

## [研究ノート] エージェントシミュレーションのす すめ

著者	谷田 則幸
雑誌名	関西大学経済論集
巻	53
号	4
ページ	381-397
発行年	2004-03-15
その他のタイトル	[Note] An Introduction to Agent Based Simulation
URL	<a href="http://hdl.handle.net/10112/13607">http://hdl.handle.net/10112/13607</a>

## 研究ノート

## エージェントシミュレーションのすすめ

谷 田 則 幸

## 概 要

本稿では、最近社会科学の分野でも注目されているマルチエージェントモデルによるシミュレーションを扱う。社会科学におけるシミュレーションについて歴史やその必要性を概観するとともに、マルチエージェントシミュレーションのプラットフォームとして開発されたいくつかのシミュレータを紹介する。とくに、構造計画研究所により開発されたKK-Multi Agent Simulator (KK-MAS) を取り上げるとともに、シミュレーション事例についても議論する。

キーワード：エージェント、シミュレーション、マルチエージェントシミュレータ、創発、  
マイクロ・マクロ・ループ

経済学文献季報分類番号：02-21

## 1. はじめに

経済学をはじめとする社会科学において、コンピュータシミュレーションが利用されるようになったのは、1960年代であるが、注目を集めるようになったのは1990年代になってからであろう。社会科学におけるコンピュータシミュレーションの歴史については、[Troitzsch97] に詳しく紹介されている。1960年代は、大学の研究にコンピュータが導入された時期であり、その能力も現在のコンピュータと比べると非力なものであった。そのため、当時は、離散事象シミュレーションやシステムダイナミクスによるシミュレーションなどに限られていた [Troitzsch97]。

比較的、成功を収めたと考えられるものに、政策分野でのマイクロシミュレーションがある。マイクロシミュレーションは、確率過程論に理論的な端を発する。マイクロシミュレーションでは、個人や家計などの大規模なマイクロデータのサンプルを用い、遷移確率を用いて時間経過をシミュレートすることで、そのサンプルに変化を与える。それを繰り返すことで、得られた何ステップか後のサンプル集合に統計的解析を施し、得られた統計量を推定値として用いるというものである。

社会シミュレーション研究での大きな変化は、1990年代に現われたマルチエージェントモデルに基づくシミュレーションであろう。マルチエージェントモデルは、自立的主体（エー

エージェント）が複数存在し、それらが相互作用しながら行動する（マイクロレベル）ことで、次元の異なる現象を創発する（マクロレベル）という仕組みをシミュレートするものである。このようなシミュレーションをマルチエージェントシミュレーションというが、これは、コンピュータサイエンスにおける人工知能研究、とくに分散人工知能における研究的蓄積と密接に関係している [Doran97]。また、人工知能研究の一分野である「機械学習 (Machine Learning)」も、エージェントがシミュレーションでの行動の中で、環境や他のエージェントの行動から学習するということの必要性から、深く関係している [Gilbert&Troitzsch99, Sutton&Brato98]。

これらの社会シミュレーションの方法以外にも、マルチレベルシミュレーションモデル、セル・オートマトンモデルなどが存在する [Gilbert&Troitzsch99]。

本稿では、このような社会シミュレーションの発展の流れの中で、マルチエージェントシミュレーションに着目し、それを実現するためのツールとしてのシミュレータを紹介する。とくに、日本語環境での使用が可能で、ある程度の複雑なシミュレーションモデルも構築しうる KK-MAS（構造計画研究所）のインターフェースを示すとともに、シミュレーション適用事例を紹介する。

## 2. マルチエージェントシミュレータ

マルチエージェントシミュレーションを行うプログラムをマルチエージェントシミュレータ（以下では、MAS）という。MASもプログラムであるので、C++やJavaといったプログラミング言語を用いて、ユーザ自身が作成することもできる。しかしながら、MASは単にある入力を得て、処理をし、出力をする、というレベルにはとどまらない。多くのエージェントの相互作用をシミュレートすることをベースとして、そのエージェントが存在する「場」を提供し、エージェントがどのように行動し、どのように変化するか、そしてエージェントの行動により「場」がどのように変化するか、などなど、実現しなければならない要素が多く存在する。しかしながら、そのような要素は、マルチエージェントシミュレーションを行う場合には、共通に必要なものであり、このようなシミュレーションを行うプラットフォームの共有が強く叫ばれてきた。また、シミュレーションを行いたいユーザは、必ずしもコンピュータサイエンティストやコンピュータエンジニアではなく、本稿で扱う社会シミュレーションに限っていえば、大半が社会学者である。そのような意味でも、既存ソフトウェアとしてのMASが必要であるといえよう。本章では、いくつかの代表的なMASを紹介する。

## 2.1 Swarm

Swarm は、米国 SantaFe 研究所で開発された MAS であり、複雑適応系を扱うことができる。MAS の中では歴史が古く、その機能の豊富さから世界中の多くのユーザに支持されている。Swarm は、Unix 上で、X-Window、GNU Objective C を用いたクラスライブラリーとして構成されている。複雑なシミュレーションにも耐える機能を有し、その自由度もきわめて高い。その一方で、高いプログラミングのスキルが要求され、多くの研究者は、専任のプログラマを雇っているときえ言われるほどである。過去の研究で作成されたライブラリが利用できるという大きなメリットはあるが、初心者には敷居が高いシミュレータであるといえよう。より詳しくは、Swarm のオフィシャルホームページを参考にされたい [Swarm04]。

## 2.2 Starlogo

Starlogo は、米国 MIT で開発された MAS である。このシミュレータのベースには、プログラミングの容易な言語「Logo」が用いられており、Swarm と比べるとかなり身近なシミュレータであるといえよう。言い換えれば、あまりプログラミング経験を持たない研究者や学生向けであるといえよう。また、最近、Java 版もリリースされ、Windows でも利用可能になったこと、そして2次元空間での GUI によるモデリングのしやすさは魅力である。Swarm ほどの機能を必要としない場合には、選択肢の一つとなるであろう。より詳しくは、Starlogo のオフィシャルホームページを参考にさせていただきたい [Starlogo04]。

## 2.3 KK-MAS

Swarm や Starlogo は、いずれも海外の研究機関で開発されたものであり、シミュレーションに日本語は使えない。日本語環境でのその大きなデメリットを克服したのが、構造計画研究所 KK-MAS であろう。GUI によるモデリングや設定は容易であり、また、Microsoft 社の VisualBasic に似た文法ルールを採用していることも初心者にも親しみやすいものである。実際、筆者のゼミナールでも今年度から KK-MAS を用いて卒業研究を実施したが、思った以上に面白いシミュレーションが行われていたように思う。また、KK-MAS を紹介した書籍やホームページがあるので参考にされることをおすすめする [KK-MAS04、山影・服部02]。

## 3. KK-MAS の概要

### 3.1 KK-MAS によるプログラミングの環境

KK-MAS を起動し、ファイルメニューから新規作成を選ぶと、「ツリー」ウィンドウ (以

下、ツリー)、「キャンバス－WORLD」ウィンドウ（以下、キャンバス）と「コントロールパネル」ウィンドウ（以下、コントロールパネル）がともに、右下に「選択」、「新規空間」、「新規エージェント」、「新規変数」の4つのボタンが表示される（図1）。ただし、4つのボタンが表示されるのは、キャンバスがアクティブのときのみである。KK-MASによるプログラミングのうち、モデルを設定するフェーズでは、WindowsのGUIを用いて、キャンバスに絵を描くような要領で実現できる。まず、図1の画面中に存在するウィンドウやこの画面から作成されるものについて簡単にまとめておく。

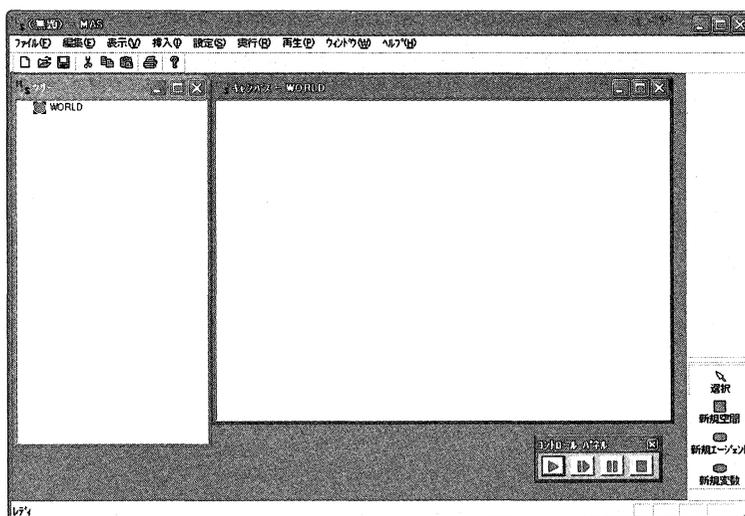


図1：KK-MASによる新規作成の画面

### ① ツリー

新規作成時には、WORLDのみが表示されているが、作成が進むにつれて、シミュレーションに用いる構成物（MASでは、コンポーネントと呼んでいる）が追加表示され、WORLDをルート（根）、各コンポーネントをリーフ（葉）とする樹形図（ツリー）となる。WORLD以下には、その設定に応じて、「空間」、「エージェント」、「変数」が登場する。

### ② キャンバス

新規作成時には、「キャンバス－WORLD」しかないが、「空間」や「エージェント」を作成すれば、それらに対応するキャンバスも現れる。すなわち、WORLD、空間、エージェントは内部にコンポーネントを持つことが出来、それをそれぞれのキャンバス上で作成することになる。そのためには、対象となるキャンバスをアクティブにし、右下にある4つのボタンをクリック後、キャンバス上でもう1度クリックすることにより表示できる。実際には、キャンバスとツリーは同期が取られているため、キャンバス上にコンポーネントを追加するたびにツリーにも同じコンポーネントが表示されることになる。

### ③ WORLD

WORLD も MAS 上ではひとつのエージェントとして存在している。しかし、他のエージェントとは異なり、一つのシミュレーションモデルに対し、ユニークに存在し、かつ他のコンポーネント（空間、エージェント、変数）の最上位に位置する。

### ④ 空間

WORLD の下に作られるが、必ずしもシミュレーションモデルの中に存在する必要はなく、そのシミュレーションモデルによる。逆に、複数の空間を一つの WORLD の中に定義することも出来る。空間の下には、エージェント、変数が定義できる。

### ⑤ エージェント

シミュレーションの主役であり、設定されたルールに従って行動をする。他のコンポーネントに影響を及ぼしたり、影響を受けたりする。エージェントの下には、エージェントと変数が定義できる。

### ⑥ 変数

それぞれのコンポーネントが持つ情報を保持するものであり、数値だけでなく文字も値として持ちうる。

### ⑦ コントロールパネル

新規作成時には、シミュレーションの実行、停止、一時停止、コマ送り（1ステップずつ実行）を制御する4つのボタンだけである。必要に応じて他の機能を実現するボタンなどを追加することも出来、これを用いれば、シミュレーションのパラメータ設定を容易に実現することも出来る。

## 3.2 KK-MAS の設定例

前述のように、KK-MAS におけるモデル設定は、Windows の GUI により、視覚的に行える。本節では、それぞれのコンポーネントを設定する方法を示すことにより、設定できる項目を見る。KK-MAS のようなできばえのシミュレータを用いる場合には、設定できない項目やサイズの限界など、いくつかの制限事項が存在し、それによって行いたいシミュレーションモデルが構築できないことがあり、シミュレータに合わせたモデルを作らなければならない。そのためにも、「出来ること」をあらかじめ知っておくことは重要である。

### 3.2.1 空間

空間を作成するためには、次のいずれかの方法により、「空間プロパティ」ウィンドウを開くことから始まる。また、WORLD 上に複数の空間を設定する場合は、複数回繰り返さな

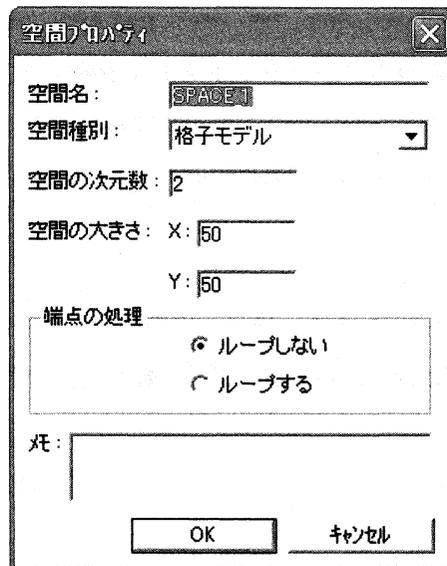


図2：空間プロパティ ウィンドウ

ければならない。

- ① ツリーの WORLD (あるいは、「キャンバスー WORLD」) を右クリックし、ポップアップメニューから「空間の追加」を選択
- ② 右下にある4つのボタンの中から、「空間」をクリックし、続けて、「キャンバスー WORLD」をクリック
- ③ ツリーの WORLD か「キャンバスー WORLD」がアクティブな状態で、「挿入」メニューから、「空間の追加」を選択

まず、「空間名」を入力する。デフォルトは、SPACE\_1になっているが、あとでどのような空間を表しているのかわかるようなネーミングをしておくことが望ましい。日本語(漢字)での入力も可能である。

次に、「空間種別」を設定する。空間種別は、空間の形状を意味し、「格子モデル」か、「六角形モデル」の2種類から選択できる。

「空間の次元数」は、「2」で固定されている。これは、2次元空間、すなわち平面を意味するので、3次元以上の空間は定義できない。

「空間の大きさ」は、縦横それぞれのセルの数によって設定する。空間の大きさは、シミュレーション時間にも影響を及ぼすので、不必要に大きくすることは望ましくない。

「端点の処理」は、エージェントが空間の上下左右の境界線を越えて移動したときに、どのように扱うかを設定できる。「ループする」の場合は、たとえば右端の境界を超えたら左端から現われるが、「ループしない」の場合は、そのエージェントは画面上から消滅する。

すなわち、前者は、「丸い地球」のイメージを表わしているといえよう。

「メモ」は、単なるコメントを書くところで、何を書いてもシミュレーションにはまったく影響しない。ただし、日本語127文字、半角文字256文字までである。

### 3.2.2 エージェント

空間の設定と同様に、エージェントの設定にも複数の方法があるが、類推可能と思われるので、この小節以降は特にことわることなく、一つの方法のみを記述することにする。

エージェントを追加するためには、ツリーの WORLD (あるいは、空間やエージェント) を右クリックし、ポップアップメニューから「エージェントの追加」を選択し、「エージェントプロパティ」ウィンドウを開き、必要な設定を施す。

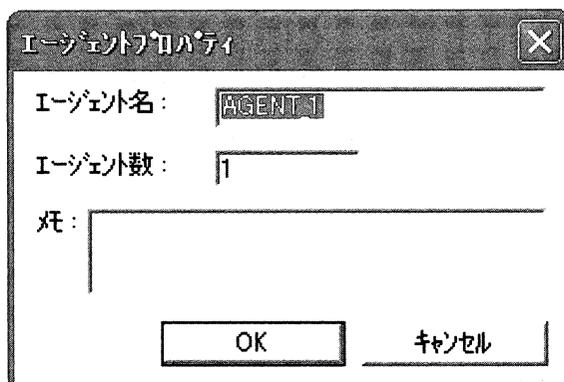


図3：エージェントプロパティ ウィンドウ

まず、「エージェント名」を入力する。デフォルトは、AGENT\_1になっているが、あとでどのようなエージェントを表しているのかわかるようなネーミングをしておくことが望ましい。日本語（漢字）での入力も可能である。

次に、「エージェント数」を設定する。同じタイプのエージェントをいくつ用意するかを意味する。一般に、エージェントシミュレーションでは、エージェント間の相互作用を見ようとしているので、通常は2以上が設定されることが多いが、シミュレーション内容により1ということもあり得る。エージェント数の大小は、シミュレーションに要する時間に大きく影響を及ぼすので、注意が必要である。

複数種類のエージェントが必要な場合には、同様の設定を繰り返すこととなる。

### 3.2.3 変数

変数は、WORLD、空間、あるいはエージェントの下に作成することが出来る。空間上に

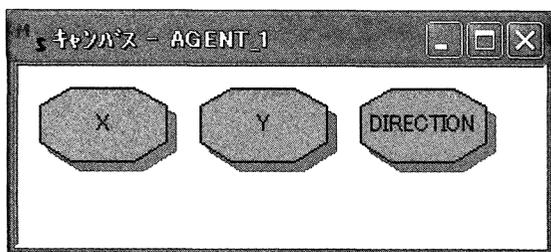


図4：空間上のエージェントのキャンパス

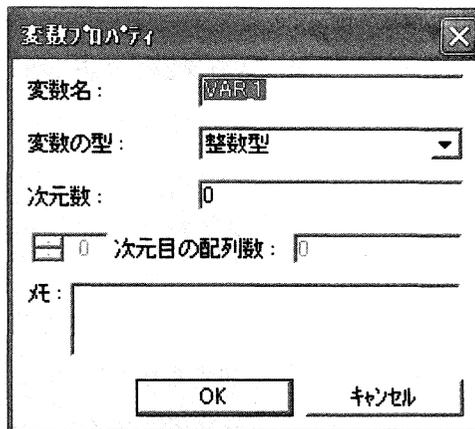


図5：変数プロパティ ウィンドウ

表1：変数の型

型の種類	型の名前	取りうる値
ブール型	Boolean	真のときは True、偽のときは False の値を返す
文字列型	String	文字数は、0~1000
整数型	Integer	-2147483648~2,147,483,647
実数型	Double	(負の場合) $-1.79769313486232 \times 10^{308} \sim -4.94065645841247 \times 10^{-324}$ (正の場合) $4.94.65645841247 \times 10^{-324} \sim -1.79769313486232 \times 10^{308}$
オブジェクト型	Object	エージェントの実態値
コレクション型	Collection	オブジェクト型実数の配列

エージェントを作成した場合には、自動的に3つの変数、「X」、「Y」、「Direction」が作成される(図4)が、それ以外の場合には、ツリーのWORLD(あるいは、空間やエージェント)を右クリックし、ポップアップメニューから「変数の追加」を選択し、「変数プロパティ」ウィンドウ(図5)を開き、必要な設定を施す。

変数プロパティのウィンドウでは、まず変数名を入力する。デフォルトは、VAR\_1になっているが、あとでどのような変数を表しているのかわかるようなネーミングをしておくことが望ましい。日本語(漢字)での入力も可能である。

変数の型では、その変数に入る変数のタイプ(型)を設定しておく。

次元数では、変数のデータ配列を設定する。変数に、1つの値だけを入れる場合には、1つの格納先があればよいが、それを「0次元」の変数といい、デフォルトになっている(図6)。

配列数は、次元数が1以上のときに設定できる。1次元のときは列ベクトル(図7)を、2次元のときは行列(図8)をイメージするとわかりやすい。配列数で指定した数字に1を加えた数の格納先があることに注意されたい。



図 6 : 0次元



図 7 : 1次元、配列数 5



図 8 : 2次元、配列数 (5,2)

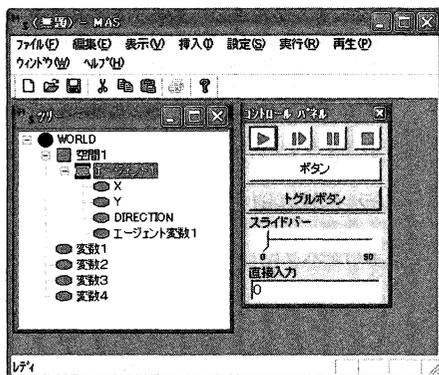


図 9 : ツリーとコントロールパネル

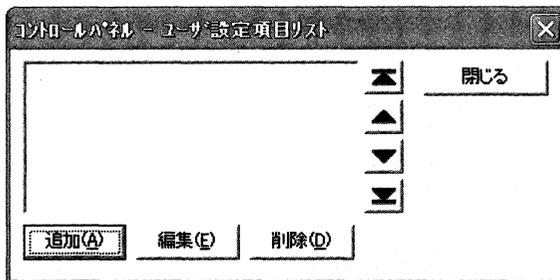


図10 : ユーザ設定項目リスト ウィンドウ

### 3.2.4 コントロールパネル

コントロールパネルでは、デフォルトのシミュレーションの「実行」、「ステップ実行」、「一時停止」、「停止」のほかに、シミュレーション実行中に変数の値を変更するための仕組みを設定することが出来る。ただし、ここでいう変数は、WORLD 直下にあるものに限られる (図9のツリー参照)。

「設定」メニューから「コントロールパネル設定」を選択すると、「コントロールパネル - ユーザ設定項目リスト」ウィンドウが表示される (図10)。

つぎに、「追加」あるいは「編集」ボタンをクリックすると、「ユーザ設定項目」ウィンドウが開くので、必要な設定を加える (図11)。

まず、「コントロール名」を記入するが、ここで入力した名前は、図9のボックスにリストアップされると同時に、コントロールパネルにもボタンの名称などとして現われる。したがって、出来るだけ分かりやすい名称を設定すべきである。また、あまり長い名前をつけると、コントロールパネル上のボタンとして見にくいので、長さにも注意が必要であろう。変数名とは必ずしも同じである必要はないことに注意されたい。

設定対象には、WORLD 直下に定義された変数のリストが表示されているので、設定したい変数をこのリストの中から選択する。

インターフェースでは、コントロールパネル上での出現形状を指定する。「ボタン」は、クリックしたときに「ON」の状態になり、任意の値が代入できる。「トグルボタン」は、

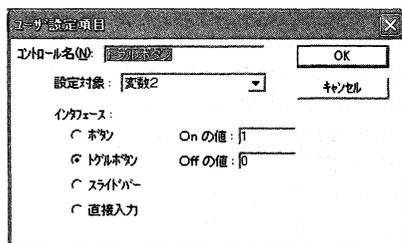


図11：ユーザ設定項目

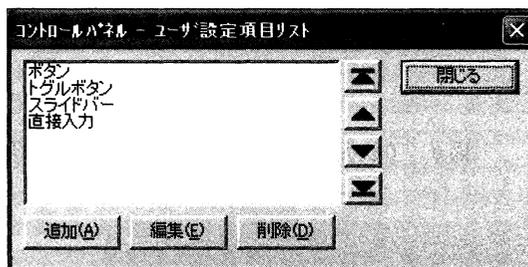


図12：ユーザ設定項目リスト ウィンドウ

「ボタン」と同じ形状をしているが、「ON」、「OFF」の2状態を持ち、それぞれに任意の値を代入することが出来る。「スライドバー」は、左右にスライドさせることにより、あらかじめ設定した範囲の任意の値を取得することが出来る。「直接入力」は、変数への代入値をシミュレーション中に明示的に変更(代入)出来る。図12のように設定したとき、図9のコントロールパネルのようなレイアウトとなる。

### 3.2.5 ルールの記述

3.2.1から3.2.4において、シミュレーションモデルのおおよその枠組みが完成できたことになる。次に実際にシミュレーションを実行するために、ルールを実装することになる。

#### 3.2.5.1 エージェントのルール

エージェントのルールは、同じ種類のエージェントの場合は同じ内容のルールを持つことになる。一方、エージェントの種類が違う場合には、一般的には異なるルールを持つ。エージェントが同じルールで行動していたとしても、そのエージェントの位置や状態などは異なるため、同一時刻において同じ行動をとっていないことに注意されたい。

エージェントルールは、2種類の部分からなる。ひとつは、シミュレーションでエージェントが生成されたときに一度だけ実行される初期ルール `Agt_Init` である。エージェントごとの初期値の設定などに使われるのが一般的である。もうひとつの部分は、`Agt_Step` といい、エージェントルールの本質的な部分である。ツリーの適当なエージェントを右クリックし、ポップアップメニューから「ルールエディタ」を選択すると、「ルールエディタ」ウィンドウ(図13)が開くので、このルールエディタを用いてエージェントのルールを記述する。

図14に、MAS ユーザマニュアルより引用した簡単なエージェントルールの例を示す。Microsoft社のVisual Basicになじみのある読者には、きわめて親しみやすい文法になっていることが分かるだろう。紙数の都合から、ルールの記述の詳細については、MAS ユーザ

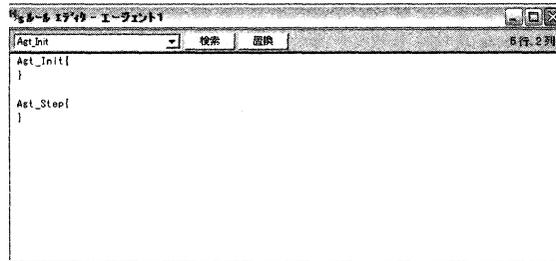


図13：ルールエディタ

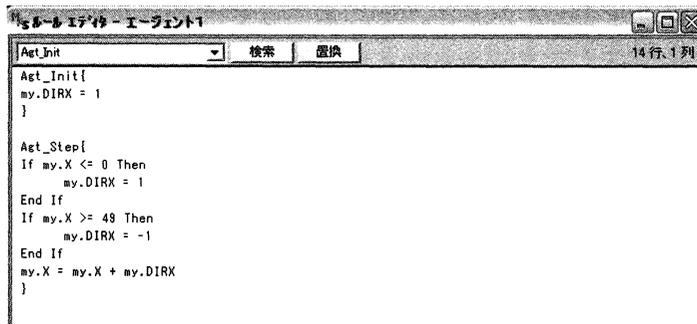


図14：ルールエディタを用いて記述されたルールの例

マニュアルの「第3章 エージェントルール文法」に委ねることとする [KK-MAS02]。

### 3.2.5.2 WORLD のルール

WORLD もひとつのエージェントであるので、そのルールもエージェントと同様に Agt\_Init と Agt\_Step の2種類の部分からなる。WORLD のルールは、その下にあるエージェントを制御することができるので、たとえばエージェント全体に対する一括処理を実現するために用いることができる。それ以外については、WORLD 自身についてのルールを記述する。

### 3.2.6 初期値設定とデフォルト値設定

ツリーの適当なエージェントを右クリックし、ポップアップメニューから「初期値設定」を選択すると、「初期値設定」ウィンドウ (図15) が開く。エージェントの新規作成時に設定したエージェント数の分だけ行が表れるので、それぞれのエージェントごとに、変数値を設定できる。図15において、左端の列中の0から9はエージェント ID (エージェントの番号) を表わしている。エージェントではなく、変数に対して初期値設定する場合には、ツリーの適当な変数を右クリックし、ポップアップメニューから「初期値設定」を選択する

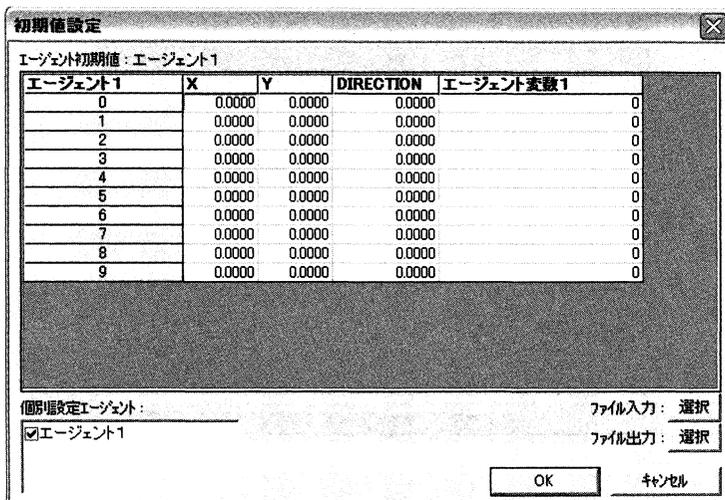


図15：初期値設定 ウィンドウ

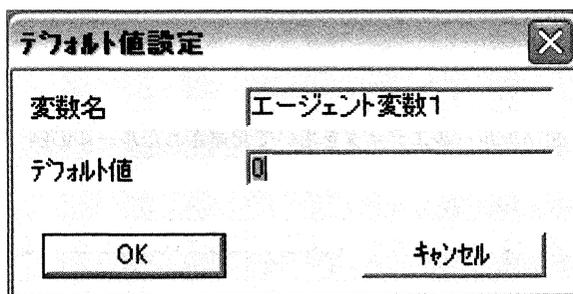


図16：デフォルト値設定 ウィンドウ

と、「初期値設定」ウィンドウに変数の部分だけが表示される。

ツリーの適当な変数をクリックし、「設定」メニューから「デフォルト値設定」を選択すると、「デフォルト値設定」ウィンドウが開くので、デフォルト値を設定する。一般的に、変数に対しては、前述の初期値設定を用いるが、シミュレーション実行中にエージェントが追加された場合には、ここで設定されたデフォルト値が設定されることになる。

### 3.2.7 出力設定の定義

シミュレーションでは、単に実行を開始し、ある一定の時間の経過後に終了し、その結果のみを検証するというものではない。とくに、エージェントシミュレーションの場合は、ミクロレベルで、エージェントが時々刻々どのように移動・行動するか、他のエージェントとどのように影響を及ぼしあうかを観察するとともに、マクロレベルでその WORLD 全体がどのように変化し、どのような結果に収束するか（あるいは、収束しないか）など、ミクロ・マクロの両方のレベルで起こる多くのことに興味がある。そのためには、どのような内

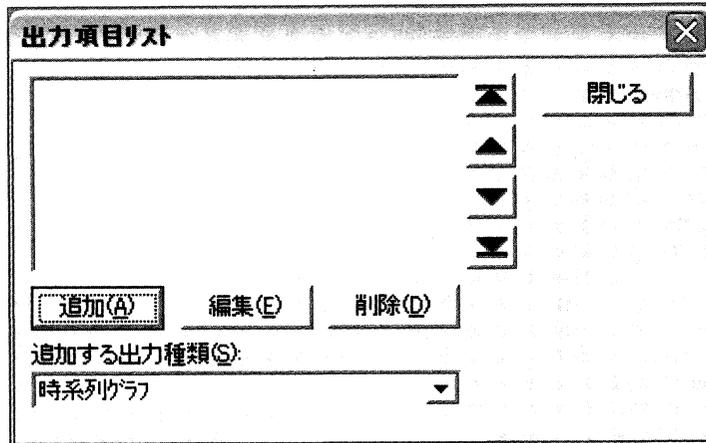


図17：出力項目リスト ウィンドウ

容のものが、どのような表現方法で得られるか、すなわちシミュレータとしてどのような結果出力が可能であるかは、シミュレータ選択の大きな要素となる。また、既成のものである以上、自分で作成したり、オーダーメイドによるプログラムのように、思った出力が用意されていない場合もあるだろうが、そのようなときに、思った出力を作成してくれる（すなわち、そのためのデータを提供してくれる）柔軟性が要求される。

MAS の場合は、後発のシミュレータであるということもあり、必要だと思われる出力がおおむね備わっているといえよう。MAS での出力設定は、「設定」メニューから「出力設定」を選択したときに現われる「出力項目リスト」ダイアログから行える（図17）。

図17の「追加する出力種類」のリストには、以下のような出力の選択が可能である。いずれかを選択し、「追加」ボタンをクリックすることによりそれぞれの設定ウィンドウが開く。

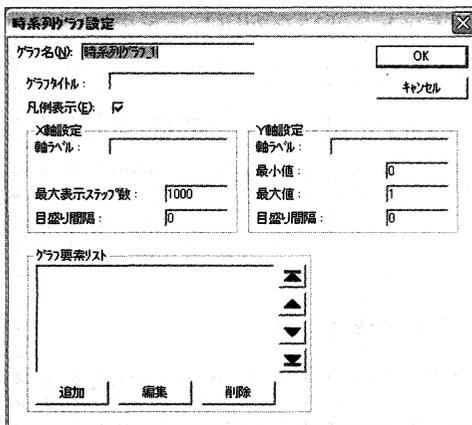


図18：時系列グラフ設定

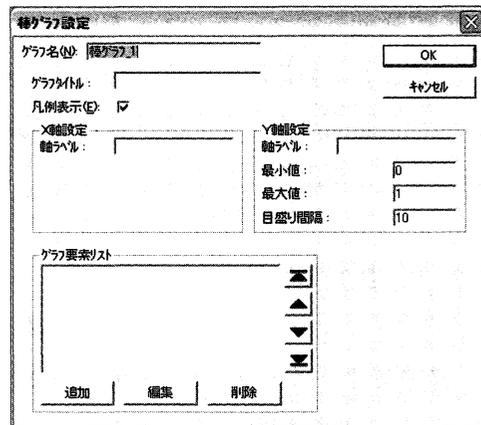


図19：棒グラフ設定

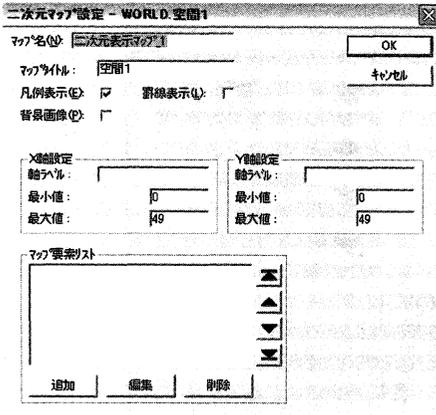


図20：二次元マップ設定

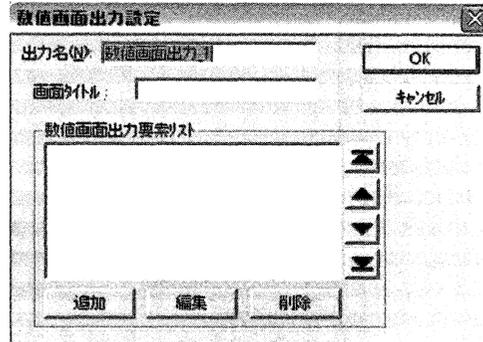


図21：数値画面出力

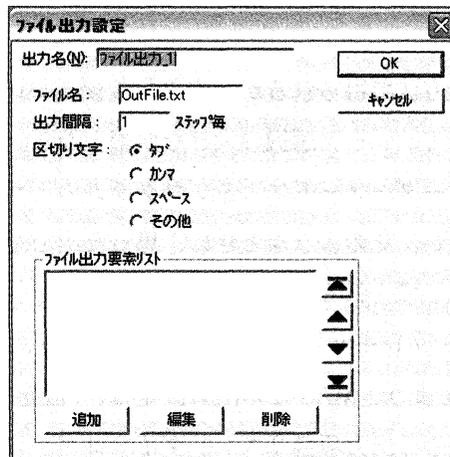


図22：ファイル出力設定

- ① 時系列グラフ：変数の値が、時間軸を持つ折れ線グラフとして出力される (図18)
- ② 棒グラフ：コンポーネントが平面上の棒グラフとして出力される (図19)
- ③ 二次元表示マップ：マップ上に、エージェント、変数を配置し、その時点でのエージェントの位置や変数の値の変化を出力する (図20)
- ④ 数値画面出力：コンポーネントの数値を画面上に出力する (図21)
- ⑤ ファイル出力：シミュレーションのログをファイルに出力する (図22)

### 3.2.8 実行環境設定

シミュレーションの実行の際の制御情報の設定ができる。設定項目には、「シミュレーション終了条件 (ステップ数、時間など)」、「ログファイルのファイル名の設定」、「エージェントルール実行順序」、「シミュレーションを連続実行させるときのパラメータとしての変数初期値の変化方法」、「レポート出力の設定 (ファイル名や区切り文字の指定など)」が

ある。いずれも、効率よくシミュレーションを行うために必要なものである。

#### 4. KK - MAS を用いたシミュレーション事例

本章では、筆者のゼミナールに所属する学生が KK-MAS を用いて行ったシミュレーションを紹介する。本来であれば、KK-MAS の開発元である、構造計画研究所主催のコンペティションでの優秀賞などについて触れるべきところかもしれないが、学部生レベルであっても、比較的興味深い結果を導出できるということの例として紹介することとした。

このシミュレーションは 3 種生物間の生存競争と系の安定性について考察したものである。

2 種生物間の捕食・被捕食関係を微分方程式でモデル化した Lotka-Volterra 方程式は、数理生物学にとどまらず、Goodwin の景気循環モデルと関連して、経済学においても馴染み深いモデルである。[小林04] では、Hungry Lotka-Volterra 方程式にヒントを得て、多種間の捕食関係のシミュレーションを行った。オリジナルのオオカミ-ヒツジの捕食関係に加え、オオカミキラー-オオカミという捕食関係を導入した。この系では、「オオカミ増→ヒツジ減、オオカミキラー増→オオカミ減→ヒツジ減、オオカミキラー減→オオカミ増」とい

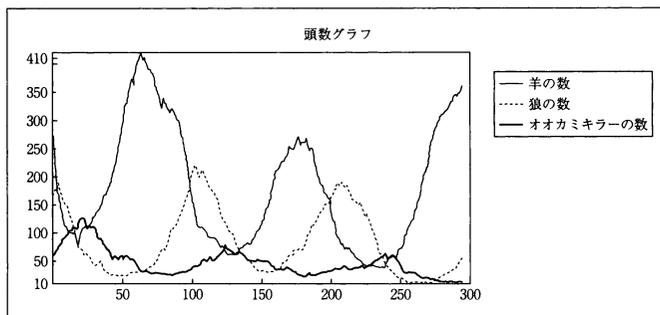


図23：3 種間のシミュレーション

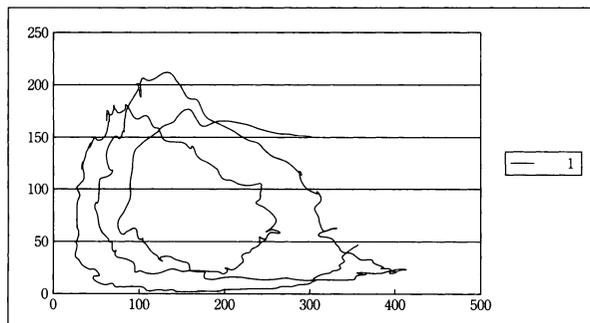


図24：オオカミ (Y) とヒツジ (X) の頭数の変化

う構図が予想されるが、意外にもオオカミキラーが早々と姿を消し（図23）、残ったオオカミとヒツジが比較的安定した周期の増減を繰り返すことを、シミュレーションの出力結果から確認した（図24）。

このような結果の原因を次のように考察している。ヒツジはオオカミに食べられない限り、牧草を食べ、複利的に増加するため、オオカミは増減を繰り返しながらも、比較的安定してヒツジを捕食することができた。しかし、オオカミキラーはオオカミの増減が激しいため、安定して捕食することができず、結果として飢えて絶滅してしまった、というものである。

続いて、5種間のシミュレーションも実施されたが、結果は芳しくはなかったようである。

## 5. まとめ

本稿では、社会シミュレーションの分野で活発な研究がなされているマルチエージェントシミュレーションについての概要述べるとともに、代表的なシミュレータについての紹介を行った。とくに、構造計画研究所のKK-MASは、日本語環境でも使え、その利用も比較的容易であり、研究者のみならず、学部学生にも使ってもらいたいという思いから、その仕様の解説を行い、卒業論文でのシミュレーション事例を紹介した。

マルチエージェントシミュレーションは、優れたシミュレータの登場により、多くの研究者がシミュレーションを行っている。しかし、他のシミュレーションと同様、その初期値によって結果が大きく異なるなどの問題は抱えている。また、各エージェントの行動を左右する学習能力についての理論的研究も平行して行われなければならない。しかし、エージェントシミュレーションには大きな可能性を秘めていることも事実であり、今後の発展が大いに期待される。

## 謝 辞

本稿では平成15年度関西大学学部共同研究（財政・金融政策の有効性に関する理論的・実証的研究）の成果報告の一部である。また、株式会社構造計画研究所より、KK-MASを貸与していただきました。ここに記して、謝意を表したい。

## 参考文献

- [Doran97] J.E. Doran, Foreknowledge in artificial societies. In R. Conte, R. Hegselmann and P. Terna (eds), *Simulating Social Phenomena, Lecture Notes in Economics and Mathematical Systems*, Vol.456, pp.457-470, Springer-Verlag, Berlin, 1997.
- [Gilbert&Troitzsch99] N. Gilbert & K.G. Troitzsch, "Simulation for the Social Scientist", Open University Press, 1999.
- [KK-MAS02] 構造計画研究所, 「Multi Agent Simulator 日本語版ユーザーマニュアル」, 2002.

- [KK-MAS04] 構造計画研究所 MAS コミュニティホームページ (URL <http://www.kke.co.jp/mas/>, 2004年2月1日現在).
- [Starlogo04] MIT Starlogo Home Page (URL <http://education.mit.edu/starlogo/>, 2004年2月1日現在).
- [Sutton&Barto98] R.S. Sutton & A.G. Barto, "Reinforcement Learning: An Introduction", MIT Press, 1998.
- [Swarm04] The Swarm Development Group Home Page (URL <http://www.swarm.org/>, 2004年2月1日現在).
- [Troitzsch97] K.G. Troitzsch, "Social simulation - origins, prospects, purposes". In R.Conte, R. Hegselmann and P. Terna (eds), *Simulating Social Phenomena, Lecture Notes in Economics and Mathematical Systems*, Vol.456, pp.41-54, Springer-Verlag, Berlin, 1997.
- [小林04] 小林亮, 「Multi Agent Simulator による多種生物間の生存競争と系の安定性に関する考察」, 2003年度関西大学経済学部谷田ゼミナール卒業論文.
- [山影・服部02] 山影進・服部正太編著, 「コンピュータのなかの人工社会」, 共立出版, 2002.