と構造を定義したものである．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」（What）を洗い出し，記述する際に，この概念モデル・フレームワークを用いることができる．加えて，概念モデル・フレームワークがシミュレーションモデル・フレームワークと一貫性を有することから，シミュレーションモデルへの移行をシームレスに行うことができる．概念モデルの作成において，概念モデル・フレームワークは，主に次の 3 つの役割を果たす．

2.1.1 現実世界の認識のための準拠枠の明示化

概念モデル・フレームワークは，対象となる現実世界を認識する際の準拠枠となる．概念モデル・フレームワークを，「世界の中に含まれるものを拾い集めようとして用いる，一種のふるい」[4] として用いることにより，動的でとらえどころのない世界を把握することが比較的容易になる．

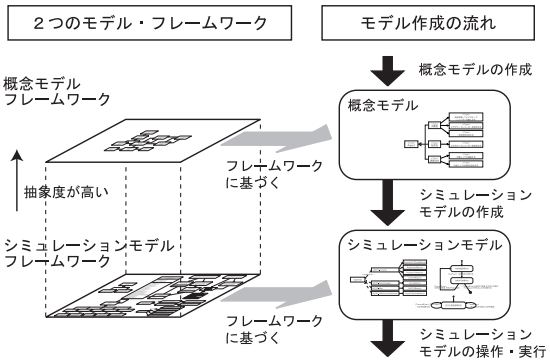

図 1 2 つのモデル・フレームワークとモデル作成の流れ
Fig. 1 Two model frameworks and the flow of model building.

2.1.2 概念モデルを記述するための語彙の提供

概念モデル・フレームワークにおけるモデル要素は，モデルを記述するための語彙として用いることができる．認識や分析で得られた概念は，それを記述するためのなんらかの表現手段が必要となるが，概念モデル・フレームワークは，そのドメインに特化した語句と，それらの可能な組合せの規則を提供する．

2.1.3 概念モデルの共有化とコミュニケーションの支援

概念モデル・フレームワークは，複数のモデル作成者に共通の視点を与えるため，モデル要素の粒度やとらえ方がモデルごとに異なってしまうという問題を回避できる．また，共有された語彙を用いることで，詳細な情報を伝達することなく重要な点を強調することができるため，円滑で効率的なコミュニケーションが可能となる．

2.2 シミュレーションモデル・フレームワーク

シミュレーションモデル・フレームワークは，得られた概念モデルを，シミュレーションモデルとして「どのように実現するか」（How）を規定するものである．シミュレーションモデル・フレームワークは，ソフトウェア・フレームワークとして実行環境の一部となることで，シミュレーションモデルを実行することができる．シミュレーションモデル・フレームワークは，主に次の 3 つの役割を果たす．

2.2.1 シミュレーションモデルへの変換方法の明示化

シミュレーションモデル・フレームワークは，シミュレーションモデルとして何をどのように記述すべきかを明らかにする．概念モデル・フレームワークと一貫性があるため，概念モデルをシミュレーションモデルに変換する手続きを事前に規定することができる．

2.2.2 シミュレーションモデル作成の支援

シミュレーションモデル・フレームワークは，モデルコンポーネントの仕様についての基本的な定義を行うため，シミュレーションモデルの実行環境をあらかじめ用意することが可能となる．また，一部のプログラムを自動生成する作成支援ツールを開発できるようになる．

2.2.3 シミュレーションモデルの共有化と再利用の支援

シミュレーションモデル・フレームワークは，モデルコンポーネント間の仕様やそれらの接続方法を規定するため，モデルコンポーネントを共有したり，再利用したりするための仕組みを提供する．

3. 提案モデル・フレームワーク：Boxed Economy Foundation Model（BEFM）

エージェントベースアプローチによって，社会・経済のモデルを作成するためのモデル・フレームワークとして，「Boxed Economy Foundation Model」（以下，BEFM）を提案する．BEFM の主な特徴としては，エージェント間の相互作用を，財（情報が付随することがある）のやりとりとして明示化するという点と，エージェントの行動を，エージェントとは別のモデル要素として定義するという点である．以下では，「BEFM 概念モデル・フレームワーク」，および「BEFM シミュレーションモデル・フレームワーク」について順に説明する．

3.1 BEFM 概念モデル・フレームワーク

BEFM 概念モデル・フレームワーク（BEFM Conceptual Model Framework）は，World，Space，Clock, Entity, Agent, Goods, Information, Behavior，Relation，Channel などのクラス（型）から構成されている（**図 2**）．各クラスについてまとめると以下のようになる．

3.1.1 World，Space，Clock

対象世界を表現する土台が「World」である．World は，その世界に固有の空間と時間によって規定されている．空間は「Space」によって，その世界の地理的な構造が表される．また，不可逆的な時間の流れを表すために「Clock」があり，この時間が経過することで現象が進行する．World には，後述の Entity，すなわち Agent と Goods が配置される．

3.1.2 Entity，Agent，Goods

世界に存在する実体が「Entity」である．Entity には，Agent と Goods の 2 種類があり，どちらにも後述する Information を付随させることができる．

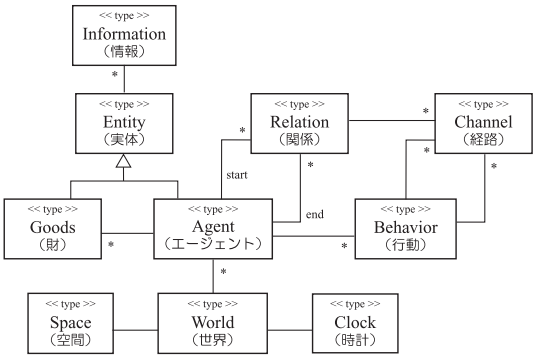

図 2 BEFM 概念モデル・フレームワークのクラス図
Fig. 2 Class diagram of BEFM Conceptual Model Framework.

社会・経済において，さまざまな活動を行う個人や社会集団（企業・政府・家族・学校・地域社会・国）が「Agent」である．また，動的に変化する環境や「モデルの外部」なども Agent として表現することがある．Agent は Goods を所有し，その行動（振舞い）は，後述するように Behavior として定義する．

Agent に所有し交換される有形/無形のものが「Goods」である．たとえば，自動車，石油，トウモロコシ，株，書籍，広告，回覧板，水，声，騒音，ごみ，貨幣などは，どれも Goods である☆．

3.1.3 Information

Entity が保持する情報は「Information」として表される．たとえば，Agent が記憶した情報や，Goods に付随して取引される情報などが，Information である．

Information は単独では存在せず，必ず Entity，つまり Agent や Goods によって保持されている．Agent が保持する Information は，主体の内部に貯蔵されている「記憶」や「遺伝子」をはじめとして，Agent のさまざまな属性を表現する．Goods に Information が付随するというのは，たとえば，新聞は「紙」（Goods）に「記事内容」（Information）が付随したものであり，会話は無形で瞬間的な「声」（Goods）に「会話内容」（Information）が付随したものととらえるということである．Goods に付随している Information は，その Goods が他の Agent に渡されると，Goods とともに送られる．また，Agent によって保持されている Information は，すでに持っている Goods か，そのために作成した Goods に付随させて，他の Agent に送ることができる．

☆ BEFM における「Goods」とは，人間の欲求を充足する性質を持つという経済学における狭義の意味ではなく，世界におけるさまざまなものを示す広義の概念として用いている．

3.1.4 Behavior

エージェントの行動は，「Behavior」として表される．たとえば，企業における生産行動や販売行動，個人における購買行動や労働行動などは，どれも Behavior である．オブジェクト指向によってエージェントをモデル化する場合には，エージェントの行動を Agent クラスの操作（メソッド）として記述するのが一般的であるが，BEFM では行動をモデル要素の 1 つとして分離する．これは，状況によって振舞いが動的に変化することを表現できるようにするためである．

BEFM では，Agent は複数の Behavior を並列的に実行することができる．内部状態を各 Behavior に持たせることで，複数の行動の多様な組合せを実現することができる．

3.1.5 Relation，Channel

ある Agent から他の Agent への関連性は，「Relation」によって表される．これにより，友人関係や家族関係，雇用関係などの関係を表現することができる．実際のコミュニケーションの際には，この Relation に基づいて開設されるコミュニケーション・パスである「Channel」を通じて，商品や会話，貨幣などの Goods をやりとりする．

Relation と Channel は，密接に関わっている概念であるが，これらは別のものである．Relation は「参加」という「構造的関係」[5] を表しており，Channel は「活動」という「過程の関係」[5] である．また，Relation は Agent 間の関連性を表すが，それに基づいて開設される Channel は，Behavior どうしを接続する．同一の Agent 内における Behavior 間のやりとりであっても，Channel によるやりとりで表現される．これにより，エージェント間のコミュニケーションにおいても，エージェント内部の行動の連携においても，Behavior どうしのやりとりはすべて，Channel を通じた Goods のやりとりとして抽象化されることになる．

3.2 BEFM シミュレーションモデル・フレームワーク

BEFM シミュレーションモデル・フレームワーク（BEFM Simulation Model Framework）は，BEFM 概念モデル・フレームワークと一貫性を持つように設計されている．また，このフレームワークに基づいて作成されたモデルは，Boxed Economy Simulation Platform [6] 上でシミュレートすることができる（**図 3**）．BEFM シミュレーションモデル・フレームワークは，モデルを動作させるための多くのクラスから構成されているが，ここでは，モデル表現に関連する特徴的な点についてのみ言及することにしたい．

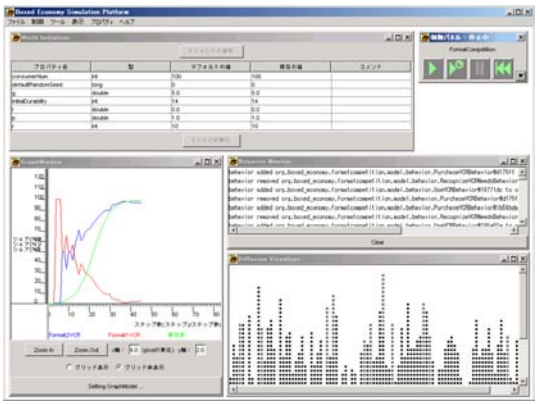

図 3 Boxed Economy Simulation Platform におけるシミュレーション画面
Fig. 3 ScreenShot of Boxed Economy Simulation Platform.

3.2.1 シミュレーションモデルへの変換方法

BEFM シミュレーションモデル・フレームワークでは，概念モデルのモデル要素のそれぞれに対し，種類の違いをどのように表現するのかを規定している☆．種類の表現方法としては，「継承」による方法と「パワータイプ」による方法の 2 種類がある．

継承による方法は，クラスを特化したサブクラスを定義することで，種類の違いをクラスレベルで実現するものである．この方法を用いるものには，Behavior と Information がある．これらは，種類の違いが，単なる属性値の差異ではなく，振舞いや内部構造の差異であるためである．Behavior については，後述するように，記述方法が定義されている（3.2.3 項参照）☆☆．

パワータイプによる種類の表現は，種類クラスを作って，その種類クラスのインスタンスレベルで多様性を実現するというものである [7),8)]．この方法によって種類を表現するものは，Agent と Goods，そして Relation である．この場合，新たにクラスを作成することなく，インスタンスレベルで差異を表現することができる☆☆☆．BEFM シミュレーションモデル・フレームワークでは，後述する「Type」クラスをパワー

☆ このような方針が必要なのは，概念モデルでは，モデルに登場する要素がすべて BEFM 概念モデル・フレームワークのクラス（型）を「特化」するかたちで記述されるのに対して，シミュレーションモデルの設計・実装では，それらを必ずしもクラスの「継承」として実現するわけではないからである．

☆☆ Information の形式は，モデルによってさまざまであるため，BEFM では詳細を定めていない．

☆☆☆ 継承を用いた実装には，動的な変更ができないことや，多重継承ができないなどの限界があり，また，実装のしやすさや実行効率などの問題で，継承を用いるべきでないこともある．この問題を回避するための代表的な方法が，パワータイプを用いた設計である．

タイプとして用いることができる（3.2.4 項参照）．

3.2.2 オブジェクト コンポジションを用いたエージェントの設計

BEFM シミュレーションモデル・フレームワークでは，Behavior をオブジェクトとして外部化し，「オブジェクトコンポジション」によって付加するという方法を採用している．オブジェクトコンポジションとは，役割を外部化するためのオブジェクトを用意して振舞いを委譲し，そのオブジェクトを実行時に関連づけるという設計のことである[9),10)]．同様に，エージェントの記憶（情報）や，他のエージェントへの関係も，Information や Relation のオブジェクトコンポジションとして保持する設計になっている．

このような設計により，シミュレーションモデルにおけるコーディング作業は，Behavior や Information の記述が中心となり，それらの組み合わせ方によって，エージェントが設定できるようになる．この場合，Agent 自体はそれらの行動を束ねる役割を果たしているにすぎない．たとえば，Agent をインスタンス化すると，単なる Agent オブジェクトが得られるが，そこに PurchaseBehavior を加えると，購買行動を行う「消費者エージェント」となる．このようなエージェントの設計は，新しい行動の追加や削除，そして行動の組み換えなどを簡単に行えるという柔軟性がある[11)]．また，すでに作成されている Behavior を利用することも可能になるため，再利用性を考慮した設計といえる．

3.2.3 状態機械としての Behavior

BEFM シミュレーションモデル・フレームワークでは，エージェントの持つそれぞれの Behavior を「状態機械」として定義する．状態機械とは，何らかのトリガとなるイベント（影響を及ぼすさまざまな出来事）を受け取ると，現在の状態に応じた「アクション」（動作）を行い，新しい状態へ遷移するシステムである．ある時点を見てみると，Behavior オブジェクトは，必ずどれか 1 つの状態にとどまっている．Behavior の状態遷移を引き起こすイベントには，時間が経過したことを表す「TimeEvent」と，Goods が送られてきたことを表す「ChannelEvent」がある．

このような設計により，外界のイベントの種類と，現在の自分の状態によって，振舞いが異なるというモデルを実現できる．本来，システムの内部状態とは，システムのすべてのパラメータの値の集合のことであるが，状態機械では，それらの状態のうちの一部を意識的に切り出して注目することになる．このことは，本論文で目指している「状態の変化によって振舞いのルールが変化するモデル」を記述する際に，モデルの状態の複雑さを隠蔽し，注目すべき状態を強調できるという点で利点となる．

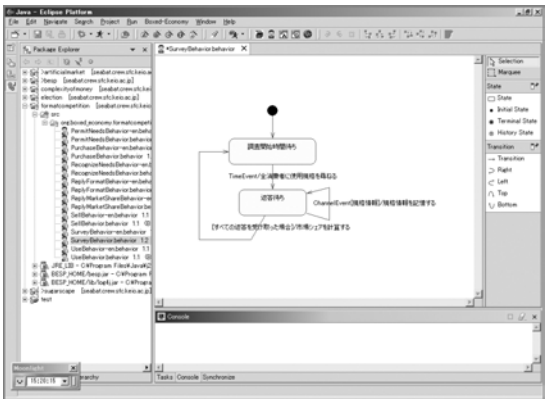

図 4 Behavior の作成を支援するコンポーネントビルダー
Fig. 4 Component Builder for building Behavior.

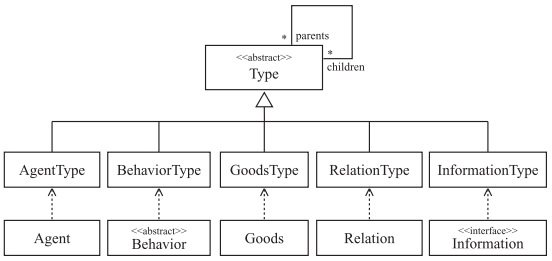

図 5 BEFM シミュレーションモデル・フレームワークにおける Type クラス
Fig. 5 Type classes in BEFM Simulation Model Framework.

エージェントが複数の行動を並列的に動作させる場合には，その内部状態は複雑にならざるをえないが，BEFM に基づくモデル化では，Behavior という分かりやすい単位ごとに状態を把握することができる．

私たちは状態遷移のフレームワークに基づいた Behavior を簡単に作成するための「コンポーネントビルダー」を提供している．モデル作成者は，GUI によって Behavior の状態遷移図を記述することで，Java プログラムの雛型を自動生成させることができる（**図 4**）．

3.2.4 Type による種類の表現

BEFM シミュレーションモデル・フレームワークでは，モデルに存在するオブジェクトの要素を分類するための識別子として，「Type」クラスが用意されている．Type には，AgentType，GoodsType，InformationType，RelationType，BehaviorType の 5 種類がある（**図 5**）．

Type の導入により，柔軟な検索・取得・識別が可

能となる．たとえば，同じ振舞いをする Agent であっても，異なる Type が付加してあれば，別々に識別可能となる．また，Type 間に親子関係を定義することができるので，上位概念・下位概念を表現することができる．これによって，異なる Behavior であっても，同じ親 Type を持つものどうしであれば，一括して扱うことができるのである．

4. 提案モデルフレームワークの適用事例

BEFM の特徴と有効性を示すために，以下では 3 つの事例を取り上げることにする．

4.1 規格競争モデル

BEFM に基づくモデルでは，エージェント自身が必要に応じて Behavior を追加・削除することができる．このことを示すために，日本における家庭用 VCR（ビデオカセットレコーダ）の規格競争モデル[12] を取り上げたい．このモデルは，消費者が，他の消費者の選択に影響を受けて規格選択を行うというモデルである．

この規格競争モデルの基本的な流れをまとめると，次のようになる．ある特定の割合で，家庭用 VCR 製品に対する欲求を認識した消費者エージェントが，自分の周囲の規格シェアと，市場全体の規格シェアを調べる．その情報探索で得た情報を用いて，購買前代替案評価を行い，その評価をもとに一方の規格を購入する．そして，耐久残存年数が経過して故障するまで使用し，故障した場合には，所持製品を処分し，新たな家庭用 VCR を購入する．

この規格競争モデルの全体像を，BEFM 概念モデル・フレームワークに基づいて記述したものが，**図 6** と**図 7** である．この概念モデルに基づいてシミュレーションモデルを作成すると，全体像は**図 8** と**図 9** のようになる☆．各エージェントが持つ Behavior を状態遷移図で記述すると，**図 10**，**図 11**，**図 12** のようになる．これらの状態遷移図における「〜Behavior を起動する」の部分が，Behavior の生成と削除を行う部分である．RecognizeVCRNeedsBehavior では，PurchaseVCRBehavior を生成して，自らはその役割を終えて消滅する（図 10）．また，PurchaseVCRBehavior では，VCR を受け取った後に UseVCRBehavior を生成し，自らはその役割を終えて消滅する（図 11）．そして，所持している家庭用 VCR が耐久残存年数を迎えた場合には，次のステップで PurchaseVCRBehavior を生成し，自らはその役割を終えて消滅する（図 12）．Consumer エージェントにおける Behavior の追加と削除の流れをまとめると，**図 13** のようになる．

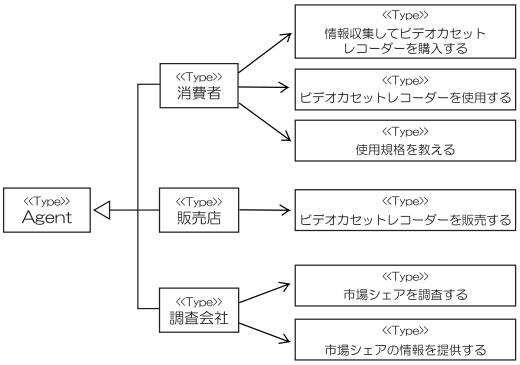

図 6 規格競争モデルにおけるエージェントとその行動（概念モデル・クラス図）

Fig. 6 Agents and Behaviors in the format competition model (class diagram of conceptual model).

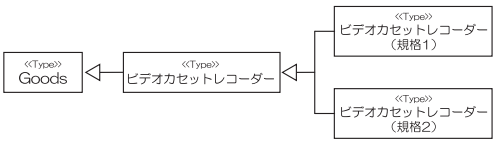

図 7 規格競争モデルにおける財（概念モデル・クラス図）

Fig. 7 Goods in the format competition model (class diagram of conceptual model).

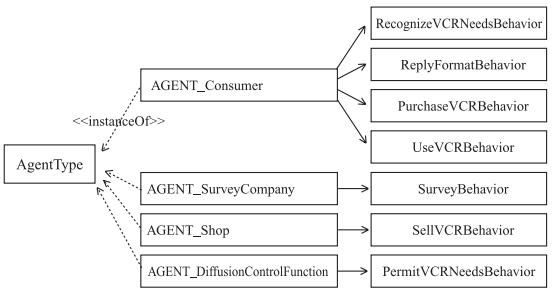

図 8 規格競争モデルにおける AgentType と Behavior（シミュレーションモデル・クラス図）

Fig. 8 AgentTypes and Behaviors in the format competition model (class diagram of simulation model).

図 9 規格競争モデルにおける GoodsType（シミュレーションモデル・クラス図）

Fig. 9 GoodsTypes in the format competition model (class diagram of simulation model).

☆ 図 9 のように，GoodsType の親子関係を定義することで，モデル上の Goods の概念の体系化を行っている．市場における「VCR」の普及率を数えたい場合には，Consumer エージェントに「GOODS_VCR」を所有しているかを尋ねればよく，各規格の市場シェアを調べたい場合には，「GOODS_Format1VCR」と「GOODS_Format2VCR」のそれぞれについて尋ねればよいことになる．

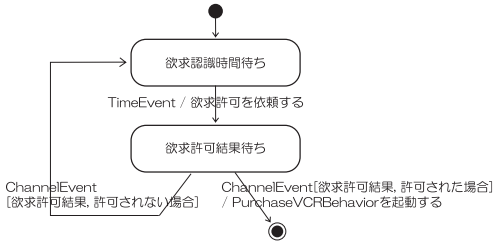

図 10 Consumer エージェントの RecognizeVCRNeedsBehavior（状態遷移図）
Fig. 10 RecognizeVCRNeedsBehavior of Consumer Agent (statechart diagram).

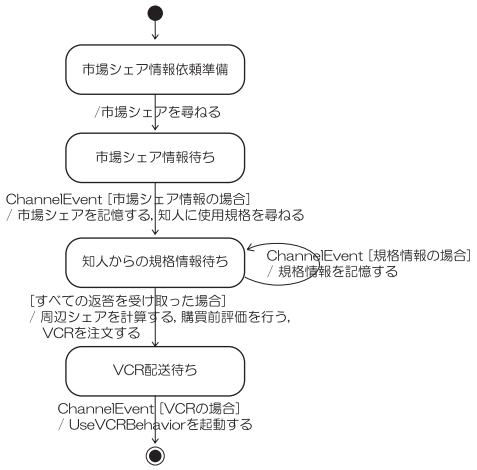

図 11 Consumer エージェントの PurchaseVCRBehavior
Fig. 11 PurchaseVCRBehavior of Consumer Agent (statechart diagram).

図 12 Consumer エージェントの UseVCRBehavior
Fig. 12 UseVCRBehavior of Consumer Agent (statechart diagram).

4.2 人工証券市場モデル

BEFM に基づくモデルでは，Behavior の内容をシミュレーション実行時に動的に変化させる（組み換える）ことができる．このことを示すために，人工証券市場モデル[13] を取り上げたい．このモデルは，各トレーダが市場の動向によって自分の意思決定アルゴリズムを変化させるモデルである．

Trader エージェントは，意思決定を行って注文を出す OrderBehavior を持つが，これは内部の意思決定アルゴリズムが異なる OptimisticOrderBehavior，PessimisticOrderBehavior，FundamentalOrderBehavior

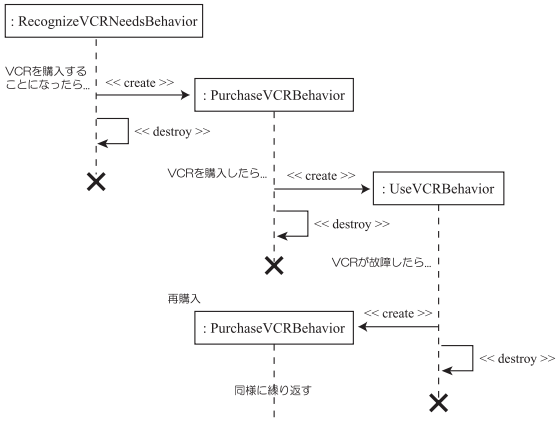

図 13 規格競争モデルにおける Behavior の生成と消滅
Fig. 13 Creation and destruction of Behavior in the format competition model (sequence diagram).

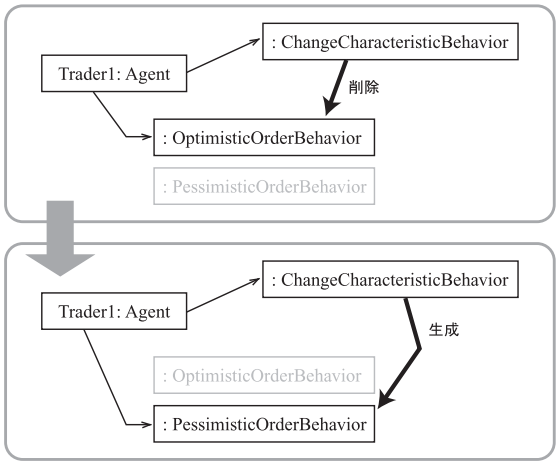

図 14 人工証券市場における ChangeCharacteristicBehavior による意思決定行動の変更
Fig. 14 Switching the decision-making Behavior by ChangeCharacteristicBehavior in the artificial financial market model.

として実現される．各 Trader エージェントは，この 3 つの意思決定行動のうちのどれか 1 つを採用しており，ChangeCharacteristicBehavior によってその種類を変えることができる．ChangeCharacteristicBehavior は，まず最初に意思決定の変更を行うべきかを計算し，変更する場合には，現在の意思決定行動のオブジェクトを削除し，新しい意思決定行動のオブジェクトを追加する．たとえば，OptimisticOrderBehavior を PessimisticOrderBehavior に変更する場合には，**図 14** のようになる．

このような動的な変更が可能なのは，意思決定アルゴリズムが Behavior として個々に定義されているため，組み換えが可能であるということと，組

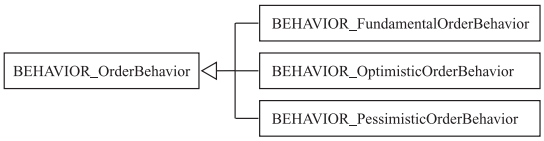

図 15 人工証券市場における BehaviorType 間の親子関係
Fig. 15 The hierarchy of BehaviorTypes in the artificial financial market model.

み換えた場合でも，他の Behavior との相互作用がこれまでどおり実行できるためである．Behavior から Behavior への Channel の開設は，相手のエージェントを指定するための RelationType と，やりとりを行う Behavior を特定するための BehaviorType を指定することで行われる．ここでは，BEHAVIOR_OptimisticOrderBehavior と BEHAVIOR_PessimisticOrderBehavior の親 Type が，BEHAVIOR_OrderBehavior であるため（**図 15**），どちらの場合でも，BEHAVIOR_OrderBehavior 宛に送られてくる財（および情報）を受信することができる．

4.3 Sugarscape

BEFM は，社会・経済における主体間の相互作用に注目して構成されたフレームワークであるため，「環境」が動的に変化するモデルを扱う場合には，工夫が必要となる．このことを示すために，Sugarscape[14] の基本モデルを取り上げたい．

Sugarscape は，2 次元セル空間の一部に，時間とともに再生する砂糖が配置されており，その砂糖をエージェントが取得するというモデルである．この Sugarscape を直接的に表現するならば，エージェントは，環境（空間に配置された砂糖）と相互作用することになるが，BEFM では，このようなモデルをそのまま記述することはできない．なぜなら，空間に（どのエージェントにも所有されていない）Goods を配置することや，その Goods が自動的に変化することを表現できないからである．この制約が生じるのは，BEFM が「時間経過とともに変化するものは，エージェントとしてモデル化する」という方針をとっているためである．

空間に配置された砂糖が時間とともに再生することを表現するためには，次の 2 つの実現方法が考えられる．第 1 の方法は，2 次元セル空間の全セルに，砂糖を生成・保持するエージェントを配置するというものである．第 2 の方法は，2 次元セル空間上の砂糖を管理する環境エージェントを作成するという方法である．ここでは，実装の容易さと実行負荷を考慮して，後者の方法によってモデル化することにする．

Sugarscape における「エージェント」を SSAgent とし，砂糖の山などの環境を制御するエージェントを EnvironmentAgent とする．SSAgent は，2 次元セル空間上の 1 セルに存在し，時間とともに移動する．EnvironmentAgent は，セル上のどこにも存在しないが，SSAgent と関係を持ち，砂糖の取引を行うことができる．エージェントが配置される 2 次元空間は，Space クラスを継承した CellSpace クラスを作成して表現する．この CellSpace クラスを用いて，Cell クラスと Agent クラスを関連付けることによりエージェントのセルへの配置を表現する．また，この関連付けを変えることで，エージェントの移動を表現する．各セルの砂糖の量は，EnvironmentAgent の持つ SSFieldInformation によって保持されている．以上をまとめると，**図 16** のようになる．

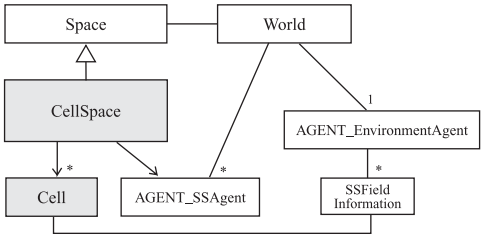

図 16 Sugarscape モデルのための CellSpace クラス
Fig. 16 CellSpace class for Sugarscape model.

5. 考　　察

本論文で提案したモデル・フレームワークの有効範囲と先行研究との差異を検討することにしたい．先行研究との比較おいては，概念モデル・フレームワークに関するものと，シミュレーションモデル・フレームワークに関するものに分けて議論する．

5.1 提案モデル・フレームワークの有効範囲

提案モデル・フレームワークの適用が有効なモデルは，第 1 に，エージェントが財や情報をやりとりするモデルである．財や情報をエージェントの操作（メソッド）の中に含めるのではなく，オブジェクトとして明示化するため，財の取引を追跡的に分析することができる．また，Type によって財の種類を体系化することができる．本論文で取り上げた規格競争モデルや人工証券市場モデルが，これらの特徴を活かしている．この 2 つの例のほかにも，物々交換によって貨幣の自生や自壊が起こるモデル[15] などのように，主体間の相互作用が中心となるモデルであれば，適しているといえる．

第 2 に，状態によって振舞いのルールが変化するモデルに有効である．このことは，規格競争モデルや人工証券市場モデルで見てきたとおりである．この特徴

は，構成要素の振舞いのルールが状況によって動的に変化する「複雑系」を記述することを可能とする．

これに対し，提案モデル・フレームワークの適用が有効とはいえないモデルとしては，第1に，環境が動的に変化するモデルがあげられる．これは，Sugarscapeの事例で見たとおりである．第2に，空間を動き回るエージェントどうしが「出会う」ということも，直接的な方法では表現できない．そのようなモデルを実現するためには，両者の距離をモニタリングするエージェントを作成する必要がある．このように，提案フレームワークでは，エージェントと環境が相互作用するモデルを作成する場合には，環境も一種のエージェントとしてモデル化する必要があることが分かる．

5.2 概念モデル・フレームワークに関連する先行研究

概念モデル・フレームワークの先行研究としては，社会・経済の基本構造を記述した社会システム論[16),17)]がある．そこでは，「主体」，「主体間の相互作用」，「複合主体」，「社会システム」，「社会システムとしてのネットワーク」というように，本論文で扱う多くのモデル表現やシステム構成の重要性を指摘している．また，ビジネスモデルをオブジェクト指向によってモデル化し，UML（統一モデリング言語）[18)]で記述するという試み[19),20)]もある．

しかし，これらの先行研究で焦点になっているのは，社会システムやビジネスモデルの記述であり，これらを操作して振舞いを理解したり，定量的な分析を行ったりということに関してはほとんど考慮されていない．本論文では，社会・経済の記述だけでなく，シミュレーションの実行までを含めて，モデル・フレームワークと実行環境，そして作成支援ツールを提案している．

また別の観点では，本論文で概念モデル・フレームワークの役割としてあげている3点は，オブジェクト指向における「アナリシスパターン」[8)]の役割と類似している．これは，概念モデル・フレームワークは，概念モデルを作成する際に「こうするとうまくいく」というパターンの側面を含んでいるためである．しかし，本論文で，概念モデル・フレームワークを「アナリシスパターン」と呼ばないのは，それが，パターンの側面だけでなく，フレームワークの側面をあわせ持っているためである．そのため，使用者にとっては，フレームワークの要素と構造は所与で強制力を持っており，変更することができない（変更すると意味がない）という特徴がある．もちろん，この枠組みを設けることで，モデル作成の指針や支援ツールを提供することが可能になっていることはいうまでもない．

5.3 シミュレーションモデル・フレームワークに関連する先行研究

エージェントベースシミュレーションを支援するための言語やツールは，これまでにもいくつか開発されている．たとえば，Swarm Simulation System[21)]，RePast[22)]，Ascape[23)]．KK-MAS[24)]などがあり，どれも，シミュレーション作成におけるプログラミング支援の必要性という問題意識から，汎用的なライブラリの提供を行っている．しかし，本論文の言葉でいうならば，これらの支援システムは，シミュレーションモデル・フレームワークを提供するが，概念モデル・フレームワークを提供していないか，明示化していない．本論文の提案では，シミュレーションモデル・フレームワークの抽象度を高くした概念モデル・フレームワークも含めて，一貫したモデル・フレームワークを提供していることに特徴がある．概念モデル・フレームワークを提供するのは，シミュレーションモデル・フレームワークが，モデル化の認知枠として用いるには粒度が細かすぎるということと，社会・経済のモデル作成者が，必ずしもオブジェクト指向による設計・実装に馴染みがあると想定できないからである．

さらに，シミュレーションモデル・フレームワークの設計においても，従来のフレームワークに比べて優位性がある．提案フレームワークでは，柔軟性と再利用性を考慮したエージェントの設計のために，「継承」ではなく「オブジェクトコンポジション」を用いている．オブジェクト指向によってエージェントを表現する場合，従来は，継承を用いてクラスとして設計することが多かった（たとえば，先行研究25)を参照）．そのような設計では，行動の種類そのものが動的に変化するエージェントを作成する場合や，エージェントの一部を再利用する場合に限界が生じてしまうのである．

6. さいごに

ここ半世紀の間，ソフトウェア工学は大規模で複雑なシステムをどのように構築するのかという問題に取り組んできた．その中で生まれた考え方は，社会科学の研究においても有効であると，私たちは考えている．特に，オブジェクト指向分野で標準化されたUML（統一モデリング言語）を用いてモデル記述を行うということや，開発プロセスに関する議論は，社会科学のモデル化や研究プロセスの発展に大きな影響を与えると思われる．

本論文では，社会・経済のモデル化に，オブジェクト指向とフレームワークの考え方を導入して，Boxed Economy Foundation Model（BEFM）を提案した．

「概念モデル・フレームワーク」と「シミュレーションモデル・フレームワーク」の 2 つを含む BEFM は，社会・経済のモデル化のプロセスを明確化し，シミュレーション分析へとつなぐ重要な役割を担っている．これらの成果により，各研究者が複雑系の記述を行えるようになるだけでなく，他の研究者のモデルの追試や拡張を行うことを支援し，健全な研究コミュニティの発展に寄与することを期待したい．

謝辞 慶應義塾大学 SFC の Boxed Economy Project のメンバ，特にモデル作成に協力してくださった津屋隆之介さんと山田悠さん，BEFM の作成に大きな貢献をした上橋賢一さん，田中潤一郎さん，浅加浩太郎さん，高部陽平さん，そして，開発に携わった青山希さんに感謝したい．また，日頃から，研究指導をしていただいている大岩元先生と竹中平蔵先生にも感謝の意を述べたい．なお，本研究は，慶應義塾大学およびフジタ未来経営研究所（複雑系の経済・経営プロジェクト，社会シミュレーションの実践的応用プロジェクト），文部省科学研究費補助金（特別研究員）のサポートのもとに遂行されたものである．

参 考 文 献

1) 出口 弘：複雑系としての経済学：自律的エージェント集団の科学としての経済学を目指して，日科技連 (2000).
2) 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).
3) 金子邦彦，池上高志：複雑系の進化的シナリオ：生命の発展様式，朝倉書店 (1998).
4) James, W.: *The Philosophy of William James: Drawn from His Own Works*, The Modern Library, Introduction by H.M. Kallen (1925).
5) 西部 邁：ソシオ・エコノミックス，中央公論社 (1997).
6) Iba, T., Chubachi, Y., Matsuzawa, Y., Asaka, K. and Kaiho, K.: Resolving the Existing Problems by Boxed Economy Simulation Platform, *Agent-based Approaches in Economic and Social Complex Systems*, pp.59–68, IOS Press (2002).
7) Martin, J. and Odell, J.J.: *Object-Oriented Methods: A Foundation*, PTR Prentice Hall (1995). 三菱 CC 研究会 OO タスクフォース（訳）：オブジェクト指向方法序説：基盤編，トッパン (1995)
8) Fowler, M.: *Analysis Patterns: Reusable Object Models*, Addison-Wesley (1996). 堀内 一，友野晶夫，児玉公信，大脇文雄（訳）：アナリシスパターン：再利用可能なオブジェクトモデル，新装版，ピアソンエデュケーション (2002).
9) Gamma, E., Helm, R., Johnson, R. and Vlissides, J.: *Design Patterns: Elements of Reusable Object-Oriented Software*, Addison-Wesley (1995). 本位田真一，吉田和樹（監訳）：オブジェクト指向における再利用のためのデザインパターン，改訂版，ソフトバンクパブリッシング (1999).
10) Coad, P. and Mayfield, M.: *Java Design: Building Better Apps & Applets*, 2nd edition, Yourdon Press, Prentice Hall PTR (1999).
11) 井庭 崇：エージェントベース経済シミュレーションのためのエージェント設計論，オペレーションズ・リサーチ経営の科学，Vol.46, No.10, pp.561–567 (2001).
12) 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).
13) 山田 悠，井庭 崇：制限値幅が市場効率性に与える影響の分析：人工市場アプローチによる分析，第 7 回進化経済学会 (2003).
14) Epstein, J.M. and Axtell, R.: *Growing Artificial Societies: Social Science from the Bottom Up*, Brookings Institution Press/The MIT Press (1996). 服部正太，木村香代子（訳）：人工社会：複雑系とマルチエージェント・シミュレーション，共立出版 (1999).
15) 安冨 歩：貨幣の複雑性：生成と崩壊の理論，創文社 (2000).
16) 公文俊平：社会システム論：社会科学総合化の試み，日本経済新聞社 (1978).
17) 公文俊平：情報文明論，NTT 出版 (1995).
18) Rumbaugh, J., Jacobson, I. and Booch, G.: *The Unified Modeling Language Reference Manual*, Addision Wesley Longman (1999). 石塚圭樹（監訳），日本ラショナルソフトウェア株式会社（訳）：UML リファレンスマニュアル，ピアソン・エデュケーション．
19) Eriksson, H.-E. and Penker, M.: *Business Modeling with UML: Business Patterns at Wwork*, John Wiley & Sons (2000). 鞍田友美，本位田真一（監訳）：UML によるビジネスモデリング，ソフトバンクパブリッシング (2002).
20) Marshall, C.: *Enterprise Modeling with UML: Designing Successful Software through Business Analysis*, Addison Wesley (1999). 児玉公信（訳）：企業情報システムの一般モデル：UML によるビジネス分析と情報システムの設計，ピアソンエデュケーション (2001).
21) Luna, F. and Stefasson, B. (Eds.): *Economic Simulations in Swarm: Agent-Based Modelling and Object Oriented Programming*, Kluwer Academic Publishers (2000).

22) RePast: RePast, *The University of Chicago's Social Science Research*. http://repast.sourceforage.net/
23) Parker, M.T.: What is Ascape and Why Should You Care?, *Journal of Artificial Societies and Social Simulation*, Vol.4, No.1 (2001). http://www.soc.surrey.ac.uk/JASSS/4/1/5.html
24) 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).
25) Bruun, C.: Prospect for an Economics Framework for Swarm, *Agent-Based Methods in Economics and Finance: Simulation in Swarm*, Luna, F. and Perrone, A. (Eds.), Kluwer Academic Publishers (2002).

(平成 15 年 2 月 5 日受付)
(平成 15 年 5 月 6 日採録)

井庭 崇（正会員）

1974 年生．1997 年慶應義塾大学環境情報学部卒業．2003 年慶應義塾大学大学院政策・メディア研究科博士（政策・メディア）取得．現在，千葉商科大学政策情報学部専任教員（助手），およびフジタ未来経営研究所リサーチフェロー．著書に『複雑系入門』（共著），訳書に『社会シミュレーションの技法』（共訳）等．人工知能学会，進化経済学会，社会・経済システム学会，政策分析ネットワーク各会員．

中鉢 欣秀（正会員）

1972 年生．1995 年慶應義塾大学環境情報学部卒業．1997 年慶應義塾大学大学院政策・メディア研究科修士課程修了．2001 年同後期博士課程単位取得退学．1997 年合資会社ニューメリック設立．現在，慶應義塾大学環境情報学部非常勤講師．オブジェクト指向分析設計技術の研究，およびコンサルテーション業務に従事．電子情報通信学会会員．

松澤 芳昭

1977 年生．2000 年慶應義塾大学環境情報学部卒業．2002 年慶應義塾大学大学院政策・メディア研究科修士課程修了．2000 年より，合資会社ニューメリック．現在，慶應義塾大学 SFC 研究所訪問研究員，および千葉商科大学政策情報学部・商経学部非常勤講師．オブジェクト指向技術を利用したソフトウェア開発に従事．プログラミング教育に興味を持つ．

海保 研

1974 年生．2000 年慶應義塾大学環境情報学部卒業．2000 年より，合資会社ニューメリック．現在，慶應義塾大学 SFC 研究所訪問研究員．Java による Web アプリケーション開発等，ソフトウェア開発に従事．

武藤 佳恭

1955 年生．1978 年慶應義塾大学工学部卒業．1983 年慶應義塾大学大学院工学研究科博士（工学）取得．ケースウェスタンリザーブ大学準教授等を経て，現在，慶應義塾大学環境情報学部教授．著書に『だれでもわかるディジタル回路』（共著），『ニューラルコンピューティング』，『応用事例ハンドブックニューラルコンピューティング』（編著），“Neural Netowork Parallel Computing” 等多数．