

Boxed Economy Simulation Platformの仕組みと支援ツールの提案

海保 研[†] 浅加浩太郎^{††} 松澤 芳昭^{††} 中鉢 欣秀[†] 津屋隆之介^{†††}
山田 悠^{†††} 井庭 崇^{††,††††}

[†] 慶應義塾大学 SFC 研究所
〒 252-8520 神奈川県藤沢市遠藤 5322
^{††} 慶應義塾大学政策・メディア研究所
^{†††} 慶應義塾大学総合政策学部
^{††††} 日本学術振興会 特別研究員

E-mail: [†]{rx78g,gogotea,macchan,yc}@crew.sfc.keio.ac.jp, ^{††}{s99593rt,s99940yy,iba}@sfc.keio.ac.jp

あらまし Boxed Economy プロジェクトでは、経済社会モデルを実際にコンピューター上でシミュレートさせるためのソフトウェアである Boxed Economy Simulation Platform を開発した。Boxed Economy Simulation Platform では、シミュレーション実行環境を作成するためのソフトウェア部品としてモデルコンポーネントとプレゼンテーションコンポーネントの2種類のコンポーネントを定義している。これらのコンポーネントは利用者によって作成されることを前提としているが、それぞれのコンポーネントのフレームワークの提供、外部入出力機能の提供、制御コマンド群の分離など作成を簡単にする工夫を行っている。また、グラフの表示やモデルの作成など BESP を拡張するツールがある。

キーワード エージェントベースモデル, シミュレーション, オブジェクト指向, フレームワーク, コンポーネント

Architecture and Supporting Tools for Boxed Economy Simulation Platform

Ken KAIHO[†], Kotaro ASAKA^{††}, Yoshiaki MATSUZAWA^{††}, Yoshihide CHUBACHI[†], Ryunosuke TSUYA^{†††}, Yuu YAMADA^{†††}, and Takashi IBA^{††,††††}

[†] Keio Research Institute at SFC
5322 Endo, Fujisawa 252-8520
^{††} Graduate School of Media and Governance, Keio University
^{†††} Faculty of Policy Management, Keio University
^{††††} JSPS Research Fellow

E-mail: [†]{rx78g,gogotea,macchan,yc}@crew.sfc.keio.ac.jp, ^{††}{s99593rt,s99940yy,iba}@sfc.keio.ac.jp

Abstract

Key words agent-based model, simulation, object-orientation, framework, component

1. はじめに

近年「エージェントベースモデル」による市場・経済の研究が盛んに行われている [1] [2] [3] [4]。エージェントベースのアプローチでは、市場や経済の構成要素をエージェント (自律的主体) としてモデル化し、それらのミクロ的な相互作用からなるシステムとして全体を構成する。エージェントベースのアプローチでは、主にシミュレーションによって理論の模索が行われているが、シミュレーションによる研究方法には特有の問題が存在している [9]。それは、シミュレーションを作成するために多くのプログラミングが必要であるということや、モデルからプログラムへの変換について検証しなければならないということなどである。このような特有の問題は、社会学者などの参加の障壁となっているため、シミュレーションの作成や共有・蓄積を支援するための仕組みについて考えることが急務であるといえる^(注1)。私たちはこのような問題意識から、エージェントベースモデルによる経済シミュレーションの作成・実行・分析を支援するプラットフォームとして「Boxed Economy Simulation Platform」(BESP) を開発し提案している。

本論文では BESP の基本的な仕組みを紹介し、「利用の容易さ」や「高い拡張性」を実現するための設計上の工夫について説明する。さらに実際に作成したコンポーネントと支援ツールについて紹介する。

2. Boxed Economy Simulation Platform (BESP) の基本構造

Boxed Economy Simulation Platform (BESP) は、エージェントベース経済モデルのシミュレーションを作成・実行・分析するためのプラットフォームである (図 1)。BESP の構造の最大の特徴は、コンポーネントとフレームワークによるソフトウェア設計を導入しているという点にある。シミュレーションのモデルや実行環境をコンポーネントとして分割して定義しているため、それらを組み替えることによって、ユーザー独自のシミュレーションを柔軟かつ容易に構築できるのである。加えて BESP では、コンポーネントの作成を容易にするための仕組みも提供されている。以下では、BESP の構造と、容易さと拡張性の鍵となるコンポーネントおよびイベントの

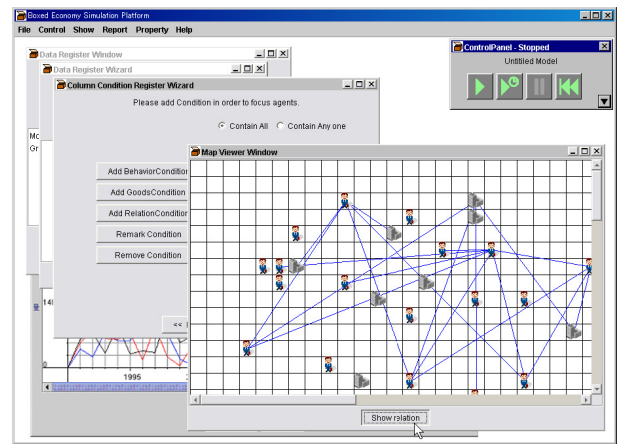
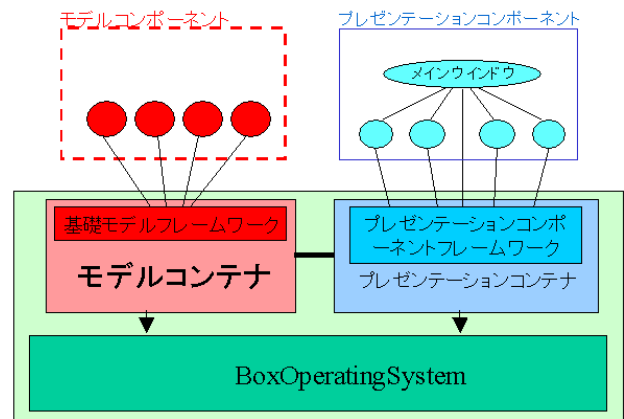


図 1 Boxed Economy Simulation Platform(BESP) の画面



Boxed Economy Simulation Platform (BESP)

図 2 Boxed Economy Simulation Platform(BESP) のモジュール構成

仕組みについて説明する。

2.1 BESP の構造

BESP は、「Box Operating System」(BOS)、「モデルコンテナ」、「プレゼンテーションコンテナ」、「モデルコンポーネント」、「プレゼンテーションコンポーネント」という 5 つのモジュールから構成されている (図 2)。

最初の BOS、モデルコンテナ、プレゼンテーションコンテナは、シミュレーション支援の基盤となる部分であり、そこに配置されるコンポーネントに対してフレームワークやその他のさまざまなサービスを提供する。この基盤の上には「モデルコンポーネント」と「プレゼンテーションコンポーネント」という 2 種類のコンポーネントが配置される。「モデルコンポーネント」はシミュレーションのモデルを構成するものであり、「プレゼンテーションコンポーネント」はシミュレーションのユーザーインターフェース

(注1): エージェントベースの分野におけるコラボレーションのためのテストベッドとして [7] [6] などがある。

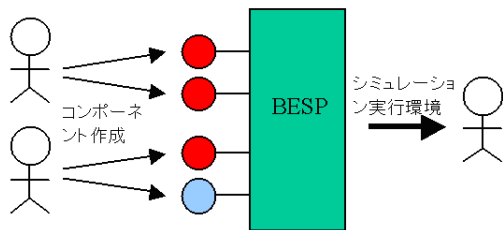


図 3 コンポーネントを組み合わせてシミュレーションを実行する

として作成されるものである。これらのコンポーネントは、BESP 内のモデルコンテナとプレゼンテーションコンテナにそれぞれ配置されるが、その作成・配置を容易にするため、コンテナにはそれぞれのコンポーネントに対応したフレームワークが含まれている。BESP では、モデルや機能がこのようなコンポーネントの協調というかたちで実行されるため、高い拡張性が実現されているのである。

BESP はコンポーネントの組み合わせによってシミュレーションのモデルと環境が構築され、シミュレーションを実行することができる(図 3)。このような構造により、さまざまな研究者が作成したコンポーネントを BESP 上で組み合わせて利用することができる。過去に作られたコンポーネントも、他の研究者によって他の目的で再利用することが可能となるのである。

2.2 BESP におけるコンポーネントの種類

シミュレーション実行のために BESP に配置されるコンポーネントは以下の 2 種類に分類される。これらはそれぞれ独立して作成することが可能であるため、しばしば異なるユーザーによって作成されることがある。

2.2.1 モデルコンポーネント

経済社会モデルのコンポーネントである。このコンポーネントは経済社会モデルにおける要素(例えば個人、企業、財など)で、これらの組み合わせがモデルとして BESP 上でシミュレートされる。Boxed Economy では、経済社会を構成する要素を分類して関係づけた体系を「Boxed Economy 基礎モデル」(Boxed Economy Foundation Model, 基礎モデル)として定義しており、BESP はこの基礎モデルに基づいてモデルを稼働させる^(注2)。

(注2): Boxed Economy 基礎モデルは、現実の経済社会のもつ構造をオブジェクト指向分析によって抽象化したものであり、エージェントベース経済モデルのための基本デザインを提供するものである。基礎モデルはエージェントベースによる経済社会の認識・

モデルコンポーネントはモデルコンポーネントデベロッパーによって作成され配置されるが、そのプログラムとしてのフレームワークは「基礎モデルフレームワーク」(Foundation Model Framework)によって提供されている。

2.2.2 プレゼンテーションコンポーネント

シミュレーション実行中にその状況を出力したり、シミュレーションの内容を操作したりするためのコンポーネントである。例えば、特定のエージェントの状態を追跡して表示するためのウィンドウや、GDP や失業率といったモデル内の経済指標を表示するためのグラフウィンドウなどである。プレゼンテーションコンポーネントは、シミュレーションを行いたいモデルの特徴や目的によって様々なものが作成されると思われる。

プレゼンテーションコンポーネントは、入出力の対象によってさらに以下の 2 種類に分類される。

- GUI コンポーネント

ディスプレイにシミュレーション状況を表示させ、キーボードやマウスによるユーザーからの操作をモデルコンポーネントやコンテナに伝えるためのコンポーネントである。たとえば、エージェントの情報を表示させるためのウィンドウやシミュレーションを制御する制御パネルなどである。

- レポートコンポーネント

シミュレーション実行の経過をファイルに書き出すためのコンポーネントである。

2.3 イベント

BESP は既に述べたように、「Box Operating System」(BOS)、「モデルコンテナ」、「プレゼンテーションコンテナ」、「モデルコンポーネント」、「プレゼンテーションコンポーネント」という 5 つのモジュールで構成されているが、これらのモジュールの通信はイベントによって行われている。イベントによる通信によってモジュールの独立性を高め、モジュールの組み換えを容易にしている。

特にプレゼンテーションコンポーネントでは、BESP およびモデルにおけるさまざまな変更をすべ

分析の参考となるだけでなく、そのモデル要素を基本語句として具体的なモデルの記述を行うことができる。また、基礎モデルに基づくモデルはそのまま BESP 上で動かすことができるため、分析から、記述、シミュレーション実行までの一貫した表現を実現するモデルの枠組みであるといえる。詳細は、論文 [10] [11] およびドキュメント [13] [16] 参照。

てイベントとして受信することができるため、イベントの送信元の仕組みをほとんど考慮せずに、ユーザーインターフェースなどに特化してプレゼンテーションコンポーネントを作成することができる。例えば、値をグラフ表示するためには、モデルの必要な部分からデータを取得してそれを画面に表示する仕組みだけを実装すれば良く、モデルの変更がどのように行われたかということを考える必要がないのである。

BESP 内のモジュール間で送受信しているイベントには以下のような種類がある^(注3)。

- ModelUpdateEvent(モデルコンポーネント更新イベント)

モデルコンポーネントが変更された時にプレゼンテーションコンポーネントに送られるイベントが ModelUpdateEvent である。このイベントはモデルコンポーネントで発火し、最終的にプレゼンテーションコンポーネントに送られる。ModelUpdateEvent には基礎モデル上の要素に対応して 12 種類あるが、必要なイベントのみを受信すればよい。ModelUpdateEvent はそれぞれが対象オブジェクトとメッセージ ID を持ち、変更に関する詳細を保持する。これにより、受信側は更に詳細に適切なイベントを抽出することができる。

- ModelChangeEvent(モデル変更イベント)

ModelChangeEvent は、モデルそのものが変更された時に送られるイベントである。プレゼンテーションコンポーネントの振る舞いをモデルが BESP 上に存在しているかどうかによって変えたい場合や、モデルが変更された場合に表示を変えたい場合などに用いる。

- ModelThreadEvent(モデルスレッドイベント)

ModelThreadEvent は、モデルを動かしているモデルスレッドの状態が変更された時に発火するイベントである。スレッドには停止中・停止処理中・配信中の 3 つの状態があり、この状態が変わる時に ModelThreadEvent が発火される。このイベントはシミュレーションの実行状態を監視して振る舞いを変えたいときなどに用いる。

- PresentationThreadEvent(プレゼンテーションスレッドイベント)

PresentationThreadEvent は、BESP における画面

表示を行っているプレゼンテーションスレッドの状態が変更された時に発火するイベントである。モデルスレッド同様に停止中・停止処理中・配信中の 3 つの状態が変わった時にイベントが発火される。

- ModelExceptionEvent(モデル例外発生イベント)

シミュレーション実行中にモデルの部分で例外が発生した時に発火されるイベントである。プレゼンテーションコンポーネントにおいて例外を監視してエラー処理を行いたい場合などにこのイベントを用いる。

- PresentationExceptionEvent(プレゼンテーション例外発生イベント)

シミュレーション実行中にプレゼンテーションの部分で例外が発生した時に発火されるイベントである。プレゼンテーションコンポーネントなどで例外を監視してエラー処理を行いたい場合などに用いる。

3. コンポーネントのためのフレームワーク

BESP では、コンポーネントデベロッパーがモデルコンポーネントやプレゼンテーションコンポーネントを容易に作成できるように、それぞれの種類のコンポーネントに対するフレームワークが提供されている。このフレームワークには、コンポーネントが BESP 上に配置・実行されるために必要な機能が実装されているので、コンポーネントデベロッパーはこれらの機能を自ら実装する必要がなくなる。これにより、モデルやプレゼンテーションにおける本質的な部分の実装に特化して取り組むことができる。

3.1 基礎モデルフレームワーク

モデルコンポーネントは、Boxed Economy 基礎モデルに基づいて表現されるが、基礎モデルは要素とその関係性の定義のみがされているに過ぎず、プログラムの実装は含まれていない。そのため、シミュレーションを実行するためには基礎モデルの要素すべてをプログラムとして実装する必要があるのであるが、BESP では「基礎モデルフレームワーク」が提供されている^(注4)。これは基礎モデルの要素について BESP で実行できるように実装したものであり、モデルデベロッパーの負担を軽減することができる。

基礎モデルと基礎モデルフレームワーク、モデルコンポーネント、BESP との関係性は図 4 のように

(注3): BESP におけるイベントの仕組みや種類についての詳細は、ドキュメント [15] [16] 参照。

(注4): 基礎モデルフレームワークについての詳細は、ドキュメント [16] 参照。

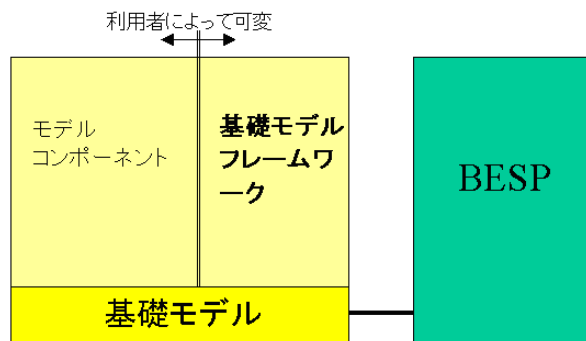


図 4 基礎モデルフレームワークの位置づけ

なる。最も抽象的なベースとして基礎モデルがあり、その土台の上にモデルの実装部分が存在する。基礎モデルは BESP と連携するが、モデルの実装部分は、ユーザーの作成するモデルコンポーネントと基礎モデルフレームワークによって構成されている。モデルコンポーネントと基礎モデルのどちらの部分をもどれだけ作成・利用するのかということはユーザーに任されている。基礎モデルフレームワークで用意されているものだけでも、モデルを作成することが可能であるため、モデルコンポーネント作成者は基礎モデルフレームワークを元にして自分が実装したい部分のみを実装すればよいということになる。

3.2 プレゼンテーションコンポーネントフレームワーク

BESP ではプレゼンテーションコンポーネントの作成に関しても、基礎モデルフレームワークと同様に「プレゼンテーションコンポーネントフレームワーク」が提供されている^(注5)。プレゼンテーションコンポーネントフレームワークは主に Java の Swing フレームワークをベースとするものであり、以下の 4 種類のテンプレートを含んでいる。

- JFramePresentationComponent

独立したウィンドウとして配置されるプレゼンテーションコンポーネントである。BESP とともに提供されているプレゼンテーションコンポーネントでは「MainWindow」および「モデルコンポーザー」がこのテンプレートを利用している。

- JInternalFramePresentationComponent

JFramePresentationComponent 上に配置されるプレゼンテーションコンポーネントである。BESP とともに提供されているプレゼンテーションコンポーネ

(注5): プレゼンテーションコンポーネントフレームワークについての詳細は、ドキュメント [15] 参照。

ントでは「空間表示ウィンドウ」がこのテンプレートを利用している

- JComponentPresentationComponent

前述の 2 種類の Frame 型のプレゼンテーションコンポーネントに配置されるパーツとしてのプレゼンテーションコンポーネントである。BESP とともに提供されているプレゼンテーションコンポーネントでは「空間表示ウィンドウ」に表示される「Agent」や「Relation」などはこのテンプレートを利用している。

- ReportComponent

GUI コンポーネントではないが、プレゼンテーションコンポーネントフレームワークで提供しているレポートコンポーネントのテンプレートである。

これらのテンプレートには、プレゼンテーションコンポーネントがイベントを受信するために必要な、受信者として登録・削除する仕組みがすでに実装されている。その際、登録・削除されるコンポーネントと親子関係にあるコンポーネントも自動的に追加・削除が行われる。例えば、空間表示ウィンドウに 3 つの Agent 表示セルが配置されていた場合、空間表示ウィンドウと Agent 表示セルを親子関係と捉え、空間表示ウィンドウをイベントの受信者から削除すれば、Agent 表示セルも自動的に削除されるということである。

このような仕組みにより、どのような種類のコンポーネントであっても、プレゼンテーションコンポーネントデベロッパーはイベント受信時の処理に特化してプレゼンテーションコンポーネントを作成することができる。

4. BESP からコンポーネントに提供される機能

4.1 Box Operating System による BESP 外部入出力機能の提供

Box Operating System (BOS) には、BESP 外部との入出力を実現する「モデル入出力」、「プラグイン読み込み」、「ログ出力」、「プロパティ管理」機能がある^(注6)。これらの機能は BESP 上のあらゆるモジュールから利用可能である。そのため、コンポーネントデベロッパーはコンポーネントの作成の際に

(注6): BOS についての詳細は、ドキュメント [15] 参照。

この機能を利用すれば、BESP 外部との入出力の具体的な実装をする必要がない。BOS が提供する入出力機能は以下の通りである。

- モデル入出力

箱庭ファイルなど、外部に保存されているモデルの読み込み、書き込みを行う機能である。

- プラグイン読み込み

モデルコンポーネントやプレゼンテーションコンポーネントをクラスパスやプラグインディレクトリから読み込む機能である。

- ログ出力

BESP 実行時のログを出力する機能である。コンポーネントデベロッパーはこの機能を利用することで、コンポーネントのアルゴリズムを検証するためなどに使うログを容易に出力することができる。

- プロパティ管理

設定ファイルを入出力するための機能を提供する。BOS 自身もこの機能を利用してプロパティを設定している。この機能を利用すれば、コンポーネントデベロッパーは設定ファイルを入出力する部分の実装をする必要がなくなる。

4.2 制御コマンド群

BESP の制御は、プレゼンテーションコンテナに含まれているコマンドモジュールを利用して行うことができる。これを利用すればプレゼンテーションコンポーネントから BESP を制御する具体的な部分を実装する必要はない。

現在は、シミュレーション実行コマンド、保存コマンド、制御パネル表示コマンドなど 31 種類のコマンドが実装されている^(注7)。

5. 提供されているプレゼンテーションコンポーネントと支援ツール

BESP には、シミュレーションの作成・実行・分析を支援するためのプレゼンテーションコンポーネントや支援ツールがいくつか提供されている [12]。以下では、それらの概要を紹介する

5.1 制御パネル

制御パネルは、シミュレーションを実行したり停止させたりするための操作を行うためのプレゼ

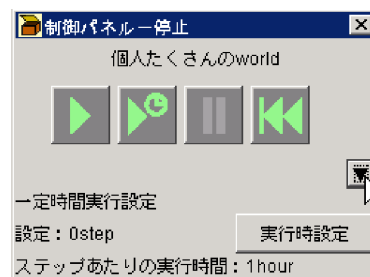


図 5 制御パネル (プレゼンテーションコンポーネント)

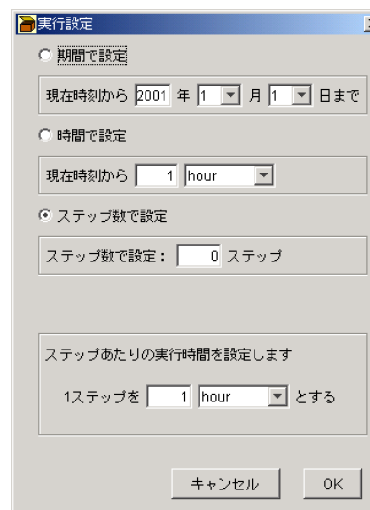


図 6 制御パネル (プレゼンテーションコンポーネント) の設定画面

ンテーションコンポーネントである (図 5)。制御パネルには、「実行」、「一定時間実行」、「停止」、「リセット」のボタンがあり、そのとき BESP に読み込まれているモデルの名前が表示されるようになっていいる。また、指定した時刻が来ると自動的に停止状態になるように実行するという「一定時間実行」の期間指定ができる。「期間による設定」「時間による設定」「ステップ数による設定」が可能となっている (図 6)。

5.2 空間表示ウィンドウ

空間表示ウィンドウは、エージェントの地理的配置および関係が表示され、それがリアルタイムで更新されるプレゼンテーションコンポーネントである (図 7)。時間経過とともにエージェントが移動するモデルの場合には、ユーザーはその過程をリアルタイムで観察することができる。

5.3 時刻表示ウィンドウ

時刻表示ウィンドウは、モデルの現在時刻を表示するためのプレゼンテーションコンポーネントである (図 8)。年・月・日・時間・分・秒という時刻モデ

(注7): 制御コマンドの一覧とその詳細についてはドキュメント [15] 参照。

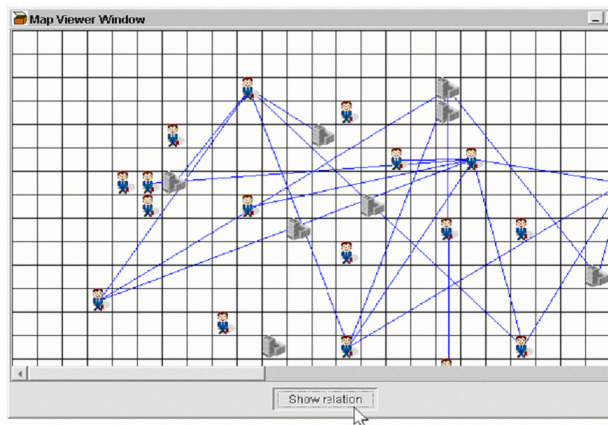


図 7 空間表示ウィンドウ (プレゼンテーションコンポーネント)

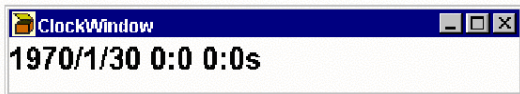


図 8 時刻表示ウィンドウ (プレゼンテーションコンポーネント)

ルだけでなく、整数による抽象的な時刻モデルなどにも対応している。

5.4 データ登録ウィンドウ

データ登録ウィンドウは、シミュレーションの実行結果を選択的に記録するためのプレゼンテーションコンポーネントである (図 9)。分析のためにはすべてのデータを採取しておくといいが、シミュレーションの進行に伴い膨大な量のデータが生成されるため、実際にはコンピュータのメモリの制約などを考慮して必要な情報を選択的に記録するという方法が現実的である。データ登録ウィンドウでは、ユーザーが列条件設定ウィザードなどを通じて入手したいシミュレーションデータを指定し、シミュレーション実行時にデータを格納する場所を作成する。

5.5 グラフ表示ウィンドウ

グラフ表示ウィンドウはデータ登録ウィンドウで登録したデータをグラフとして表示し、シミュレーションの状況をリアルタイムに反映するプレゼンテーションコンポーネントである (図 10, 図 11)。凡例や線色、ラベル、表示単位、目盛り、グリッドなどの設定を行ってグラフを表示させることができる。

5.6 モデルコンポーザー

モデルコンポーザーは、GUIを用いてモデルコンポーネントを組み合わせることによって、シミュレートしたいモデルを構成・設定するためのプレゼン

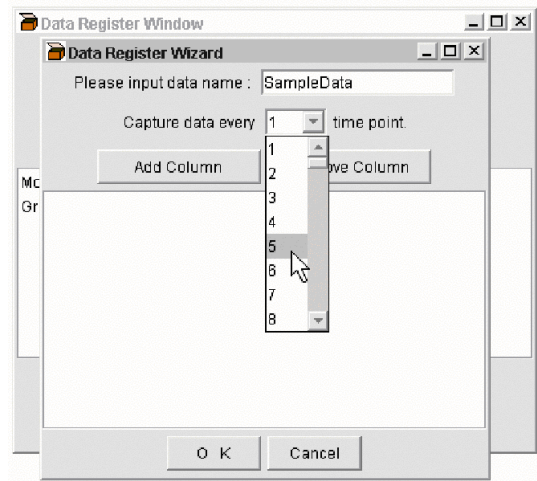


図 9 データ登録ウィンドウ (プレゼンテーションコンポーネント)

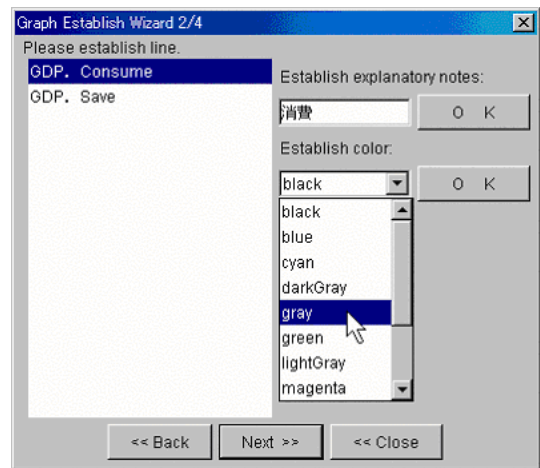


図 10 グラフ表示ウィンドウ (プレゼンテーションコンポーネント) の設定画面

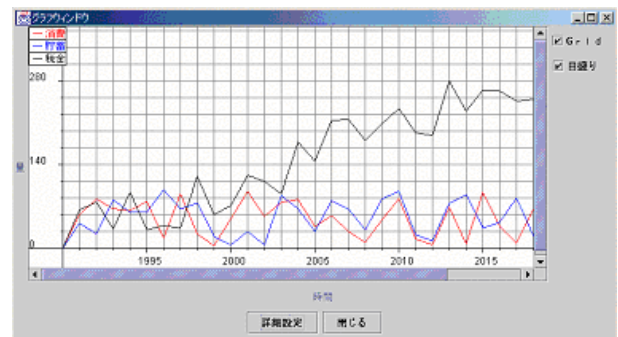


図 11 グラフ表示ウィンドウ (プレゼンテーションコンポーネント)

テーションコンポーネントである (図 12) [13] [17]。プログラミングをまったく行わずに視覚的にモデルを作成・設定することができる

5.7 モデルコンポーネントビルダー

モデルコンポーネントビルダーは、シミュレーション

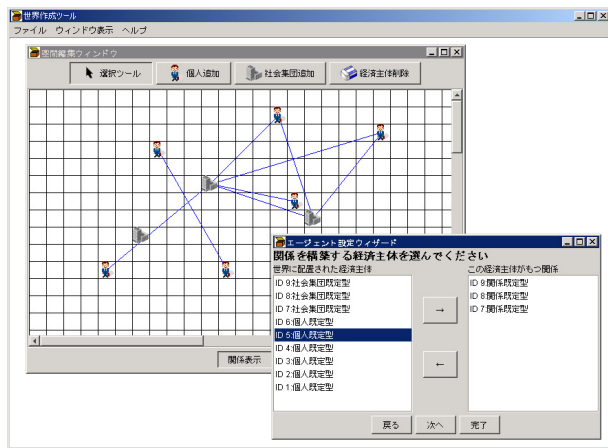


図 12 モデルコンポーザー (プレゼンテーションコンポーネント)

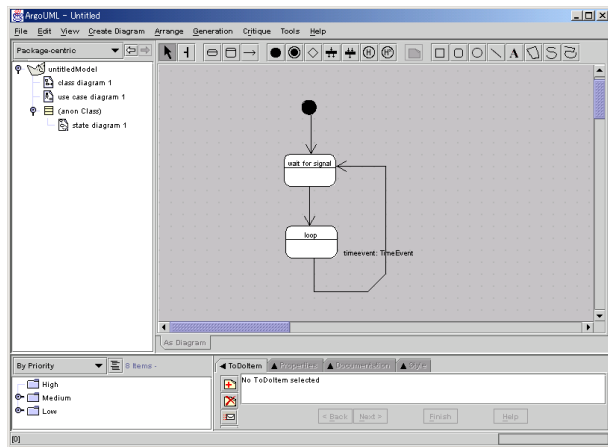


図 13 モデルコンポーネントビルダー (支援ツール)

ンにおけるエージェントの行動を記述し実装するための支援ツールである (図 13) [14]。モデルコンポーネントビルダーは、現段階ではプレゼンテーションコンポーネントではなく、Argo/UML フレームワーク上で動作する Java アプリケーションである [17]。

Boxed Economy 基礎モデルにおける行動モデルコンポーネントは「状態機械」であり、何らかのトリガーとなるイベントを受け取って状態を遷移させながら動くシステムになっている。モデルコンポーネントビルダーでは、ユーザが GUI によってこの行動モデルコンポーネントの「状態チャート図」を記述することで、モデルコンポーネントの Java コードを出力される。これにより、モデルコンポーネント開発者が書かなければならないソースコードの量を減らすことが可能となる。

6. おわりに

Boxed Economy Simulation Platform は、研究者

がモデルやプレゼンテーションなどのシミュレーション実行環境を容易に実現するためのプラットフォームとして開発された。しかし現在のところまだモデルやプレゼンテーションの種類が少ない。このため、今後は BESP を実際に利用する研究者からの意見を吸収してプラットフォームとしての有用性や扱いやすなどを高めていくことが求められる。

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] 和泉 潔, 植田 一博, "人工市場入門", 人工知能学会誌, vol.15, no.6, 2000
- [3] 寺野 隆雄, 倉橋 節也, "エージェントシミュレーションと人工社会・人工経済", 人工知能学会誌, vol.15, no.6, pp.996 - 973, 2000
- [4] 井庭 崇, "エージェントベース経済学試論", 第 5 回進化経済学会, 2001
- [5] 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
- [6] 川村 秀憲, 大内 東, 車谷 浩一, "人工市場モデルのための X-Economy システムとその利用研究", オペレーションズ・リサーチ: 経営の科学, vol.46, no.10, pp. 537 - 543, 2001
- [7] 佐藤 浩, 久保 正男, 福本 力也, 廣岡 康雄, 生天目 章, "人工市場のシステム構造", 人工知能学会誌, vol.15, no.6, pp. 974 - 981, 2000
- [8] T. Iba, Y. Takabe, Y. Chubachi, Y. Takefuji, "Boxed Economy Simulation Platform and Foundation Model", *Workshop of Emergent Complexity of Artificial Markets, 4th International Conference on Computational Intelligence and Multimedia Applications*, 2001
- [9] T. Iba, Y. Chubachi, Y. Matsuzawa, K. Asaka, K. Kaiho, "Resolving the Existing Problems by Boxed Economy Simulation Platform", *Agent-based Approaches in Economic and Social Complex Systems*, IOS Press, will be published in 2002
- [10] 井庭 崇, 中鉢 欣秀, 高部 陽平, 田中 潤一郎, 上橋 賢一, 津屋 隆之介, 北野 里美, 廣兼 賢治, "Boxed Economy の実現に向けて: エージェントベース経済シミュレーションのための基礎モデル", 電子情報通信学会人工知能と知識処理, 情報処理学会知能と複雑系合同研究会, 2001
- [11] T. Iba, Y. Takabe, Y. Chubachi, J. Tanaka, K. Kamihashi, R. Tsuya, S. Kitano, M. Hirokane, Y. Matsuzawa, "Boxed Economy Foundation Model: Toward Simulation Platform for Agent-Based Economic Simulations", *Exploring New Frontiers on Artificial Intelligence*, Springer-Verlag, will be published in 2002
- [12] Boxed Economy Project, BESP1.0 ユーザーズガイド [1] 導入&基本操作, フジタ未来経営研究所, 2001
- [13] Boxed Economy Project, BESP1.0 ユーザーズガイド [2] モデルコンポーザー, フジタ未来経営研究所, 2001
- [14] Boxed Economy Project, BESP1.0 ユーザーズガイド [3] モデルコンポーネントビルダー, フジタ未来経営研究所, 2001
- [15] Boxed Economy Project, BESP1.0 デベロッパーズガイド [1] プラットフォームアーキテクチャ, フジタ未来経営研究所, 2001
- [16] Boxed Economy Project, BESP1.0 デベロッパー

- ズガイド [2] 基礎モデル&基礎モデルフレームワーク
アーキテクチャ, フジタ未来経営研究所, 2001
- [17] Boxed Economy Project, BESP1.0 デベロッパーズ
ズガイド [3] モデルコンポーザー&モデルコンポーネ
ントビルダー アーキテクチャ, フジタ未来経営研究所
, 2001