

分散システムシミュレーション言語「StarLogo」の教育利用

本 郷 健

The Use of 「StarLogo」 for Learning about Decentralized System in Education

Takeshi HONGO

概 要

分散処理的モデルは自然科学や工学，生物生態学や社会科学など様々な領域の現象の分析に利用されている。ここでは，分散処理的な見方やモデリングを学ぶためのソフトウェアの一つとして「StarLogo」を取り上げ，その機能や言語仕様を教育的観点から考察した。学習者がモデリングを行い，コンピュータにそのモデルを表現する過程を通して，分散処理的な考え方を学習する教材として，本ソフトウェアを検討した。

分散システムシミュレーション言語として開発された「StarLogo」は，局所的な情報にもとづいて類似する相互作用を行うオブジェクトが大量に存在した時に，その総体すなわち多体効果としての世界をシミュレートする学習ツールとして利用できると考えられる。高等学校の数学，情報，総合的な学習での利用が期待される。

キーワード：モデリング，シミュレーション，StarLogo，分散システム

1. はじめに

平成10年6月の教育課程審議会の審議のまとめでは，情報化への対応として，「各学校段階・各教科等を通じる主な課題に関する基本的考え方」のなかで次のように述べている。

小学校においては「総合的な学習の時間」をはじめ各教科などの様々な時間でコンピュータ等を適切に活用することを通して，情報化に対応する教育を展開する。中学校においては技術・家庭科の中でコンピュータの基礎的な活用技術の習得など情報に関する基礎的内容を必修とし，高等学校においては，情報手段の活用を図りながら情報を適切に判断・分析するための

知識・技能を習得させ、情報社会に主体的に対応する態度を育成することなどを内容とする教科「情報」を新設し必修とすることが適当である¹⁾。

その内容については、学習指導要領の公示までは明確にされることはないが、先の「情報化の進展に対応した初等中等教育における情報教育の推進等に関する調査研究協力者会議第1次報告」では、第3章「次期学習指導要領の改訂に向けた提言」において、「情報の科学的な理解」及び「情報に参画する態度」に関する事項で構成される基礎的な科目であるとされている²⁾。

「情報の科学的な理解」の内容については、「情報活用の実践力」の深化、定着を図ることであり、さらに、様々な情報手段の原理や仕組みを理解させることで、その能力の一般化と一層の向上を図ることである。その例として、シミュレーション手法の活用を挙げている。そこでは、シミュレーション手法によって環境問題や都市計画、経済活動など、科学技術から社会事象に至る広い分野の問題解決に係わる研究が極めて効率的かつ効果的に行われてきていることを取り上げその重要性を指摘している。

シミュレーション手法を活用するためには、対象を目的に応じてモデル化し、その結果の信頼性や有効範囲などを評価する能力が必要であるとしている。そこで触れられている具体的内容は①実験・観察、調査などのデータを正しく収集し分析するための統計的見方・考え方やそのために必要となるモデル化の方法、②将来の結果予想や与えられる条件を変えることによってどのように結果が変化するかを知るために有効となるシミュレーション手法 などである。このように、モデル化やシミュレーション手法は情報教育の中で重視される傾向にある。

2. 学習におけるモデリング及びシミュレーションの利用

コンピュータを利用した学習では予め結果を予想し、解決過程を組み立てる。コンピュータに自らの考えをソフトウェアやプログラムを介して表現し、実行させる。実行した結果を見つめ、予想と異なれば自らの考えを再び構成し直す。このように、コンピュータを利用した学習では、思考実験で重視される仮説をたて、検証するという数学教育や科学教育などで重視される学びの形態が自然と組み込まれる。なかでも、モデル化やシミュレーションはこうした学習形態に適している。

シミュレーションを教育利用する場合、指導の置き方によって、2つの立場がある。一つは対象とする事象を解析し、何らかの方法で表現する段階、すなわちモデル化する過程を重視する立場であり、二つ目が完成したシミュレーションソフトを利用して、設定されたパラメータ

を任意に変化させながらモデル化された現象に与える影響を確認し、現象の変化の解釈を重視する立場である。前者がシミュレーションを作成する過程で必要となるモデル化を重視するのに対して、後者はその活用に重きを置く立場である。従来、教育でのシミュレーション利用といえば後者の場合がそのほとんどであった。

しかし、現象をより深く、探求的に理解するためにはそのメカニズムを分析し、表現する過程を辿ることが大切である。現象をモデル化してシミュレートする学習は思考実験の重要性と相まって、現象の過程やそのメカニズムをプログラムやソフトウェアの中に表現する活動を通して、自らの考えを振り返る自己内省的な学習態度を必要とすることから、学習の深まりが期待できる。

また、新科目「情報」ではモデル化する手法そのものが学習の対象となると考えられる。また、「総合的な学習」では、学習過程そのものを重視することが、学習のねらいと重なるため、モデル化やシミュレーションを生かす場面が多くあるものと期待される。

3. 分散处理的モデルの記述言語としての「StarLogo」

高等学校「情報」や「総合的な学習」でのモデル化やシミュレーションの学習では現象や考えをコンピュータの中にどのように表現するかによって、手続き的な手法やオブジェクト指向的な手法、分散处理的な手法など様々な手法を学ぶ必要がある。

近年、科学技術のみならず社会現象や生物生態学など様々な分野において、分散处理的なモデルを使った解析が注目されている。ところが従来、教育で利用されるシミュレーションソフトが扱う対象は集中处理的で決定論的モデルが多かった。分散处理的モデルが取り上げられることが少なかった理由は、そうした現象を扱う場面が少なかったこともさることながら、分散处理的なモデルの表記に適した比較的易しく、教育的に配慮された言語が開発されていなかったことも一つの原因であると考えられる。

モデル化やシミュレーションで利用されるソフトウェアは、柔軟性、多様性、汎用性が求められる。しかし、汎用性を求め過ぎると教育では使いこなすことが難しくなる。例えば、高等学校の生徒に汎用性の高いC言語を使って、モデル化した現象を記述させようとしても、現状では不可能に近い。一方、汎用性を犠牲にして、個別化した機能を多用すると、その機能に引きずられて発想が制約され、教育的魅力に欠けるものとなりがちである。これらのバランスを配慮しながらソフトウェアを検討することは、教育利用を考える上で重要である。

こうした背景のもと、教育用言語を基礎としたシミュレーション言語がMITを中心に開発

が進められている。現在、インターネットを通して実践と改良が行われている。

本稿では、分散处理的モデルの記述を容易にする「StarLogo」を紹介すると共に、その設計思想と言語的な特徴を具体的なプログラムを通して検討する。また、分散处理的な見方や考え方をモデル化する過程を通して体験できる教材の開発を試みる。

4. StarLogo の言語特性

「StarLogo」は、MITの Michel Resnick らによって、従来の「Logo」言語の機能を包含した言語として開発された分散システムシミュレーション言語である。教育への利用は MIT や Maine 大学を中心に行われているものの、我が国ではまだ殆ど行われていない。ここでの従来の伝統的な「Logo」言語とは、タートルグラフィックス機能、構造化プログラミング機能、リスト処理、再帰的手続き機能などを持った教育用言語のことである。

「StarLogo」は伝統的な基本機能に加えて、以下のような機能が拡張されている。以下では、新しい拡張機能が意味する設計思想を検討して行く。

4.1 オブジェクト指向的機能

「StarLogo」は2の基本的なオブジェクトを提供している。一つが Turtle で、二つ目が Patch である³⁾。

(1) Turtle オブジェクト

Turtle は従来の「Logo」言語で導入されたオブジェクトである。「StarLogo」では、Turtle の数をメモリが許す数だけ作り出すことができる。数千の Turtles が Turtle というクラスのインスタンスとして定義される。各 Turtle は、それぞれのプロパティを持ち、またインスタンスメソッドによって独自の行動をする。従って、観察者は何千という Turtles がそれぞれ独立した行動をした時の、その総体すなわち多体効果としての動きを見ることができる。

例えば、Turtle を鳥と見立て、各鳥にある規則性を持たせて行動させた時に全体としての現象、すなわち鳥の群としての行動を見ることができる。

(2) Patch オブジェクト

Patch の数もメモリが許す範囲に、任意に設定することができる。そのイメージを図1に示す。

ディスプレイ画面は通常では一面黒く表示されるが、図1に示すような四角の小片からでき

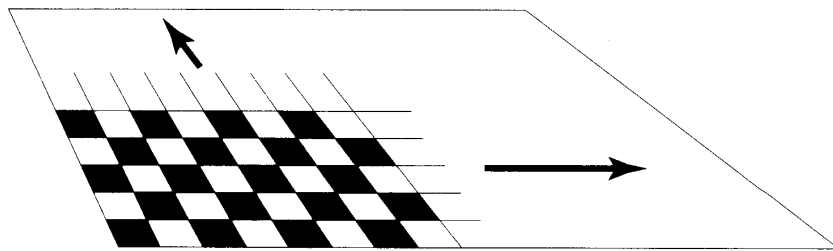


図1 画面上での Patch 構造

ている。各小片を Patch と呼び、各 Patch は一つのオブジェクトとして定義されている。画面の初期設定によって、何百や何千のオーダーの Patch オブジェクトを設定できる。例えば、中央が $(0, 0)$ で四隅が $(50, 50)$, $(-50, -50)$, $(50, -50)$, $(-50, 50)$ となるように設定することができる。Patch はプリミティブなプロパティやコマンドそして新たに局所変数やメソッドを定義できる。このように、Turtle と同様に「StarLogo」の命令を実行する。

Patches の存在は、単に画面を小片に仕切るということではなく、次のような Turtle に対する環境を作り出すことである。Patch はオブジェクトとしてプロパティを持つ。オブジェクトとして定義された Turtles は各 Patches と情報のやりとりを行って、それらのプロパティを知ることができる。例えば、Turtles は、真下に存在する Patches のプロパティの一つのパラメータを呼び出し、その値あるいは状態を知り、そのことによって自らの行動を決定するメソッドを実行することができる。このようなメソッドを持った Turtles は、あたかも自らの足下の環境と対話し、その環境の在り方によって自らの行動を決定するかのように振る舞うことが可能となる。

こうした Turtles と Patches の位置についての関係は、前述したような真上や真下といった単純な位置関係だけでなく、ある Turtle に対して、平面的な位置関係を記述することができる。

環境としての Patches の存在は、この言語にとって重要な機能を提供するため、ここでは具体的な事例を取り上げて検討しておきたい。プリミティブとして用意されている Turtles と Patches の位置関係に関する特徴的なコマンド (`pc-at dx dy`) を図2に示す。この命令は、各 Turtle が相対的な位置で dx , dy の Patch だけ離れたところの Patch の色を返すレポーターである。

このような命令群の存在は、Turtles と Patches の関係を人工的なオブジェクトとその周辺の人工的な環境として見立てることを可能にする。すなわち、Turtle を中心に考えた場合に、Patches がその周辺の環境として見立てることができる。Turtle は自分の周りの局所的な環境を

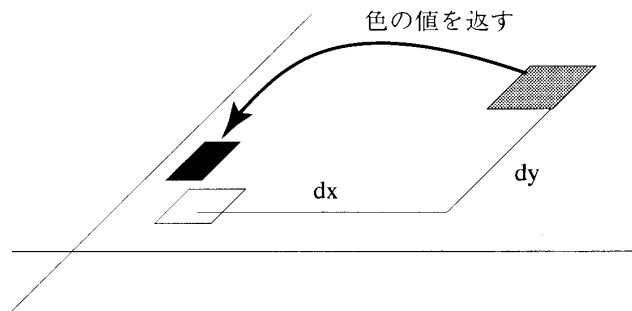
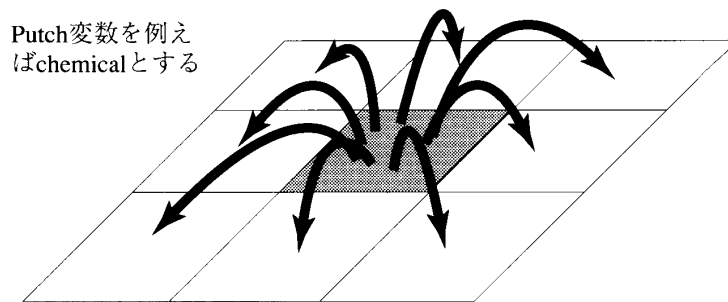


図2 Turtle と Patch のコミュニケーション関係の例

もとに、自らの行動を決定する状況に身を置くことになる。

さて、オブジェクト同士のコミュニケーションの関係は、上述した Turtle と Patch の間以外に、Turtle と Turtle, Patch と Patch との間にも成立する。このようなオブジェクト間のコミュニケーションに適したプリミティブなコマンドが用意されている。

例えば、隣接するオブジェクト同士の相互作用に使われる命令がある。`diffuse` は Patch のなかのある変数の値を隣接する8つの Patches へある割合だけ配分する。`diffuse chemical 0.5` では `chemical` 変数の値の50%を8つの隣接する Patches へ配分する。隣接する Patches の `chemical` 変数の値は50%の1/8の値だけ各隣接する Patch からもらう。このような命令はローカルで局所的な相互作用の現象の記述に適している。



Patch変数を例え
ばchemicalとする

図3 `diffuse chemical number`

上述したような機能は汎用言語を使った場合、関数やサブルーチンとして定義することで実現できることは述べるまでもない。しかし、教育用言語として利用する場合、そうした基本的なレベルからの制作では生徒にあまりにも負担をかけてしまう嫌いがある。こうした汎用言語では、目的とする現象をモデル化してシミュレーションするまで、プログラム作成上の技法の習得が学習の多くの部分を占めてしまう。

このような点で「StarLogo」は、教育的に配慮された言語であると考えられる。具体的な評

価値は今後の実践的な利用を通して行っていきたい。

4.2 並列的処理

並列処理の目的は、計算速度を上げるために利用するのではなく、同時並行的に生じている現象をシミュレーションする方法として採用されている⁴⁾。

複数の Turtles や Patches は純粹に全く独立の手順で動くのではなく、通常は互いが同じような動きをする。そのプログラム例を以下に示す。

```
To setup
  ca
  crt 600
  ifelse who < 100
    [setc black]
    [setc red
     setxy random (screen-edge * 2 + 1) random (screen-edge * 2 + 1)]
end
```

この例は、600匹の Turtles を発生させ、100番未満の Turtles は黒く塗る。それ以外の Turtles は、それぞれが発生させる乱数の値に従って画面上の位置を決める。記述された例から分かるように、600匹の Turtles に対して純粹に独立したプログラムを必要としていない。プログラムには、全体のオブジェクトに共通する事柄が記述される。しかし、個々のオブジェクトの具体的な動きは局所的な情報に従って動作している。

Turtle に共通する手順を記述すれば、システムが各オブジェクトに対して個々のメソッドを発生させる。このことは、プログラム作成者の負担を軽減させると共に、初心者のプログラム作業を可能にする。

このように、Turtles や Patches は大雑把にいえば、同じメソッドを微少時間間隔 (dt) で同期させながら実行している。こうして、大きな集団としての並列処理的な現象を作り出す。重要なのは、多体効果としての全体の現象である。

4.3 状態変数の定義

オブジェクトである Turtle, Patch, Observer に新たにローカルやグローバルな状態変数を定義することができる。変数を定義すると、自動的に値を代入する手順や変数の値を返す手順

を作成する。

例えば, turtles-own [age speed], patches-own [food], global [temperature]は, それぞれ Turtle に age と speed という変数名をもつローカル変数を定義する。また, global では広域変数 temperature を定義する。

これらの変数を定義すると同時に, これらの値を設定する手順 (setage, psetage, osetage-of など) や値を返す手順 (age, age-at id-numer など) を自動的に作成する。これらの手順を用いて Turtle と Patch の間, Turtle や Patch 同士の間で, 局所的でかつ新たな状態でのコミュニケーションや相互作用が可能となる。このことによって, Turtles や Patches の局所的な相互作用を記述し, それらが並列的に処理される。

こうした工夫は, 作成者である生徒の負担を軽減させるとともに, モデル化する世界に多様な環境設定を容易にし, 試行錯誤を必要とする学習者の豊かな発想を引き出すものと期待される。

4.4 GUI への対応による実行時の操作性の向上

学習用のシミュレーション作成ソフトウェアに求められる機能は, モデル化する段階では思考の道具としての柔軟性や思考を活性化させる道具としての簡潔性である。一方, ひとたび完成したシミュレーションソフトを活用する段階では, シミュレーションの意図を率直に表現し, 双方向性と操作性の優れたインターフェースを作成できる多機能性が求められる。

図4は実行画面を示す。図4に示すように手順を実行させるためには, 手順名を入力しても可能だが, 手順をアイコン化してインターフェース画面に配置しそれをクリックする。プログラム内で利用される変数の値はアイコン化されたバーによりマウスで任意に変えることができる。また, プログラムを実行しながら変数の値の変化を表示させたり, 値の変化をグラフ表示する PLOT 画面がある。図4に示すようにシミュレーションソフトとして利用する場合のインターフェースは双方向性を重視したグラフィック画面となっている。このように作成したソフトを実行する段階では, シミュレーションソフトとしての多機能性が埋め込まれている。

5. 分散的処理モデルの開発教材例

キャストイは複雑系のモデルの特徴として, 次の3つの条件を挙げている⁵⁾。

- (1) モデルを構成している要素 (これを「エージェント (agent)」と呼ぶ) の数は中程度である。つまり, 少なすぎてもいけないし, 多すぎても行けない。しかし, この中程度という

のはモデルに依存している。

- (2) エージェントは知性を持っている。経済や政治などにおいて各エージェント（個人かもしれないし、企業かもしれないが）は、知的に決定している。
- (3) 各エージェントは局所的な情報にもとづき相互作用をする。完全な情報を知って意志決定ができないところが重要である。

ここでは (1) や (3) の特徴をもつモデルを分散处理的なモデルと呼ぶ。

ここでは高等学校の生徒がプログラムの作成を通して、分散处理的な考え方を学習することを目的とした教材の開発を試みる。分散处理的なモデルは、従来取り扱われることが少なかったため、身近な題材を取り上げて基礎的な段階から体験的に学習できるよう工夫する。

以下のような教材を段階的に配置する。

(1) StarLogo Geometry 教材

- ① StarLogo の基本命令を理解する。
- ② 局所的な情報によって動作する object の多体効果としての図形作成を体験する。

(2) 分散处理的モデリング教材

- ① 現象を「粗視化」する見方について知る。大域的な現象を知るためには物事を大雑把に見るという粗視化の重要性を知る。
- ② 粗視化したモデルの分散处理的な記述技法を知る。

(3) 複雑系モデリング教材

キャスティの定義に沿った現象のモデル化を体験する。

5.1 StarLogo Geometry の教材例

直線 ($y = x$) は、StarLogo では次のように描くことができる。まず、画面上にランダムに点 (Turtle object) を配置する。各 object は自分の状態（この場合は位置情報）を知る。各 object は y 座標を現在の x 座標に書き換える。各 object がその操作を行うと、全体として直線が現れる。

```
to line
ca
crt 4000
setxy random screen-size random screen-size
sety xcor
end
```

同様に、円、三角関数などを描く⁶⁾。ここで表現された図形やグラフは、各 object が局所的な情報にもとづいて共通の動作を行った時に、その結果として全体として現れる形であると解釈できる。その考え方を意識的に手順化することによって、分散的な処理に気づかせる。

5.2 分散处理的モデリングの教材例

一般的に、円は「平面上で一定点から一定の距離にある点の全体でできる曲線」と定義される。また、Turtle Graphicsでは「一定の曲率半径を持つ曲線」であることを強調して、Turtleに次のような運動を行わせて円を描く。微小距離進んだら一定の方向に向きを変える、この動作を向きの変化の合計が360度になるまで繰り返す。この運動はちょうど、人がグラウンドに円を描く時の動作である。この手順は子供たちの日常的な身体的運動と結びついており円の定義としてわかりやすいと主張される。

ここでは、各オブジェクトが局所的な情報をもとに行動を繰り返すことによって、ある定常状態として構成される図形として円（丸）を描く。円を運動する Turtle object の定常状態の集合体と捉える。Turtles は互いに相互作用を及ぼし合いながら、円の一部として構成される。それぞれの Turtle は単純な2つのルールに従う。(1) 一つの Turtle は、隣の2つの Turtles と等距離を保とうとする。(2) Turtles は基本的に全てが互いに反発するように作用する。この行動は、ちょうど子供たちが手を結び、互いに引っ張り合いながら丸い輪を作る動作と似ている。そのプログラムの結果を図4に示す。

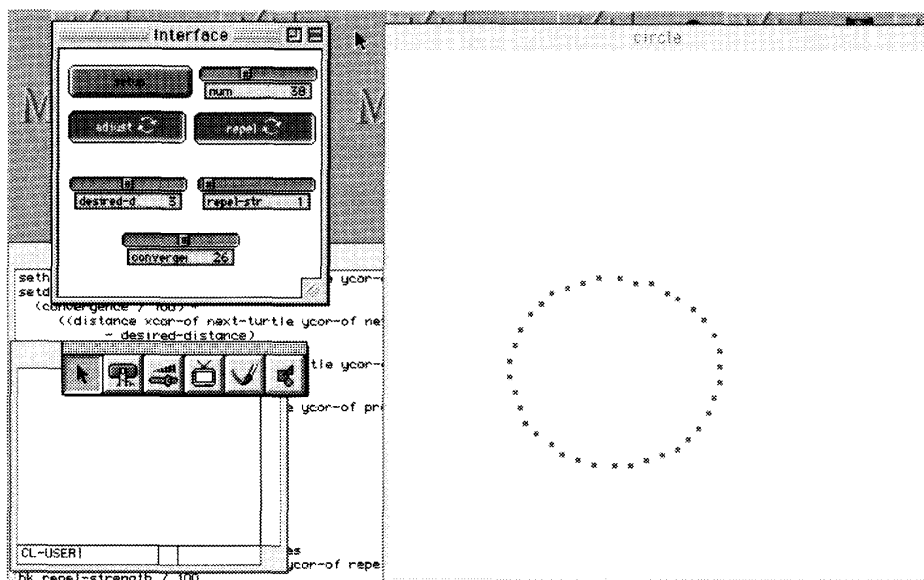


図4 動的に描いた円

円が形づけられたところで、互いの距離を決める DESIRED-DISTANCE スライダーを調節すると収縮，膨張しながら円の形を保ち続ける。その動きは輪ゴムが伸縮しながら円を形作るのに類似している。

6. おわりに

筆者らは、高等学校「情報」や「総合的な学習」における題材として、分散処理的なシステムのモデリング及びシミュレーション技法の学習を導入しようとしている。それを実現する一つのツールとして StarLogo を取り上げた。本稿では具体的な教材を試作することを通して、本ソフトウェアの持つ基本的な機能や設計意図を考察し、教材としての可能性を検討した。結果、プログラムの記述に必要な基本命令が教育用言語として実績のある Logo の拡張によって実現されているところに、教育での利用の可能性を見出すことができる。このことによって、生徒が自らの考えを比較的容易に表現できるようになり、試行錯誤しながら思考実験するような活用法が可能となるものと思われる。

扱う対象モデルは、磁性体のモデルであるイジング・モデル，森林生態系で利用される森林格子モデル，伝染病伝播のコンタクト・プロセスや浸透現象のパーコレーション・モデルなどの確率モデルや自己組織化現象などの分散処理的モデルなどが考えられよう。次のような活用場面が考えられる。

- 1) 高等学校の数学，情報や総合的な学習で，生徒が分散処理的なシステムのモデル化やシミュレーションを行う思考実験ツールとしての利用。
- 2) インターフェースが改善されたことで，教師のシミュレーション教材開発ソフトとしての利用。

今後は具体的な指導計画の開発と授業での実践を通して，教育的意義を実証的に検討することが求められる。

引用文献

- 1) 教育課程審議会，「幼稚園，小学校，中学校，高等学校，盲学校，聾学校及び養護学校の教育課程の基準の改善について（審議のまとめ）」，1998，www.monbu.go.jp/series/00000045/#1.
- 2) 情報化の進展に対応した初等中等教育における情報教育の推進等に関する調査研究協力者会議，「体系的な情報教育の実施に向けて（平成9年10月3日）」，1997，www.monbu.go.jp/series/00000026/#04
- 3) Mitchel Resnick，「Turtles, termites, and Traffic Jams」，1997，pp. 42-44.

- 4) 同上, pp. 45-46.
- 5) ジョン・キャステイ, 「複雑系による科学革命」講談社, 1997, p. 44.
- 6) 本郷健, 「分散システムシミュレーション言語「StarLogo」の教育への検討」, 1998年度数学教育学会春季年会発表論文集, 数学教育学会, 1998, pp. 22-24.