社会・環境シミュレーションにおけるインタラク ション層の構築

Construction of Interaction Layer on Socio-Environmental Simulation

鳥居 大祐

Daisuke Torii

京都大学情報学研究科社会情報学専攻

Department of Social Informatics, Kyoto University

torii@kuis.kyoto-u.ac.jp, <http://www.lab7.kuis.kyoto-u.ac.jp/~torii/>

石田 亨

Toru Ishida

(同上)

ishida@i.kyoto-u.ac.jp, <http://www.lab7.kuis.kyoto-u.ac.jp/~ishida/>

keywords: multiagent-based simulation, agent interaction, agent programming languages and environments

Summary

In this study, we propose a method to construct a system based on a legacy socio-environmental simulator which enables to design more realistic interaction models in socio-environmental simulations. First, to provide a computational model suitable for agent interactions, an interaction layer is constructed and connected from outside of a legacy socio-environmental simulator. Next, to configure the agents interacting ability, connection description for controlling the flow of information in the connection area is provided. As a concrete example, we realized an interaction layer by *Q* which is a scenario description language and connected it to CORMAS, a socio-environmental simulator. Finally, we discuss the capability of our method, using the system, in the Fire-Fighter domain.

1. はじめに

地球の温暖化, 森林火災, 地震災害, 食料問題などは世界における環境・災害問題の大きなトピックであり, 情報技術が果たす役割が期待されている. 人と環境を含めた総合的な社会・環境シミュレーションが実現すれば, 問題に対する政策や訓練などに反映することができる.

社会・環境シミュレーションでは, エージェントは他のエージェントとの交渉・協調や環境からの観測を元に自身の状態を変化させ, また, 環境へ行動を行う. したがって, 環境を表す計算モデルの他に, エージェント間やエージェントと環境間のインタラクションが重要な要素であり, シミュレーションを通じてそれらを十分検証・設計することが重要である. そのためには, インタラクションを扱う計算モデルを元に, 様々なインタラクション記述を試せることが望ましい.

しかしながら, 社会層・環境層のモデリングが可能な既存の社会・環境シミュレータの多くは, シミュレーションに便利なツールが様々用意され, 利用価値の高いものであるが, エージェントのモデル記述については, その自由度を高めるために専用の API などを使い行動モデルを一般に記述するものであり, インタラクションを専用に扱う計算モデルを元にした記述を提供していない (図 1 参照). したがって, 既存の社会・環境シミュレータに

対し, インタラクション記述を与え, シミュレーションを行える環境を構築することが必要である. さらに, シミュレーションでの検証結果を現実の社会システムの設計などに反映させるためには, インタラクションをシナリオとして記述できるだけでは不十分であり, インタラクションに関わるエージェント個人の能力のばらつきや環境から受ける能力の制限を勘案することで, より現実 に即したインタラクションを検証・設計できることが重要となる.

このような背景を受けて, 本研究では, 社会・環境シミュレーションにおいて, より現実 に即したインタラクションモデルの検証・設計を行うことができるシミュレータを, 既存の社会・環境シミュレータを利用して構築する手法を提供することを目的とする. そのために, 次のような接近法を取った.

1) インタラクションのみを独立に記述し, 既存の社会・環境シミュレータでそれを利用するためには, エージェントがシナリオを解釈し, それに従って動作する機構が必要となる. しかし, 既存の社会・環境シミュレータの拡張性を考えるとそれを内部に構築するのは現実的でない. そこで, シミュレータに大きな変更を加えず外部のインタラクション層にてシナリオを解釈しエージェントを制御するアーキテクチャを採用する. 2) また, エージェントの個人能力を微調整するためには, 既存の社会・

図 1 従来の社会・環境シミュレータ

環境シミュレータの内部を変更する必要があるが、同様にその拡張性を勘案すると内部の変更は避けたい。そこで、本アーキテクチャにおけるインタラクション層と社会・環境シミュレータの間を流れる情報を制御することで、シミュレータの内部を変更せずより簡単に実現する手法 (Connection Control) を提案する。

社会・環境シミュレーションの実現手法としては、ロボカップレスキュー [Takahashi 02] や U-MART [Kita 03] が関連研究として挙げられる。環境としてそれぞれ災害現場や株取引市場を与え、そこにエージェントが参加するためのプロトコルを規定することで様々なエージェントの参加を可能としている。したがって、両者ともエージェントのインタラクション記述を既存の社会・環境シミュレータのエージェントに外部から与える本研究とはアーキテクチャの上で異なる。また、シナリオ記述言語 Q [Ishida 02b] は、既存のエージェントに外付けでインタラクション記述を与えるものである。本研究では、より現実に即したインタラクションモデルを設計・検証するためのシミュレータを得るために、既存の社会・環境シミュレータを利活用するという観点から、Connection Control の実現を含めた適切なインタラクション層の接続手法を考察・提案した点がその意義となる。

以下、本論文では、まず、提案手法について述べる。次に、自然資源管理を目的とした既存の社会・環境シミュレータである CORMAS と上述の Q を結合した CORMAS/ Q の実装を示す。最後に、森林火災のシミュレーション [Cohen 89] を行うことで、本手法の可能性を議論する。

2. 社会・環境シミュレーションにおけるインタラクション層の構築

2.1 アーキテクチャ

社会・環境シミュレータの外部にインタラクション層を構築したシミュレータのアーキテクチャを図 2 に示す。このシミュレータは、社会層と環境層の二層より構成される (図中では実線で区切られている)。社会層は、さらに二層よりなる。交渉や協調の過程の記述 (インタラクション記述) を解釈し、解釈結果をエージェントに伝

図 2 インタラクション層が構築された社会・環境シミュレータ

図 3 システム構成

える層 (インタラクション層) と解釈結果を受け取ったエージェントにより実際のメッセージの伝達や環境の観察、環境への行動が行われる層 (実行層) である。即ち、インタラクション層により実行層の制御が行われる。前者は、専用に用意されたインタラクションモデル提供システムにより、後者は、既存の社会・環境シミュレータにより記述が提供される。環境層の記述は社会・環境シミュレータにより提供され、エージェントや自然の作用による環境内での変動の伝播が記述される。

図 2 には、インタラクションモデルとエージェント間に対応付けが示されている。尚、エージェントの協調を表すインタラクションモデルは、複数のエージェントが関わるやり取りの手順として記述される場合もあるが、エージェントが外部からインタラクションを受け取り動作する本アーキテクチャでは、最終的にそれをエージェント一体の視点で分解した記述が割当てられなければならない。

2.2 システム構成

インタラクション記述は、その計算モデルに従って解釈されるモデル解釈系を必要とする。モデル解釈系は、インタラクション記述を解釈し、それをインタラクシ

表 1 Connection Control の効果と Connection Description の指定例

インタラクションの種類	エージェント能力への効果	パラメータの指定内容	効果の例	例における Connection Description の delay のタイミング指定
他エージェントとのコミュニケーション	情報伝達能力の微調整を行うことに相当．	accuracy は通信路上でメッセージが欠落する割合，delay は本来よりも情報伝達が余計にかかる時間．	accuracy は雑音などから時々会話が届かない状況，delay は会話内容が完全に伝達できるまでにかかる時間が通常よりも長くなる状況など．	対象となるコミュニケーションがまず開始される必要があり，また，実行結果を解釈系に返すのを遅らせることで同等の効果を得られるので，情報発信者/受信者ともに (b)．
環境の観察	環境から情報収集能力の微調整を行うことに相当．	accuracy は環境からの情報収集の成功率，delay は本来よりも情報収集が余計にかかる時間．	常に遠隔地にいる人が離れた対象を把握する場合において，accuracy は情報収集の時々失敗や，delay は遅延のため常に少し過去の情報しか手に入らない状況など．	その時点での情報をまず取得する必要がある，また，実行結果を解釈系に返すのを遅らせることで同等の効果を得られるので (b)．
環境への行為	環境への行動能力の微調整を行うことに相当．	accuracy は環境への行為が成功する割合，delay は本来よりもその行為の完了に余計にかかる時間．	能力の高い人であるならすぐに行いが達成できるものでも，人によっては失敗したり，時間が余計にかかったりする状況など．	環境への行為は環境の状態を変化させるため，実行前で伝達を止めておく必要がある (a) ．

- (1) 情報伝達の成功率 (accuracy)
 (2) 情報伝達の遅延時間 (delay)

成功率は%で指定できる．例えば，80%とした場合，接続部にて5回に1回情報が欠落する．また，遅延時間は社会・環境シミュレータの時間概念の単位時間に基づいた数にて指定する．例えば，離散時間であるなら自然数にて指定し，3と指定された場合，3単位時間情報の伝達を遅らせる．

accuracy は，指定されたインタラクションを指定した確率で，エージェントへ伝えないことで実現される．delay に関しては，インタラクションの性質を考慮して望まれる効果を導くように，Connection Description にて処理のタイミングが以下のいずれかで指定される必要がある．

- (a) 実行依頼の伝達を指定時間遅らせる．
 (b) エージェントから解釈系への実行結果の伝達を指定時間遅らせる．

前節で述べた3種のインタラクションそれぞれについて，本手法の効果と Connection Description の上記タイミングの指定例を表1に示す．

Connection Control が与える能力の制限は，シミュレーション全体に渡るものに限られる．例えば，環境から受ける能力の制限に関しては，その環境から常に受ける障害を表現するものである，ある特定の状況における制限（あまりに距離が離れている時にコミュニケーションに障害がでるなど）を表現したければ，社会・環境シミュレータ内のモデル記述で実現することとなる．また，本手法はエージェントの実行の前後で情報の制御を行うので，そのために表現力に制限がある．例えば，火災の鎮火で火力に状態量がある場合，accuracy では鎮火作業を失敗したとしても火力が半分になる状況を，delay では火力を本来の倍の時間かけてゆっくりと状態量を0にする状況を，それぞれ表現することは難しい．

3.3 Connection Control の実現

Connection Control を実現するためには，実行依頼は，逐次的に行うインタラクションを含んではいけない．なぜならば，実行依頼に逐次的に行われるインタラクションが複数含まれていた場合（例：Aの後にBを行う），情報調整機能は，2番目以降のインタラクションが開始されるタイミングを得られず，ユーザが意図する delay の処理を実現できないからである（したがって，Aを依頼し，完了後Bを依頼するようにする）．つまり，これを図3におけるどちらかの情報調整機能で実現しなくてはならない．

社会・環境シミュレータが離散時間による時間進行を行う場合，delay の処理のために，社会・環境シミュレータの時間進行を Connection Control Module にて把握する必要がある．このことを考えると，Connection Description は，社会・環境シミュレータ側の情報調整機能に付属して設置されなくてはならない（図3参照のこと）．

尚，本アーキテクチャにおいて，エージェントに対する入力は3.1節における3種のインタラクション，エージェントからの出力はその実行結果で規定され，さらに本節の要件を実現する限り，3.2節で述べた効果の範囲はあるものの，Connection Control は社会・環境シミュレーションに一般に適用されるものである．

4. Q と CORMAS の接続

実際に社会・環境シミュレータである CORMAS に対して，インタラクション記述とその解釈系を持つ Q を用いて，前章で述べた手法に従って，インタラクション層を構築する．本章では，Q と CORMAS の特徴について述べた後，接続の実装を論じる．

整に与えることができ，様々な観点からインタラクションモデルを検証することができると考える．

MAS 側に Q 側の実行要求を伝える役割と、実行結果を受け取って Q 言語処理系へ返す役割がある。CORMAS Communicator は、エージェントから適宜呼び出されると、キューやアクションの実行依頼を受け取りエージェントへそれを渡す役割と、エージェントにより対応する CORMAS メソッドを呼び出され実行された結果を共有ファイルに書き込む役割がある。

また、情報調整機能は次のように実現する。 Q 言語処理系は、3・3 節で述べた条件を満たすインタラクションの集合を出力する^{*6}ので、特別な加工は必要ない。Connection Control を実現するために、CORMAS Communicator に Connection Description を与えた。情報の制御は、CORMAS Communicator がエージェントプログラムを呼び出す際に、Connection Description を参照して行われる。CORMAS は離散時間モデルによりシミュレーション時刻を進行するため、遅延時間 (delay) の実現のためには、CORMAS 上での時間の進行を CORMAS Communicator が把握する必要がある。

尚、本接続は、 Q のこれまでの接続例である FreeWalk [Nakanishi 04] や MSAgent [Ishida 02b] と比較して、情報調整機能部を構築して Connection Control を実現したという点で新たな実装例を示している。

図 5 に、アクション ‘make-fireline’ が実行された場合を例にしてこの様子を示す。

5. Fire Fighter モデル

5.1 問題

本研究での提案手法の可能性を議論するために、例題として、 Q と CORMAS により構築したシミュレータにより、Cohen らによって行われた火災シミュレーション [Cohen 89] を実装した。その目的は、様々な状況を想定した複数のシミュレーションを行うことで、適切なエージェント間のインタラクションを設計することである。

この問題では、火災の広がりを防ぐため、全体の状況を見ながら指示を出す Fireboss と、その指示に従いながら公園内で行動する Fire Fighter が存在する。Fire Fighter はブルドーザーであり、Fireboss の指示に従いながら、火災を囲むように木を取り除いた道 (fireline) を作ることで、火災の広がりを防ごうとする。環境は、セルごとに川、道路、木、火などの状態を持っている。火災は、風の速度と向き、湿度、地形の特性などによって広がっていく。仮に風が強ければ、川や fireline などの環境的境界を飛び越えて火が燃え広がっていく。

シミュレーションは公園内で火災が発生し、2 体の Fire Fighter が火災の位置と風向きなどから決定された集合ポイントに配置されるところから始まる。Fireboss は、その時点での火災の状態から 2 体の Fire Fighter が取

(a) 修正前

(b) 修正後

図 6 Fire Fighter の修正前後のインタラクションモデル (状態遷移機械)

るべきルートを決し指示を出す。各 Fire Fighter は、指示されたルートに従って時計回りと半時計回りに動き、木を取り除いていく。Fire Fighter は、指示されたルート上に火災が広がっていたり、急激な環境の変化により、Fire Fighter に突如として火災が及んだ場合は、自分の判断や Fireboss と連絡を取りながらルートの変更を行う。Fireboss は、Fire Fighter からの要求があれば、対応した指示を Fire Fighter に返信する。例えば、Fire Fighter から進行ルート上に火が広がっていたために修正ルートの要求があれば、エージェントの位置と環境全体の情報から作成したルートプランを Fire Fighter に返信する。また Fire Fighter は、川などの境界や他のエージェントによって作られた fireline にぶつかった場合、次の行動を Fireboss に問い合わせることもできる。Fireboss は fireline が完成されていると判断すれば、作業の終了を指示する。仮に、火災が他の場所に飛び火などとして広がっていれば、別のルートを指示する。

このように、Fireboss と Fire Fighter のインタラクションによりシミュレーションが進行し、最終的に火災の進行が食い止められる。

^{*6} 正確には、条件を満たすよう Q コネクタで設定可能ということである。

図 7 Fire Fighter シミュレーションのスクリーンショット: 森林は緑, 川は水色, 炎は赤, fireline は黒, エージェントは三角形で示されている

5.2 モデル

CORMAS 上には, 各セルに森林, 空き地, 川などの地形情報と風の向きと速度の天候情報を含む環境のモデルを構築し, 火災が時刻と共に広がるシミュレーションが実現された. 火災の伝播モデルはセルラオートマトンにより与えられる. 一方, Fireboss と Fire Fighter に対応するエージェントを CORMAS に定義した. Q シナリオでは, エージェント間の社会的インタラクションや, Fire Fighter と環境とのインタラクションのモデルが記述されている. また, CORMAS 側では, Q 側からのインタラクションの依頼を受けて, 他のエージェントとのコミュニケーション, 環境変化の観測, セルの状態変化や移動などの行動を行う.

すでに述べた Fireboss (FB) と Fire Fighter (FF) のインタラクションは, 状態遷移モデルで表現することができる (図 6 に Fire Fighter のモデルを示す). シミュレーションでは, Q 言語処理系は状態遷移図を直接的に処理できないので, 状態遷移モデルが素直にコーディングされた Q シナリオが用いられる.

5.3 結果

図 7 にシミュレーションのスクリーンショットを示す. 空間には森林が広がり, 一本の川が流れている. 初期値として空間の上部中央に小さな火災を発生させ, 北風により火災を拡大させていく. Fire Fighter の初期位置は, 火災発生源の北側とした (図 7(a) 参照).

ここでは, Connection Control を利用することで, インタラクションモデルの修正過程を示す (図 6 (a) から (b) へ. 点線の囲みが修正箇所).

まず, Fireboss から Fire Fighter への情報伝達が自然環境からの雑音により制限される状況を Connection Description により表現した. これは, Fireboss から Fire Fighter へのすべての通信に対して, 情報伝達の遅延時間 (delay) を 3 (ステップ) にすることで行った. 遅延は, Fireboss, Fire Fighter 共に, 3.2 節における (b) のタイミングで行うよう指定した. すると, Fire Fighter の

進行が遅々として進まず, 環境の画面下部が火でほとんど覆われてしまうこととなった (図 7(b)). 時間の進行と共に Fire Fighter の動きを中心に原因を追跡した結果, 通信遅延のため, Fireboss からの修正ルートに完全に把握する頃には, 火災が進行して修正ルート上に火災が及んでおり, 再度, Fireboss へ修正ルートを要求しなければならないことが頻繁に起こっていることが分かった.

次に, Fireboss から受信した修正ルート上に火が及んでしまっている場合は, Fire Fighter が自分の周りの情報だけ進路をさらに修正できるようモデルを改良した (図 6(b)). これにより, 受信ルート上に火災が及んでいても自分でルートを微調整することにより, Fireboss との通信回数を抑えることができた. 結果として, 遅延がない時ほど火災の進行を抑えることができないものの, fireline と環境の境界 (川) により火災を囲む状態を環境内に作ることができた (図 7(c)).

5.4 議論

本例題により, 提案手法におけるアーキテクチャとシステム構成により構築した具体的な社会・環境シミュレータが問題なく動作したことを示せたと考える.

また, 本例題である Fire Fighter のような環境と人が複雑に絡むドメインでは, 複雑なインタラクションモデルが必要となり, 提案手法が有効であると考えられる. Q では, 計算モデルとして状態遷移機械を用いることで, 同時に行う複数の観測と観測に対応した行動を記述できるので, 環境や司令部からの観測が同時に必要な本ドメインに適していると言える. Q シナリオが与える記述性や可読性は, CORMAS 単独で達成するには労力を要する. なぜならば, 状態遷移機械モデルに従った様々なシナリオを汎用的に処理するためには, CORMAS 上で Q 言語処理系と同様の機構を既存シミュレータの枠組みの中で工夫して組み込む必要があるからである.

さらに, Connection Control の使用例を示すことで, Connection Control という手法が, より現実に対応したインタラクションモデルを設計する上で有効であることも

示せたと考える。本手法は、提案した接続手法により初めて実現されるものである。従来では、既存の社会・環境シミュレータの内部を変更しなくてはならず、本手法を適用することでエージェントの能力や環境からの制限を勘案したシミュレーションが容易に実現できる。注意しなくてはならないのは、本手法の効果への解釈をドメインの専門家と議論するなどして定義する必要があるということである。このことを考慮しつつ、その他のドメインについても適用することで、引き続き本手法の評価・精錬が必要となる。

6. お わ り に

本研究は、社会・環境シミュレーションにおいて、より現実に即したインタラクションモデルの検証・設計を行うことができるシミュレータを、既存の社会・環境シミュレータを利用して構築する手法を提案した。本手法では、まず、ユーザがインタラクション設計に集中できるように既存の社会・環境シミュレータの外部にインタラクション層を構築した。さらに、接続されたシステム間を流れる情報を制御することにより、エージェント個人の能力のばらつきや環境から受ける能力の制限を表現（Connection Control）することで、より現実に即したインタラクションモデルを検証・設計できるものとなっている。また、実際に、インタラクションモデルを拡張状態遷移機械モデルで提供する *Q* 言語処理系と社会・環境シミュレータである CORMAS を接続することで、具体的な実装法を示した。最後に、Cohen らによる Fire Fighter モデルを CORMAS/*Q* により実施することで、Connection Control によるインタラクションモデルの設計プロセスを示し、提案手法の可能性を議論した。

我々は今後、本論文で示した手法を実問題への適用すると共に、次のように CORMAS/*Q* の適用を検討している。*Q* は、シナリオ制御のエージェントを容易に人間が制御するアバターに置き換え可能なので、参加型シミュレーション（人間とソフトウェアエージェントが協力しながらシミュレーションを進める）に適している。また、*Q* はシナリオの記述に拡張状態遷移機械を用いているため、参加型シミュレーションの結果から、機械学習を用いてシナリオの改善を図ることも可能である。

謝 辞

御助言を頂いたパリ第六大学の Alexis Drogoul 教授、本実装へ協力頂いた Stéphane Bonneaud 氏に感謝の意を表す。本研究は、日本学術振興会科学研究費 基盤研究 (A)(15200012, 2003-2005) の助成金により行われた。

◇ 参 考 文 献 ◇

- [Bousquet 98] Bousquet, F., Bakam, I., Proton, H. and Le Page, C. Cormas: common-pool resources and multi-agent Systems. *Lecture Notes in Artificial Intelligence*, Vol. 1416, pp. 826-838, 1998.
- [Cohen 89] Cohen, P. R., Greenberg M. L., Hart, D. M. and Howe, A. E. Trial by fire: Understanding the design requirements for agents in complex environments. *AI Magazine*, Vol. 10, No. 3, pp. 32-48, Fall 1989.
- [Ishida 02a] Ishida, T. Digital City Kyoto: Social Information Infrastructure for Everyday Life. *Communications of the ACM (CACM)*, Vol. 45, No. 7, 76-81, 2002.
- [Ishida 02b] Ishida, T. *Q*: A Scenario Description Language for Interactive Agents. *IEEE Computer*, Vol. 35, No. 11, pp. 54-59, 2002.
- [Kita 03] Kita, H., Sato, H., Mori, N. and Ono, I. U-Mart System, Software for Open Experiments of Artificial Market. In *Proceedings of IEEE SIRA2003*, 1328-1333, 2003.
- [Mathevet 03] Mathevet, R., Bousquet, F., Le Page, C. and Antona, M. Agent-based simulations of interactions between duck population, farming decisions and leasing of hunting rights in the camargue (southern france). *Ecological modelling*, Vol. 165, pp. 107-126.
- [Murakami 03] Murakami, Y., Ishida, T., Kawasoe, T. and Hishiyama, R. Scenario Description for Multi-Agent Simulation. In *Proceedings of International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS-03)*, pp. 369-376, 2003.
- [Nakanishi 04] Nakanishi, H. and Ishida, T. FreeWalk/*Q*: Social Interaction Platform in Virtual Space. *ACM Symposium on Virtual Reality Software and Technology (VRST2004)*, pp. 97-104, 2004.
- [Perez 02] Perez, P., Ardlie, N., Kunepong, P., Dietrich, C., Merritt, W. S. CATCHCROP: modeling crop yield and water demand for integrated catchment assessment in Northern Thailand. *Environmental Modelling and Software*, Vol. 17, No. 3, pp. 251-259, 2002.
- [Takahashi 02] Takahashi, T., Tadokoro, S., Ohta, M. and Ito, N. Agent Based Approach in Disaster Rescue Simulation, *RoboCup 2001: Robot Soccer World Cup V*, Lecture Notes in Artificial Intelligence, pp.102-111, 2002.

〔担当委員：栗原 聡〕

2005 年 6 月 3 日 受理

著 者 紹 介

鳥居 大祐(学生会員)

2001 年京都大学工学部情報学科卒。2003 年同大学院社会情報学専攻修士課程修了。現在、同大学院社会情報学専攻博士課程に在学中。マルチエージェントシミュレーション、P2P システムに興味を持つ。

石田 亨(正会員)

1976 年京都大学工学部情報工学科卒業、1978 年同大学院修士課程修了。同年日本電信電話公社電気通信研究所入所。ミュンヘン工科大学、パリ第六大学、メリーランド大学客員教授など経験。工学博士。IEEE フェロー。情報処理学会フェロー。現在、京都大学大学院情報科学研究科社会情報学専攻教授、上海交通大学客員教授。自律エージェントとマルチエージェントシステム、セマンティック Web 技術に取り組む。デジタルシティ、異文化コラボレーションプロジェクトを推進。