新型シミュレータ開発プロジェクト　　ワーキングペーパー・シリーズ

Project for New-Type Simulators　　Working Paper Series

Working Paper No.6

ABSの概要と類似シミュレータとの比較

服部正太*・玉田正樹*
辺見和晃*・桑原敬幸*

２０００年３月

（*株式会社 構造計画研究所）

「シミュレータ開発プロジェクト」は、研究・教育を目的としたマルチエージェント型や繰り返しゲーム型のシミュレータやソフトの開発を目指しています。このワーキングペーパー・シリーズは、プロジェクトの活動・成果の一端を公開するものです。
“Project for New-Type Simulators” is developing a multi-agent based simulator and a simulator of iterated cognitive games, among others, for scientific and/or educational purposes. This working paper series aims at disseminating interim but interesting outcomes of this on-going project.

科学研究費補助金・基盤研究（B）（１）展開（10552001）
東京大学大学院総合文化研究科国際社会科学専攻　　山影 進　研究室
お問い合わせ(E-mail)：tasuke@waka.c.u-tokyo.ac.jp
ホームページ(URL)：http://hachibei.c.u-tokyo.ac.jp/users/yamakage/ntsp1.html

1. はじめに

本稿では、構造計画研究所において開発された Agent Based Simulator（以下、ABS）の機能を技術的側面から記述し、さらに既存のアプリケーションとの比較を行う。
ABS とは一般に「エージェント・ベースド・モデル」「マルチエージェント・モデル」と呼ばれているモデルをコンピュータ上に構築し、シミュレーションを行うことを目的とする Windows アプリケーションである。ABS は日本語版 Windows 環境で動作する。対象としては、主に社会科学分野を想定し、ユーザは Visual Basic 程度のプログラミングの知識を有することを前提としている。
アプリケーションの動作環境は以下のとおりである。

- CPU　：　Pentium 233MHz 以上
- HDD　：　50MB の空き容量
- メモリ　：　64MB 以上
- OS　：　Windows 95/98/NT4.0/2000

2. ABS の特徴

ABS はその名の通りエージェント・ベースのシミュレーションを行うツールであり、その動作手順は以下の４フェーズに分けることができる。

- シミュレーション世界の構築
  シミュレーションを行う世界の構成物を決定する。ABS ではシミュレーションを行う世界も「WORLD」というエージェントとして考え、「WORLD」を起点としたエージェントの階層構造によりシミュレーションを行う世界を表現する。エージェントの内部（下層）に持つ構成物としては、エージェント、変数、空間、の３種類が存在し、これらが各エージェントの属性となる。グラフィカル・ユーザ・インタフェース（以下、GUI）により「WORLD」を含むすべてのエージェントの属性を決定することで、シミュレーション世界が決定される。
- エージェントルールの記述
  すべてのエージェントはその動作ルール、すなわちエージェントルールを持ち、ABS ではそれらをルール記述用テキストエディタにより記述する。エージェントルールの文法は、Visual Basic の言語体系に類似した文法が採用されている。
- シミュレーション設定
  シミュレーション実行における以下の設定を行う。
  - 初期値設定
    エージェント属性として定義されている変数の初期値を設定する。
  - 実行環境設定
    シミュレーション終了条件、乱数シード値など、シミュレーション動作に関する設定する。
  - 出力設定
    シミュレーション実行時に出力されるグラフ、ファイルなど出力情報を設定する。
  - コントロールパネル設定
    シミュレーションの動作をさせるためのインタフェース（コントロールパネル）に、「WORLD」の属性として定義されている変数値を操作するためのインタフェースを追加する。
- シミュレーション実行および再生
  コントロールパネルへのユーザ操作によりシミュレーションの実行を操作する。このとき設定された出力が行われる。実行後、シミュレーションのログが保存されていれば、そのログより各エージェントの属性として定義されている変数の値を再現、出力することができる。

次項より各フェーズの具体的な機能と特徴を挙げる。

3. ABS の機能

ABS は「シミュレーション世界の構築」「エージェントルールの記述」「シミュレーション設定」「シミュレーション実行および再生」の４つのフェーズにより構成される。

- シミュレーション世界の構築

キャンバス編集機能

・ 指定したエージェント、空間をウィンドウ表示する。また、その内部に存在するエージェント、変数、空間を表示する。
・ 指定したエージェント、空間について、新規にエージェント、変数、空間を定義する。
・ 指定したエージェント、変数、空間を削除する。
・ 指定したエージェント、変数、空間をコピーする。
・ 指定したエージェント、変数、空間をペーストする。
・ 指定したエージェント、変数、空間をカットする。
・ 指定したエージェント、変数、空間の属性を表示する。
・ 指定したエージェント、変数、空間の属性を変更する。
・ 指定したエージェント、変数、空間の表示位置を移動する。

ツリー編集機能

・ 現在編集しているモデルを階層構造で表示する。
・ 指定したエージェント、空間について、階層下の表示／非表示を指定する。
・ 指定したエージェント、空間について、キャンバスを表示する。
・ 指定したエージェントについて、ルールエディタを表示する。
・ 指定したエージェント、変数、空間を削除する。
・ 指定したエージェント、変数、空間をコピーする。
・ 指定したエージェント、変数、空間をペーストする。
・ 指定したエージェント、変数、空間をカットする。
・ 指定したエージェント、変数、空間の属性を表示する。
・ 指定したエージェント、変数、空間の属性を変更する。

ツールボックス編集機能

・ 登録されてあるツールボックスを表示する。
・ 指定したエージェント、変数、空間を新規に登録する。
・ 指定したエージェント、変数、空間を登録から削除する。
・ 指定したエージェント、変数、空間を新規に登録する。
・ 指定したエージェント、変数、空間の属性を表示する。
・ 指定したエージェント、変数、空間の属性を変更する。
・ 指定したエージェント、変数、空間をコピーする。

エージェント属性編集機能

・ エージェント名、エージェント数、メモを表示する。
・ エージェント名、エージェント数、メモを変更する。

変数属性編集機能

・ 変数名、変数の型、次元数、メモを表示する。
・ 変数名、変数の型、次元数、メモを変更する。

空間属性編集機能
・ 空間名、空間種別、空間の大きさ、端点の処理、メモを表示する。
・ 空間名、空間種別、空間の大きさ、端点の処理、メモを変更する。

- エージェントルールの記述

ルール編集機能
・ 指定したエージェントのルールを表示する。
・ 指定したエージェントのルールを変更する。
・ ルール編集中、指定した関数へジャンプする。
・ ルール編集中、文字列を検索する。
・ ルール編集中、文字列を置換する。
・ ルール編集中、行をインデントする。
・ ルール編集中、エージェント、空間を記述したとき、次に指定されるべきエージェント、変数、空間を表示する。
・ ルール編集中、カーソル位置の行数、列数を表示する。

- シミュレーション設定

初期値設定機能
・ 指定した空間について、マップによりエージェント座標、変数の値を表示する。
・ 指定した空間について、マップによりエージェント座標を変更し、変数の値を変更する。
・ 指定したエージェント、変数について、表形式により値を表示する。
・ 指定したエージェント、変数について、表形式により値を変更する。

実行環境設定機能
・ シミュレーションの終了条件を指定する。
・ ログファイルの出力形式を指定する。
・ エージェントルールの実行順序を指定する。
・ シミュレーションの連続実行条件を指定する。
・ レポート出力の形式を指定する。

出力設定機能
・ 登録されてある出力設定を表示する。
・ 登録されてある出力設定を削除する。
・ 時系列グラフを設定する。
・ 棒グラフを設定する。
・ 二次元表示マップを設定する。
・ 数値画面出力を設定する。
・ ファイル出力を設定する。

コントロールパネル設定機能
・ 登録されてあるコントロールパネル設定を表示する。
・ 登録されてあるコントロールパネル設定を削除する。

・ ボタンを設定する。
・ トグルボタンを設定する。
・ スライドバーを設定する。
・ 直接入力（テキストボックス）を設定する。

- シミュレーション実行および再生

シミュレーション実行制御機能

・ 指定したモデルのシミュレーションを設定した条件で実行する。
・ ログファイルを保存する。

出力実行機能

・ 登録されてある時系列グラフを出力する。
・ 登録されてある棒グラフを出力する。
・ 登録されてある二次元表示マップを出力する。
・ 登録されてある数値画面出力を出力する。
・ 登録されてあるファイル出力を出力する。

コントロールパネル実行機能

・ 実行、１ターン実行、一時停止、停止の各動作を制御する。
・ 登録されてあるボタンを出力し、シミュレーションを制御する。
・ 登録されてあるトグルボタンを出力し、シミュレーションを制御する。
・ 登録されてあるスライドバーを出力し、シミュレーションを制御する。
・ 登録されてある直接入力（テキストボックス）を出力し、シミュレーションを制御する。

シミュレーション再生機能

・ 指定したログを含むジョブファイルの実行し､格納されているシミュレーションを再生する。

4. 既存ツールとの比較

4.1. 既存ツール

本章では ABS 同様にエージェント・ベースのシミュレーションを行う既存の道具に関して考察し、ABS との比較を行う。
現在エージェント・ベースのシミュレーションを行うアプリケーション、プラットフォームとして比較的良く知られているものは以下のとおりである。

- Swarm
- StarLogo

次項よりこの２つの道具の概要および ABS との比較を記述する。

4.2. Swarm

米国 Santa Fe 研究所にて開発されたエージェント・ベースのシミュレーションを行うプログラミング環境であり、（基本的には）UNIX システム上で動作するプログラミング言語である Objective-C 言語のライブラリという形態で提供されている。Objective-C 言語とは C 言語をオブジェクト指向に発展されたものであり、属性（インスタンス変数）とルール（メソッド）を持つという点で､エージェントをオブジェクト指向プログラミングにおけるオブジェクトとして扱うことは適当である。Swarm ではそれらを補助する機能及びシミュレーションに必要なユーティリティが提供されている。
Swarm のライブラリは主に以下のような構成となっている。

- Defobj
  Swarm（シミュレーション）に適したプログラミングを行うための Objective-C 言語（最も基本的なオブジェクト）の拡張
- collections
  Swarm のオブジェクトの集まりを扱う
- activity
  各エージェントのシミュレーションにおける動作を扱う
- objectbase
  エージェントの基本となるオブジェクト
- space
  空間的な（主に二次元空間などの）シミュレーションを簡易に行う
- random
  乱数生成、多様な分布の生成を行う
- simtools
  シミュレーションのための機能を提供する
- simtoolsgui
  シミュレーションのための GUI の機能を提供する
- gui
  シミュレーションで用いられる GUI のライブラリ
- analysis
  シミュレーションデータの分析を行う

シミュレーションの構築に上記ライブラリを使用することにより、プログラミングが容易になり、また動的な分析を含むシミュレーション等も可能になる。

主な特徴

- オブジェクト指向プログラミング言語（Objective-C）のライブラリ
- （基本的に）UNIX 環境で動作
- あらゆるシミュレーションに有効

4.3. StarLogo

米国 MIT において開発された分散型システム（交通渋滞やアリの巣の形成、鳥の群れ等）のシミュレーションを行うアプリケーションである。以前はプラットフォームとして Macintosh のみを対象としていたが、現在では Java アプリケーション版が開発され、UNIX、Windows においても動作する。
StarLogo の世界には以下の３つの構成物が存在する。

- Turtle（カメ、複数存在する）
- Patches（２次元平面、カメが動作する地面・環境）
- Observer（シミュレーション管理者）

複数の Turtle（エージェント）が同時に Patches（二次元空間）上を決められた条件で移動し、Patches の状態を変化させることによりシミュレーションが行われる、すなわち、二次元空間上のマルチエージェントシミュレーションである。
実際には、Turtle はシミュレーション中に行われる動作手順を持ち、また、シミュレーション全体を管理する Observer も動作手順を持つ。これら２つの動作手順によりシミュレーションが動作する。動作手順は（？で有名な？）Logo プログラミング言語を拡張した簡易言語で記述され、特徴的なものとして以下のような機能を持つ。

- Turtle の状態の定義・取得
- Turtle の生成・消滅
- Turtle の方向転換・移動
- Patches の状態の定義・取得
- 近傍状態取得
- Patches の拡散

また StarLogo では、各動作手順やシミュレーション設定値に対応する GUI がマウス操作により簡単に作成でき、そのインタフェースを用いてシミュレーションの操作を行う。シミュレーション結果は StarLogo の世界（Turtle、Patches）を表す二次元マップや設定された値が簡単なグラフに出力される。

主な特徴

- マルチプラットフォーム
- 簡易言語（Logo プログラミング言語の拡張）
- 二次元空間上の限定的なシミュレーション
- シミュレーション操作 GUI 作成

4.4. 比較

エージェント・ベースのシミュレーションを行う既存の道具を、以下の視点により比較を行う。

道具の種類

- ABS
- Swarm

- StarLogo
- 既存の開発環境（VB、VC++ など）

場面

- インストール（環境構築）
- シミュレーション世界の構築
- エージェントルールの記述
- シミュレーションの設定
- シミュレーションの実行環境（操作・出力）
- その他

比較項目

- 必要知識
- 難易度　　（難：×　→　易：◎）
- 作業時間　　（多：×　→　少：◎）
- 自由度　　（無：×　→　有：◎）
- 機能　　（少：×　→　多：◎）
- アピアランス　　（悪：×　→　良：◎）
- 実行速度
- その他

＊比較項目に関しては、フェーズにより取捨選択を行う。

インストール（環境構築）

インストールを行い、サンプル等を用い動作を確認する。

	ABS	Swarm	StarLogo	既存の開発環境
概要	・インストーラによるインストール ・サンプルあり	・Windows 環境 Windows インストーラによるインストール サンプルのコンパイル・実行 ・UNIX 環境 様々な環境のインストール Swarm のコンパイル サンプルのコンパイル・実行	・インストーラによるインストール ・サンプルあり	・インストーラによるインストール ・サンプルなし
必要知識	特になし	UNIX に関する一般的な知識	特になし	動作確認を行うための知識
難易度	◎	×	◎	○
作業時間	◎	×	◎	○

シミュレーション世界の構築

シミュレーションを行う世界（構成物、属性）の設定を行う。

	ABS	Swarm	StarLogo	既存の開発環境
概要	・エクスプローラ風のマウスによるモデル、属性の設定 ・エージェント、空間、変数の階層構造 ・ABS においてモデルを試行錯誤できる 頭の中や紙の上でのモデリングではなく ABS 上でのモデル設計	・設計されているモデルをプログラミング（オブジェクトの作成） ・基本的なエージェント､空間などのライブラリは用意されている	・既定 二次元上のカメと二次元空間 ・カメの属性は手順（プログラム）内で記述	・すでに設計されているモデルをプログラミング
必要知識	ABS 操作の取得	Objective-C の理解 Swarm ライブラリについての把握	StarLogo 操作の取得 手順の文法の把握	プログラミング言語の習得
難易度	◎	△	○	×
作業時間	◎	△	◎	×
自由度	○	◎	×	◎

エージェントルールの記述

各エージェント、及びシミュレーション全体の動作の設定を行う。

	ABS	Swarm	StarLogo	既存の開発環境
概要	・ABS 内の専用エディタにて記述 ・簡単なプログラム文法 VB 風 自由なアルゴリズム ・シミュレーションに特化した関数	・適当なエディタにおいてソースコードとしてのプログラムを作成 自由かつ複雑なアルゴリズム ・シミュレーションに特化したライブラリ	・StarLogo 内のエディタにて記述 ・やや特殊な文法・アルゴリズム Logo の拡張	・専用エディタ等においてソースコードとしてのプログラムを作成 自由かつ複雑なアルゴリズム ・デバッグ環境
必要知識	VB 風の簡易言語の把握	Objective-C の理解 Swarm ライブラリについての把握	動作手順の文法（Logo の拡張）の把握	プログラミング言語の習得
難易度	○	×	○	×
作業時間	◎	△	◎	△
機能	○	○	×	△
自由度	○	◎	×	◎

シミュレーション設定

シミュレーションの実行環境、及び実行結果出力の設定を行う。

	ABS	Swarm	StarLogo	既存の開発環境
概要	・GUI による設定 ・様々な実行環境設定 ・５種類の出力形式 ・初期値設定 ・再生機能 　ログ出力 ・連続実行 　初期値変化設定 　レポート出力 ・コントロールパネル 　シミュレーション実行操作 　変数値変更	・フレームワークのプログラミング ・実行用ライブラリ 　実行スケジュール 　変数操作 GUI 　etc. ・入出力用ライブラリ 　グラフ 　マップ 　ファイル ・プログラミングによるカスタマイズ	・既定の実行環境 ・簡易な出力形式 ・GUI による操作インタフェース設定	・すべてプログラミング
必要知識	特に無し	Objective-C の理解 Swarm ライブラリについての把握	動作手順の文法（Logo の拡張）の把握	プログラミング言語の習得
難易度	◎	×	◎	×
作業時間	◎	△	◎	×
機能	○	△	△	×
自由度	○	◎	△	◎

シミュレーション実行環境（操作・出力）

シミュレーションの実行を行う。

	ABS	Swarm	StarLogo	既存の開発環境
概要	・変数値操作 ・連続実行 ・再生機能	・エージェント、変数値操作 ・プログラムによりカスタマイズされた機能	・変数値操作 ・動的な手順の実行 用意された手順 実行途中に入力	・すべてプログラミングに依存
機能	◎	△？	○	？
アピアランス	○	△？	○	？
実行速度	△	○	△	◎
自由度	○	◎	×	◎

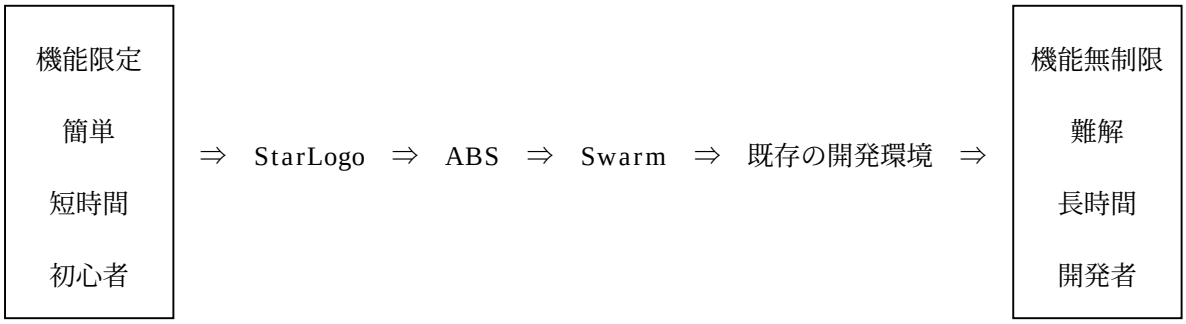
その他

その他の環境。

	ABS	Swarm	StarLogo	既存の開発環境
日本語環境	◎	△	△	◎
OS	Windows	UNIX、Windows	UNIX、Windows、MacOS	—
形式	アプリケーション	ライブラリ	アプリケーション	開発環境
対象	パソコン中級者	開発者	パソコン初心者	開発者
形式	アプリケーション	ライブラリ	アプリケーション	開発環境
自由度	○	◎	×	◎

4.5. まとめ

前項まで細かな比較を行ったが、前出の道具を最も簡単な軸で並べると、以下のようになる。



StarLogo は上の図からも読み取れるように、やはり、教育的な分野で非常に有用であると考えられる。また、Swarm は開発者レベルの研究者を対象としている。
ABS は社会科学の研究、すなわち社会科学者を対象として開発が進められたという経緯があるが、「プログラム開発者程度の技術と時間はないが、比較的複雑なシミュレーションを行いたい」といった、比較的ニッチとなっていた潜在的ニーズに応えるものとなっていることがわかる。
シミュレーションを行う道具にとって、この軸はその道具を特徴づけているように考えられる。

上記以外に、ABS が唯一、モデルの構築に貢献することに着目したい。他の道具においては「頭の中」や「机上で練られた」モデルを、道具を使って具現化するのに対し、ABS はアプリケーション上で「モデルを構築し、実際に動かしてみる」ことが可能である。さらに、異なるマシンでシミュレーションを実行、シミュレーションログの再生ができることから、グループでアイデアを創発して研究する場面においてその真価を発揮する。

最後に、エージェント・ベースのシミュレーションを行う道具をまとめることによって、改めて確認されたことは、各道具には優劣はなく、それぞれ一長一短があり、得意分野が存在するということである。

5. おわりに

エージェント・ベースのシミュレーションを行おうとするとき、どのようなアプリケーションもしくはプラットフォームを選ぶのが適切なのか。前節までに出てきた道具は、いずれも実績があり、ある範囲においては非常に有用であるが、やはり一長一短が存在するため、必要に応じて使い分けることが重要である。今後、これらの道具がそれぞれの道においてさらに発展し、シミュレーションを行う人々により良い環境が提供されることを期待するとともに、ABS がその１つとして広く認識されていくことを願っている。

ワーキングペーパー・シリーズ
Working Paper Series

No.６（最新号）
服部正太・玉田正樹・辺見和晃・桑原敬幸
ABS の概要と類似シミュレータとの比較

No.5
板山 真弓・田村 誠　　Schelling 分居モデルを超えて２
--色盲エージェントの追加　----

No.４
山本 和也　　森林火災の拡大と樹木の密度 ---- ABS の使用例 ----

No.３
阪本 拓人　　生物個体群における自然選択と個体数変動の関係
---- ABS を用いたシミュレーション ----

No.２
板山 真弓・田村 誠　　Schelling 分居モデルを超えて
----　ＡＢＳモデルの検討 ----

No.１
鈴木 一敏　　空間上の生態系モデルにおける個体密集度と系の安定性
----ＡＢＳによるシミュレーション ----