

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. 9

コミュニケーション競争モデル

ABSによる組織形成メカニズム分析

高橋伸夫*・桑嶋健一*・玉田正樹**

2000年10月

(* 東京大学大学院経済学研究科)

(** 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>

．はじめに

社会現象、生命現象に対する新しいアプローチとして知られる複雑系(complexity)の分野を象徴するものとして、遺伝的アルゴリズムやカオス理論、そしてここで取り上げるマルチ・エージェント・シミュレーションがある。ここでいうエージェントとは、ユーザの設定したルールに基づいてコンピュータ上で行動する主体を指している。マルチ・エージェント型ではエージェントが複数いて、そのエージェント同士が互いに影響を与え合うことになるので、ルール自体は簡単なものでも、個別エージェントの行動を積み上げた全体では予測できない複雑な動きをすることになる。

マルチ・エージェント型のシミュレーションは、1990年代に注目を集めるようになった比較的新しいものだが、幅広い分野で試みられている。教育を目的としたマルチ・エージェント・シミュレータとしては、複雑系のメッカ、米国 SantaFe 研究所で開発された Swarm が有名で、世界で広く使われている。マサチューセッツ工科大学では、教育用の StarLogo も使用されてきた。しかし、Swarm では、日本語を使うことができず、シミュレーションを組むのに特殊な言語 (Objective C) を習得しなければならなかったり、インストールが難しかったりと、特に社会科学系の研究者や学生にとって、大きな負担になっていた。そこで、Swarm のコンセプトを継承しつつ、日本語環境で利用しやすいマルチ・エージェント・シミュレータとして開発されたのが ABS である。ABS とは Agent Based Simulator の略で、マルチ・エージェントのシミュレーションを行うためのソフト、シミュレータの一つである¹。

この研究では、このシミュレータ ABS を使うことを念頭に置いて、マルチ・エージェント型のモデルを考えてみることにしよう。本研究で扱うモデルは、ある意味でコミュニケーション自体を餌としている競争モデルなので、「コミュニケーション競争モデル」と呼んでいる。このモデルを使って、組織がどのように形成され、どのような性質をもった組織が競争に生き残るのかについて、そのメカニズムをコミュニケーションに絞ってモデル化し、ABS を使ったコンピュータ・シミュレーションによって考察してみよう。また、実際の製薬企業におけるコミュニケーションを例に挙げて、その実際的なインプリケーションについても提示してみよう。

．コミュニケーションとゲートキーパー²

それでは、経営学でコミュニケーションがどのように取り扱われてきたのか、イノベーション、研究開発との関係で、整理しておこう。

経営学において、イノベーションに関する体系的な研究が始まったのは、1960年代半ばのことである。その嚆矢的研究成果の一つとして、Myers & Marquis (1969)がある。この研究では、5つの産業における567のイノベーションを対象として、イノベーションの成功要因について包括的な分析が行われた。そして、イノベーション研究において、研究開発組織のコミュニケーションに焦点を当てたいわゆる「コミュニケーション・アプローチ」の基礎を作りその後多くの関連研究を刺激したのが、Allen (1977)である。Allen はもともとボーイング社のリサーチ・

¹ (株)構造計画研究所創造工学部のホーム・ページ <http://www2.kke.co.jp/> から ABS 試用版をダウンロードして使うことが出来る。また ABS を教育目的で使用する場合には、無料貸与もされている。

² この節の記述は、桑嶋(2000)を元にしてている。

エンジニアであったが、R&D マネジメントに興味をもって MIT のマネジメント・スクールに留学し、Marquis 教授と共同研究をしたのをきっかけに経営学者となった。

Allen は、「コミュニケーションが研究開発のパフォーマンスを向上させる」という予測を立て、研究開発組織におけるコミュニケーションの実態調査を実施した。その結果、パフォーマンスの高いプロジェクトでは、低いものに比べて、プロジェクト内でも、プロジェクト・メンバー以外の社内同僚との間でも、コミュニケーション回数(接触量)が多かった。しかしこれはパフォーマンスの高いプロジェクトはそれだけ研究開発に多くの時間を投入しているためであり、投入時間当たりのコミュニケーション回数とパフォーマンスとの間には有意な関係は見られなかった。

Allen は、外部とのコミュニケーションがパフォーマンスに結びつかないのは、各研究所や組織にはその組織固有の考え方や文化、あるいは用語などがあり、その違いがセマンティック・ノイズとなり、コミュニケーションを阻害するためではないかと考えた。ここでセマンティック・ノイズ(semantic noise)とは、コミュニケーションをとっている当事者間に共通概念が欠如していることが原因となって生じる「意味上の雑音」のことであり、解釈ミスを引き起こすと考えられている。しかし、研究所にとっては、外部からの情報は必要不可欠である。そこで Allen は、研究所における技術者集団のコミュニケーション・ネットワークを詳細に調べたところ、集団のなかには、集団内の誰とでも何らかの形で接触している「スター」的な人間がいることが明らかとなった。こうして、組織にはコミュニケーションのキーとなるスター的な人間、「ゲートキーパー」が存在し、彼らの外部の情報との接触頻度が他の同僚とは明らかに異なっていることがわかった。ここで、ゲートキーパー(gatekeeper)とは、直訳すれば「門番」のことであるが、経営学では、組織や企業の境界を越えて、その内部と外部を情報面からつなぎ合わせる人間のことを指す。さらに、このゲートキーパーは、一般の技術者と比べて、高度の技術専門誌を含めた読書量が圧倒的に多いということも明らかになったのである。よりわかりやすくいえば、ゲートキーパーは、組織内の誰とでも何らかの形で接触しているいわばスター的な存在であるとともに、組織外部との接触もきわめて多い人間である。

このことから、ゲートキーパーを中心とした次のような技術情報の流れが想定できる。すなわちゲートキーパーは、外部と頻繁に接触することにより、セマンティック・ノイズに煩わされることなく外部情報を獲得することができる。またゲートキーパーは、高度な専門誌の内容をよく理解していることから、そうした情報を一般の技術者に分かり易いように変換し、説明することもできる。こうして組織内の一般技術者は、ゲートキーパーを介して外部の最新の技術情報を獲得することが可能になっていたのである。

では、こうした重要な役割を果たすゲートキーパーとは具体的にどのようなプロフィールを持った人たちなのであろうか。ゲートキーパーと一般の技術者とを比較して、アレンはゲートキーパーの特徴として次の3点をあげている。

ゲートキーパーは高度の技術達成者である。

ゲートキーパーの大半は第一線の管理者である。

技術系の経営者は、ちょっと気をつければ誰がゲートキーパーであるかを正確に見分けられる。

ここで注目すべきは である。実は、アレンが詳細なコミュニケーション調査の結果ゲートキーパーであると判断した人と、マネジメントに対して誰がゲートキーパーであるかと質問した答えは90%以上一致していた。アレンおよびその後の関連研究より、ゲートキーパーの存在は研究開発のパフォーマンスにプラスの影響をあたえるという結果が得られているが、経営者が

このようなゲートキーパーに報いるためには、骨をおってコミュニケーション調査まで行う必要はなく、ちょっと組織内のコミュニケーションに敏感であればよいのである。

この研究以後、イノベーション研究は、大きく二つの流れに分かれる。一つは、ここで紹介したアレンによって始められるコミュニケーション研究である。アレンが提示した「ゲートキーパー」という概念は、その後の多くの研究者を刺激し、研究開発組織を対象としたコミュニケーション研究が多数行われた。日本でも最近、原田(1999)の研究がある。

もう一つの流れは、1980年代半ばに始まった、企業の競争力の源泉を探る視点から、イノベーション(特に製品開発)のプロセスに焦点を当てた研究である。野中郁次郎と竹内弘高は、新製品開発プロジェクトの事例分析より、スピードと柔軟さを同時に要求される製品開発においては、従来型の逐次段階的ないわゆる「リレー型」ではなく、プロジェクトのフェイズをオーバーラップさせた「ラグビー型」の方が有効であると主張した。この研究の流れを受け継ぎ、野中らと同様に企業の競争力と製品開発プロセスに焦点をあてながら、「製品開発組織のパターン」と「パフォーマンス」との関係について世界的な規模で定量的な実証分析を行ったのが、藤本隆宏とキム・クラークの研究である。藤本とクラークは、自動車産業を対象とした実証分析より、製品が持つ多様な製品属性の全体的な調和・一貫性を指すプロダクト・インテグリティ(product integrity)が重要な自動車の製品開発における有効な組織パターンの一つとして、「重量級プロダクト・マネジャー」制度を提示した。ここで重量級プロダクト・マネジャー(heavy weight product manager; HWPM)とは、製品のインテグリティを高めるために、部門間調整(内部統合)と、製品コンセプト推進(外部統合)の2つの機能を兼ね備えた、強力なリーダーをさす。HWPMは、製品コンセプト作成の責任者でありながら、生産、営業、設計現場への影響力も強い。当該プロジェクトに関しては、各機能部門長よりも強い権限をもっており、単なる調整役にとどまらず、自らが創造したコンセプトをもとに、製品全体を強力にまとめ上げる。

・ コミュニケーション競争モデル

1. モデルの概要

このモデルは次のような疑問から出発して考え出された。

組織はどのように形成されてくるのか。

どのような性質をもった組織が競争に生き残るのか。

このモデルでは、そのメカニズムを「コミュニケーション」に絞ってモデル化している。これは、ある意味でコミュニケーション自体を「餌」としている競争モデルなので、「コミュニケーション競争モデル」(communication competition model)、略して「ComComモデル」と呼ぶことにしよう。このモデルの基本的なルールは簡単である。すなわち、

エージェントは、より多くの「アイデア」とコミュニケーションできるようなポジションを求めて移動する。(このことでエージェントは単独でいるよりも、クラスターに参加することを選好する。)

複数のクラスターが存在する場合には、エージェントは、より多くの「アイデア」とコミュニケーションできるクラスターの方を選択する。(このことで、複数のクラスターが衝突した場合には、勝敗が生まれる。)

さらに、このモデルでは、「表面積」の異なる数種類のエージェントが存在する。ここで

いう表面積とは比喩的な表現で、一般的にはコミュニケーション能力の大きさを表していると考えてほしい。モデル上は、コミュニケーション能力の大きいエージェントは、表面積の大きなエージェントとして表現される。こうすることで、表面積が大きいエージェントは、より多くのエージェントと接触ができるようになる。つまりコミュニケーション能力も高くなるわけである。

こうして、エージェントの中にはコミュニケーション能力の異なるものが含まれていることで、


- (a) クラスタ間の競争にどのように影響するのか。
- (b) クラスタの大きさはどのように変化するのか。
- (c) コミュニケーションはどのように変化するのか。

という点が注目される。

2. モデルの仕様

それでは、実際にシミュレーションのモデルを作る必要がある。そのためには、まず、いくつかの概念を定義しておく必要がある。

(1)パス長 L: path length


具体的には、で示してあるような例のようにパス長を図る。パス長は次のように定義される。

同色の二つのエージェントAとBの間がすべて同色のエージェントでつながっている時、それをパスと呼び、AとBはコミュニケーション可能とする。

パス長Lは、エージェントAからBに到達するまでに経由するエージェントの個数とする。ただし、B自体も数えるので、AとBが隣接している場合でも $L=1$ となる。



いくつかパスがあった時には、そのうち最短のものをLと定義する。

(2)クラスタ値 C: clustering value

具体的には、 (A)で示してあるような計算の仕方をする。クラスタ値の定義は次のようになる。

あるエージェントにとって、コミュニケーション可能な同色のエージェントのクラスタのサイズ。

ただし、クラスタのサイズとはいっても、Cはクラスタを構成する同色のエージェントの個数の単純な合計ではない。Lが大きくなると伝達に時間がかかるため、 $1/L$ で加重した合計個数になる。

そのため、 (A)の例でもわかるように、同じクラスタに所属していても、そのクラスタ内のポジションでエージェントのCの値は変わってくることになる。 (A)では、エージェントAよりもB、さらにDの方がクラスタ値が大きくなり、一般に、エージェントのポジションがクラスタの中央に近くなるほど、そのエージェントのCの値は大きくなる。

このクラスタ値を使って、コミュニケーション競争モデルのルールを記述することが

できる。まず、各エージェントにとっては、自分のクラスター値 C の値が大きければ、一定時間の間に、それだけ多くの情報を収集できることになり、より大きな C をもつエージェントが競争優位に立つ。そこで、

各エージェントは、より大きなクラスター値 C を求めて移動する。






各エージェントが 1 期に移動できる距離は 1 に固定する。

各エージェントは距離 1 の範囲で他のエージェントをサーチする。逆に言えば、距離 2 以上離れると、エージェントは他のエージェントを感知することはできなくなる。

別の色のクラスターと接触したエージェントは、もし相手方に組した方が C の値が大きくなるのであれば乗り換え、寝返る(つまり色が変わる)。

というルールを設定する。このモデルは後の比較のために「合理モデル」と呼ばれるが、本来は の性質から、「限定合理モデル」と呼んだ方が正確である。

(3)大きなエージェント

大きなエージェントとは、例えば、先ほどの  (A) の例で、今度は、エージェント D がセル 3 個分の大きさのエージェントになったとしよう。このときクラスターを構成するエージェントの個数は変わらないものとするが、そうすると  (B) のようになる。大きなエージェントの特徴は、「表面積」が大きいことと、どんなに大きいエージェントでもパス長を測る時には 1 個と数えるということにある。したがって、 (B) に示されるようにクラスター値が変化することになる。ただし、大きなエージェントを投入したからといって、単純にクラスター内のエージェントのクラスター値が大きくなるわけではない。実際、 (A) と  (B) を比較すると、エージェント A はクラスター値が 6.5 から 7.0 に増加しているが、エージェント B の方は、逆にクラスター値が 8.0 から 7.5 に減少している。

(4)エージェントの初期配置

クラスターの形成は、シミュレーション開始時のエージェントの初期配置に大きく影響される。コミュニケーション競争モデルの場合、乱数のシード値を指定することで、エージェントの初期配置を再現することができる。乱数シード値は実行環境設定からのみ指定できるので、シミュレーション 1 回毎に変更しなければならない。実際にコンピュータ・シミュレーションを行う場合には、乱数シード値を固定してから行うので、例えば、(i)合理モデルで、(ii)すべて小エージェントを使って、(iii)乱数シード値が 3 の場合には、「合理/大 0/S3」と表すことにする。

ただし、同じ乱数シード値を使っても、すべて小エージェントの時と、大エージェントを投入したときとは、若干、初期配置が異なってしまう。例えば、全体で 20 個のエージェントのうち、2 個を大きさ 3 の大エージェントに変えると、その大エージェント 2 個を含めて 4 個のエージェントの初期配置が変わってしまう。しかし、極力、両者の初期状態を同じものにしてから比較するために、シミュレーションは乱数シード値を固定して行うことにした。

3 . シミュレーションの結果

それでは、さっそくシミュレーションを行ってみることにしよう。ComCom モデル用のプログラムは ComCom.abs として提供されている。コンピュータ・シミュレーションを行う際には、次の三つの指標が注目されることになる。

活動度: これは各ステップにおいて動いたエージェントの数を表示している。ただし、各エージェントの動ける範囲は距離 1 に限定されている。 $t+1$ 期に出力される「活動度」は t 期のマップから $t+1$ 期のマップの状態に推移する際の活動量を表示している。

合計クラスター値: これは、クラスターを構成しているすべてのエージェントのクラスター値を合計したものである。図 1 からわかるように、格子モデルでは、クラスターの角で接触しているエージェントも同じクラスターの一員として扱っている。

平均クラスター規模: クラスターを形成するエージェントの数をクラスター規模と呼ぶ。平均クラスター規模は、全エージェント数を「クラスター数」で割って求める。

ComCom.abs では、いずれの指標も数値とグラフで表示されるようになっている。

そこで、空間を、種別が「格子モデル」、大きさは 20×20 、端点処理は「ループする」に設定する。その上で

(ケース 1) 長さ 1 の赤いエージェント 10 個、青いエージェント 10 個。

(ケース 2) 赤いエージェントは長さ 1 が 8 個に長さ 3 が 2 個の計 10 個、
青いエージェントはケース 1 と同じく長さ 1 のものが 10 個。

各ケースで乱数シード値を 1~30 まで 1 ずつ増やしながら計 30 回試行し、各試行で 300 期まで各期の「平均クラスター規模」、「クラスター値の合計」を記録していくことにしよう。


(1)意外な結果をもたらした大エージェント: ゲートキーパー

当初、大エージェントが投入されると、その大エージェントを核にして、より大きなクラスターが形成されると予想していた。ひょっとすると、大きなエージェントが何個か偶然につながって、コミュニケーション・ハイウェイのようなものが形成されたクラスターが、強みを発揮して巨大に成長するかもしれないとも予想していた。当然、大エージェントを抱える赤陣営の方が青陣営に比べて競争を有利に進めるはずであった。ところが、シミュレーションの結果、次のようなことがわかった。


(a) クラスター間の競争にどのように影響するのか。

大きなエージェントを投入したことで、赤陣営が強くなったということはなかった。表 1 にも示されているように、300 期目の結果を比較すると、すべて小エージェントだけであったケース 1 の場合の、赤陣営の平均は 10.07 個とほぼ 10 であったが、赤陣営の二つの小エージェントを大エージェント二つに入れ替えると、赤陣営の平均はかえって 7.70 個に低下してしまった。平均値の差は有意でこそなかったものの($t=1.633; p=0.108$)、むしろ成績が悪くなってしまったのである。実際、表 1 からわかるように、ケース 1 では赤陣営が過半数を占めていたにもかかわらず、ケース 2 になって赤陣営に大エージェント 2 個を投入した結果、赤陣営が逆に少数派になってしまったのは 8 試行にもなる。これは、ケース 1 で赤陣営が少数派だったが、ケース 2 になって赤陣営に大エージェント 2 個を投入した結果、赤陣営が逆転して多数派になった試行数 5 を上回っている。シミュレーションの経過を観察してみると、これは、大きなエージェント自体がすぐに相手方に寝返ってしまうことが原因であった。

(b) クラスターの大きさはどのように変化するのか。

既に(a)で、大エージェントの投入は勝敗には影響しないことがわかったので、赤陣営、青陣営に関わらず、全体で見ることにして、各期における平均クラスター規模を 30 回の試行で平均して求めると、 のようになる。これによると、大エージェントを投入したケース 2の方が平均クラスター規模は早く立ち上がる傾向があることが分かった。ただし、300 期には、どちらのケースでも平均クラスター規模は 6 個強になっており、その差は縮まっている。つまり、大エージェントの投入により、より大きなクラスターが形成されるようになるのではなく、より速くクラスターが形成される傾向があることになる。

(c) コミュニケーションはどのように変化するのか。

(b)同様に、赤陣営、青陣営に関わらず、全体で見ることにして、各期における合計クラスター値を 30 回の試行で平均して求めると、 のようになる。これによると、平均クラスター規模でみたときと同様に、大エージェントを投入したケース 2の方が合計クラスター値は早く立ち上がる傾向がある。ただし、300 期には、どちらのケースでも合計クラスター値は 100 強になっており、その差は縮まっている。これは(b)の結果を反映したものと考えられる。

それでは、大エージェントを投入しても、赤陣営が強くなったというわけでもないのに(むしろ弱くなっていた)、なぜ大エージェントの投入で、より速くクラスターが形成されるようになるのであろうか。その理由は、当初予想していなかった大エージェントの意外な振る舞いにあった。大エージェントは当初の予想のようにクラスターの中心にいるのではなく、ほぼ常にクラスターの周辺部に位置していたのである。より正確に記述すれば、ケース 2の「大エージェント 2 投入」の場合、シミュレーションの最中、大エージェントはほぼ常時、クラスターの表面に露出していた。

例外的だったのは、乱数シード値=12 の試行のときで、大エージェントの周りを 9 個の小エージェントが取り囲むようにしてぐるぐる回る現象が観察されたが(この場合も、正確には大エージェントは露出している)、この場合、正確には、大エージェントを核にクラスターが形成されたというよりも、小エージェントが大エージェントの動きを封じていたと記述する方が正しい。モデルを着想した当初は、大エージェントの周りに小エージェントが群がる姿を想像していたので、これは意外であった。

この乱数シード値=12 の試行を除けば、実際のシミュレーションでは、大きなクラスターが形成されてくると、大エージェントはクラスターの中央ではなく、周りをぐるぐる回り、まるでアメーバが触手を伸ばすかのような動きをしながら周辺のエージェントをかき集めてくる役割を果たすようになる。そのために、大エージェントを投入するとクラスターの形成が早くなるのである。大エージェントはまさにクラスターのゲートキーパーとしての位置を保持し、ゲートキーパーとしての機能を果たしていたのである。

(2)均衡の実像

活動度で見ると、ケース 1 のすべて小エージェントのケースでは、30 試行のうち 7 試行(乱数シード値= 6, 8, 12, 13, 14, 18, 23)、ケース 2 の大エージェント 2 投入のケースでは、30 試行のうち 6 試行(乱数シード値= 4, 9, 18, 19, 22, 28)は、300 期になる前に、活動度が 0 になっている。つまり全 20 エージェントが活動を停止したのである。これは、エージェントが現在のポジションから移動するとかえってクラスター値が低下してしまうために、現在のポジションに留まり続けるために起こる現象で、「均衡」状態に対応している。

それ以外の試行では活動度は 0 にはならないが、その場合でもいくつかのクラスターは「均衡」に到達している。正確に言えば、「均衡」の概念はクラスターについてのもので、

すべてのクラスターが均衡状態に達したときに、全体の活動度が0になるのである。整理しやすいすべて小エージェントのクラスターについて、シミュレーションの観察結果を整理すると、**図5**のようになる。クラスター規模が、4, 6, 8, 9, 10, 13, 15, 16, 20の場合には、均衡状態となるクラスターの形状が存在し、時間をかければそのいずれかの形状に到達し、クラスターは「均衡」する。それに対して、クラスター規模が1, 2, 5のクラスターは常に形状を変えながら「ランダム・ウォーク」をすることになる。

その中間で、クラスター規模が3, 7, 11, 12のクラスターは、活動度が0になるわけではないが、ある領域の範囲内にエージェントが封じ込められ「ロックイン」の状態になる。例えば、クラスター規模3のクラスターでは、2×2の四つのセルの範囲内で三つのエージェントが移動し続ける状態が続く。クラスター規模7のクラスターでは、3×3の9個のセルの範囲内で、真中の一つのエージェントの周りを残り6個のエージェントが移動し続ける。クラスター規模11のクラスターでは、3×3の9個のエージェントの周りを残り2個のエージェントがうろつき続ける。クラスター規模12のクラスターでは、2×2の4個のエージェントの周りを、残り8個のエージェントがうろつき続ける。(クラスター規模14, 17, 18, 19のクラスターは、今回のシミュレーションでは観察できなかった。) こうして、クラスター規模によって、すべて小エージェントからなるクラスターの状態は**表2**のように分岐することになる。

表2. クラスター規模によって決まるクラスターの状態

クラスター規模	クラスターの状態
1, 2, 5	ランダム・ウォーク
3, 7, 11, 12	ロックイン(ある範囲内にエージェントが封じ込められる)
4, 6, 8, 9, 10, 13, 15, 16, 20	均衡(活動度が0になる)
(14, 17, 18, 19)	(観察できなかった)

こうして、**表3**では、ケース1のすべて小エージェントのケースについては、30試行すべてについて、300期でのクラスターの分布を整理している。この表からわかることは、ランダム・ウォークのクラスターがまだ300期でも残っている乱数シード値=2, 3, 5, 11, 24の5試行については、300期以降もさらにシミュレーションを続けると、別の状態に変化する可能性が残されているが、それ以外の15試行については、たとえ活動度が0にならなくても、すべてのクラスターが「均衡」または「ロックイン」の状態にあり、これ以上シミュレーションを続けても、その状態は変化しないことになるのである。

4. 合理モデルの均衡を超えて: 徘徊モデル

すべてのクラスターが「均衡」または「ロックイン」の状態にあり、シミュレーションをそれ以上続けても状態が変化しないことは、はたして望ましい状態なのであろうか。そのことを考えるために、これまでの合理モデルに加えて、「徘徊モデル」も考えて、合理モデルとの比較の中で、均衡が望ましいといえるのかどうかを考察してみよう。

これまでの合理モデルは、エージェントはより高いクラスター値を求めて行動するのだが、違いを明らかにするために、多少くどい表現をすると、次のようなモデルだった。

合理モデル: エージェントは、現在ポジションも含めて、移動可能なポジションの中で、もっともクラスター値の高くなるポジションに移動する。

したがって、現在ポジションでのクラスター値の方が移動可能なポジションへの移動後のクラスター値より大きいまたは等しければ移動しない。そのため、既に述べたように、クラスターが均衡することもあったわけである。それに対して、ここでは次のようなモデルを考える。

徘徊モデル: 現在ポジション以外の移動可能なポジションの中で、もっともクラスター値の高くなるポジションに移動する。つまり、移動可能な場所がある限り、必ず移動する。

徘徊モデルは、合理モデルに比べて、短期的には明らかに劣ったパフォーマンスしかもたらさないはずのモデルである。なぜなら、現在ポジションにとどまっていた方が、クラスター値が高くなるような場合でも、あえて現在のポジションを捨てて、他へと移動してしまうからである。まさに近代経済学的な均衡概念を否定した行動といえる。実際、徘徊モデルのように設定すると、いつまでもたっても活動度が下がらず(=いつまでたっても活動度が0になることはない)、均衡しなくなってしまう。そして、大きなクラスターがまるでアメーバのような動き方をするようになる。

(1)均衡かチャレンジか

「徘徊モデル」という名称は、実はエージェントの動き方を見てつけられたものである。しかし、モデル自体は唐突に思いついたわけではない。実際の日本企業の行動様式を分析する際に用いられる「変化性向」の概念から着想されたモデルである。変化性向とは、もともと日本企業のぬるま湯的体質を解明するために、高橋(1989)が最初に提唱した「体感温度仮説」において使われ始めた概念である。現状に甘んじることなく変化を求める傾向、現状を打破して変化しようとする傾向、これを変化性向と呼んだ。そして、組織メンバーが現状を打破して、変化をもたらそうとする意欲がどの程度あるのかを表す指数として「体温」を定義し、これで組織のメンバーの組織人としての変化性向を測定した。他方、お湯の温度に相当する「システム温」は、組織のシステムとしての変化性向であり、組織のシステムがメンバーの変化を受け止め、あるいは促す仕組み、制度にどの程度なっているのかを表す指数と定義した。体感温度仮説とは、この体温とシステム温との温度差で、ぬるま湯感が説明できるというもので、ぬるま湯と感ずるかどかは、組織人としての体温をベースとした体感温度の問題なのだと考えられている。既に1万人以上のデータによって体感温度仮説が成り立つことが検証されており、こうした一連の研究成果は高橋(1993; 1997)にまとめられている。

また変化性向の概念は、チャレンジの概念と密接に結び付いたもので、外発的には動機づけられていないのに、組織に貢献している「ぬるま湯」現象が日本企業で多く観察されるということは、まぎれもなく、日本企業に勤める多くの従業員が、少なくとも動機づけの場面においては、近代経済学的な期待効用原理の世界ではなく、チャレンジに重きをなす未来傾斜原理の世界に住んでいるということを示している。その点で、近代経済学的な均衡の世界観とは別の世界観を形作る概念としても変化性向は注目されるのである。

(2)徘徊モデルがもたらす高パフォーマンス

そこで合理モデルのときと同様に、徘徊モデルでシミュレーションを行ってみよう。すなわち、空間は、種別が「格子モデル」、大きさは 20×20 、端点処理は「ループする」に設定して、

(ケース1) 長さ1の赤いエージェント10個、青いエージェント10個。

(ケース 2) 赤いエージェントは長さ 1 が 8 個に長さ 3 が 2 個の計 10 個、
青いエージェントはケース 1 と同じく長さ 1 のものが 10 個。

各ケースで乱数シード値を 1~30 まで 1 ずつ増やしながらかつ 30 回試行し、各試行で 300 期まで各期の「平均クラスター規模」、「クラスター値の合計」を記録していくことにしよう。シミュレーションの結果、次のようなことがわかった。

(a) クラスター間の競争にどのように影響するのか。

合理モデル同様に、大きなエージェントを投入したことで、赤陣営が強くなったということとはなかった。既に掲げてある表 1 にも示されているように、300 期目の結果を比較すると、すべて小エージェントだけであったケース 1 の場合の、赤陣営の平均は 9.07 個であったが、赤陣営の二つの小エージェントを大エージェント二つに入れ替えると、赤陣営の平均はかえって 7.33 個に低下してしまっただ。これは合理モデルと同じ傾向である。平均値の差は有意でこそなかったものの ($t=0.720; p=0.475$)、むしろ成績が悪くなってしまったのである。実際、表 1 からわかるように、ケース 1 では赤陣営が過半数を占めていたにもかかわらず、ケース 2 になって赤陣営に大エージェント 2 個を投入した結果、赤陣営が逆に少数派になってしまったのは 7 試行にもなる。これは、ケース 1 で赤陣営が少数派だったが、ケース 2 になって赤陣営に大エージェント 2 個を投入した結果、赤陣営が逆転して多数派になった試行数 4 を上回っている。これも合理モデルと同様の傾向で、シミュレーションの経過を観察してみると、これは、大きなエージェント自体がすぐに相手方に寝返ってしまうことが原因であった。

(b) クラスターの大きさはどのように変化するのか。

既に(a)で、大エージェントの投入は勝敗には影響しないことがわかったので、赤陣営、青陣営に関わらず、全体で見ることにして、各期における平均クラスター規模を 30 回の試行で平均して求めると、図 6 のようになる。この図には比較のために合理モデルのときの図 3 を重ね合わせてあるが、合理モデルと比較して、徘徊モデルの平均クラスター規模は圧倒的に大きい。300 期には、合理モデルでは平均クラスター規模が 6 個強だったものが、徘徊モデルではケース 1 で 12 個強と 2 倍、さらにケース 2 では 20 個近くになっている。つまり徘徊モデルでは大エージェントを投入すると、初期配置にかかわらず、300 期ではほぼ一つのクラスターに集約されるようになるのである。

このように徘徊モデルは、合理モデルと比べて、圧倒的に大きなクラスターが形成されることになるが、同時に、ケース 1 とケース 2 の差も大きくなっている。しかし、これは大エージェントの投入により、より大きなクラスターが形成されるようになったのではない。合理モデルと同様に、大エージェントを投入したケース 2 の方が平均クラスター規模は早く立ち上がる傾向があるのであるが、このことは後により詳細に検討される。しかし図 6 から、小エージェントだけのケース 2 では、300 期ではまだクラスター形成の途中であり、平均クラスター規模がまだ増加傾向にあることは容易に読み取れる。

(c) コミュニケーションはどのように変化するのか。

(b) 同様に、赤陣営、青陣営に関わらず、全体で見ることにして、各期における合計クラスター値を 30 回の試行で平均して求めると、図 7 のようになる。この図には比較のために合理モデルのときの図 4 を重ね合わせてあるが、合理モデルと比較して、徘徊モデルの合計クラスター値は圧倒的に大きい。300 期には、合理モデルでは合計クラスター値が 100 強だったものが、徘徊モデルではケース 1 で 170 強、さらにケース 2 では 200 強になって

いる。つまり徘徊モデルの方が合計クラスター値が2倍にもなるのである。ただし、これも平均クラスター規模でみたときと同様に、大エージェントを投入したケース2の方が合計クラスター値は早く立ち上がる傾向がある。小エージェントだけのケース2では、300期ではまだクラスター形成の途中であり、平均クラスター規模がさらに増加していけばケース1とケース2の合計クラスター値の差は縮まることが予想される。

いずれにせよ、合理モデルに比べて、短期的には明らかに劣ったパフォーマンスしかもたらさないはずの徘徊モデルが、圧倒的に高いパフォーマンスをもたらすことになる。現在ポジションにとどまっていた方が、クラスター値が高くなるような場合でも、あえて現在のポジションを捨てて、他へと移動してしまうという、まさに近代経済学的な均衡概念を否定した行動が、高パフォーマンスをもたらすのである。徘徊モデルでは、大きなクラスターがまるでアメーバのような動き方をするようになるというのは「機械」対「有機体」という対比を連想させる。

(3)触手をもったアメーバ: 大エージェントがもたらす立ち上がりの速さ

それでは、本当に、徘徊モデルのケース1では、300期でもまだクラスター形成の途中経過しか見ていないのであろうか。そこで、合計クラスター値を使って、そのことを分析してみよう。ここで平均クラスター値を使わないのには理由がある。徘徊モデルでは、平均クラスター規模は単調に増加しないで、上下に変動しながら増加する傾向がある。これは、徘徊モデルでは、一旦はクラスターを形成したエージェントが、時々1個だけ波しぶきのように離れ、また集団にくっつく現象が起こるためである。しかしクラスターが1個になった後でも、時々2個になることがあると、実際にはクラスターが分断されたのではなく、エージェントが時々1個だけ波しぶきのように離れただけでも、平均クラスター規模は半分になってしまう。このように平均クラスター規模は徘徊モデルの場合、振れが大きく、あまり実態を反映していないことがある。それに対して、合計クラスター値は比較的安定しているので、こちらを使うのである。

そこで、今度は300期の最後の10期、すなわち291期から300期の合計クラスター値を平均して、それで30の試行をケース1、ケース2の内で、それぞれ順位づけてみることにしよう。すると図8が得られる。この図からわかるように、実は、「すべて小エージェント」のケース1でも、30試行中、上位17番目の試行までは、「大エージェント2投入」のケース2と同じ合計クラスター値なのである。ところが、ケース1の場合にはそれ以降の試行では、300期までに、まだ一つのクラスターを形成するに至っていないのである。つまり、ケース1では、300期ではまだ半分の試行で、クラスター形成の途中で打ち切られた状態で観察していることになる。実際、ケース1の上位17番目の試行までだけで、合計クラスター値の平均を計算すると、図9のようになり、「大エージェント2投入」のケース2と比べて、立ち上がりが遅いだけであることがより鮮明に分かる。おそらく、もっと時間をかければ、ケース1でも、すべての試行でクラスターは一つになるのだろうが、現在のコンピュータでは、既に計算時間が限界にきており、数千期のシミュレーションを行うことは現実的ではない。

いずれにせよ、徘徊モデルでは大きなクラスターがアメーバのような動き方をするようになるが、大エージェントはクラスターの周辺で露出しながら、時々クラスターから突き出して、まるでアメーバの触手のような役割を果たしているのである。ゲートキーパーとは、まさにそのような機能をもった人のことをさすのかもしれない。

このシミュレーションの結果は、大エージェントの投入は、クラスターの形成・立ち上げを迅速にするためには効果があることを示している。したがって、時間との競争をして

いるような研究開発においては、効果があるかもしれない。しかし、品質を造り込んでいくような品質管理を重視する実際の製造現場では、かならずしも大エージェントは必要ないかもしれない。時間はかかっても、いずれクラスター規模、合計クラスター値の指標で追いつくことになるからである。

・インプリケーションを超えるマルチ・エージェント・シミュレーション

このようにマルチ・エージェント・シミュレーションは、いくつかの示唆を与えてくれる。しかし、局所的な問題から学び、全体的な解を發明しようとする(learn from local problems and try to devise a global solution) マルチ・エージェントらしい実用的な使い方もある。Thomas & Seibel (1999; 2000)は、サウスウェスト航空(Southwest Airlines)の貨物事業(cargo business)において、Bios Group がマルチ・エージェント・シミュレーションを使って、貨物のオペレーションの改善を行い成功した事例を報告している。

サウスウェスト航空は、かつて貨物オペレーションでいくつかの問題を抱えていた。

貨物室スペース(bin space)は、満載の状態と比べて、重量的にも容量的にも、約 7%程度しか使われていなかった。

にもかかわらず、便によっては運ばなくてはならない貨物の量が多く、やりくりする貨物室スペースが不足して、飛行機の出発時間が遅れた。しかし、サウスウェスト航空は基本的に旅客航空であり、貨物収入は全体収入のわずか 2~3%にすぎず、貨物の理由で飛行機の出発遅れは許されない。

こうした問題に対して、マルチ・エージェント・シミュレーションを使って「局所的な問題から学習し、全体的な解を考案する」(learn from local problems and try to devise a global solution)ことが行われ、成功を収めたという。

より具体的に言えば、まず、運送貨物エージェント(freight agent)とランプ・エージェント(ramp agent)をエージェントとして組み込んだシミュレーション・モデルを組む。ここで、「運送貨物エージェント」は積荷目録を作成する運送貨物事務所(freight office)に対応し、「ランプ・エージェント」とは、米国の空港で飛行機の脇にいてベルトコンベアーで貨物の積み下ろしをしている作業員(ramp personnel)のことである。日本では、貨物は空港で一旦、貨物コンテナに詰め込んでから飛行機に積み込んでいるが、米国の国内線では、貨物コンテナは使われず、棚型の車両に貨物をばらばらのままで積んで飛行機の脇まで持ってきて、一個ずつ送り先を確かめながら、ベルトコンベアーと手作業で積み下ろしする。つまり、そのままバラ積みになっている。その作業をしているのがランプ・エージェントなのである。

ここで運送貨物エージェントのルールとは、基本的に積荷目録(manifest)の記載ルールのことである。したがって、「積荷目録記載戦略」(strategies for manifesting)を創発的に作り上げていく作業を、シミュレーションで全体的な効率性を確認しながら行おうというわけである。そのための手順としては、

サウスウェスト航空の(a)1998年の積荷の記録(重量、個数、出発地と目的地、取扱サービスのクラスといった情報を含む)、(b)飛行計画、(c)運送日誌(貨物よりも優先される郵便や手荷物を除いて実際に利用可能な貨物室スペースを示している)といった実際のデータを使って、現在の貨物オペレーションをシミュレーションで再現する(reproduce)。運送貨物エージェントのルールを、現場から提起される問題点や提案を取り入れて書き

直してみても、そのルールをもとにコンピュータが作り出した(computer-generated)積荷目録とランプ・オペレーションを使って、シミュレーションを行ってデータを集める。

という作業を繰り返して、全体的な効率の点で、より良いルールを求めていくのである。

このうちの問題点と提案については、たとえば、貨物を扱うステーションの中には、実際には、取扱量の多いステーション(heavily used station)と取扱量の少ないステーション(lightly loaded station)がある。そのために前者では、貨物の積み下ろしに時間がかかり過ぎて出発時刻が遅れたり、貨物が一夜を過ごすことも出てくる(overnighting)。こうして考えられたのが same-plane strategy である。これは、最短経路にこだわらず、回り道をしてもいいから、なるべく積み替えなくても済むように同じ飛行機に載せっ放しにしておくルールである。貨物を最短経路で運ばないと、コスト的に割高になるという心配もあるが、実際には、貨物室が空であろうと満杯であろうと飛行機を飛ばすコストに大差はなく、ほとんど貨物室が空であっても飛行機は飛ばさなくてはならないので、貨物を回り道させても、コスト的には問題にならない。問題は貨物の積み替えの時間の方である。

この same-plane strategy を使ってシミュレーションを行うと、例えば、ボトルネックになっていたロサンゼルスからフェニックスへの便は、貨物量が従来は1万ポンド運送しなければならなかったものが、2000ポンドだけで済むようになり、ボトルネックが解消された。こうした局所的な問題解決を取り入れて、個々のエージェントのルールを改良していき、サウスウェスト航空の会社全体のシミュレーションを行うと、次のような measure の改善が見られた。

- (a) 各ステーションで取り扱われる貨物量(積み替えも含む): 例えばフェニックスでの積み替え貨物量は週16万ポンドから5万ポンドへと約1/3に激減する。全体での取扱貨物量は325万ポンドから250万ポンドに減少する。
- (b) 一夜を過ごして保管される貨物量(安全上の理由で、夜間は飛行機から倉庫に移して鍵をかけて保管し、翌朝また飛行機に戻される): システム全体で週24万ポンドあったものが5万ポンドへと約1/4に激減する。このことで、料金的にかなり高い空港施設を使うことになる貨物保管のコストを削減できる。
- (c) 各荷物のサービス水準(例えば、NFG: Next Flight Guaranteed や Rush つまり24時間サービスなど)が達成されたかどうか: Rush については達成率がやや落ちるが、NFG については達成率が改善する。

以上が、サウスウェスト航空の事例である。こうしたシミュレーションの手法を使いながら試行錯誤で出された解の最適性が保証されないことは明らかである。しかし、現実のエージェントのルールは複雑すぎて、数学的に定式化したり解いたりすることは難しいし、さらに重要なことは、最適かどうかを議論する前提になるべき代替案もしくは選択肢が、われわれには常にそのごく一部しかわかっていないという現実である。だからこそ、局所的な視点からそうした代替案を学習していくプロセスが重要になるのである。その意味では、問題を可能な限り単純化することで最初から全体最適を求めるというアプローチを放棄し、全体的な効率を少しでも改善するような代替案を局所的に探索することを選んだともいえる。その局所的な代替案が、全体的な効率にどのような影響を及ぼすのかを調べるためには、マルチ・エージェント型のシミュレーション・モデルは有用である。

そして、こうしたマルチ・エージェント型のシミュレーション・モデルの特徴は、なんといっても個々のエージェントのルールをある程度共通に決めてしまい、あとはそうしたエージェントの総体の全体的な振る舞いに着目するという点にある。つまり、局所的な

「知恵」を、すべてのエージェントのルールに反映することが可能なのである。それに対して、システム・ダイナミクスなどでは、要素間の関係・結びつきが変化することで、システム全体の振る舞いがどのように変化するかを調べるのであって、個々の要素のルールを変化させることは、そもそも分析の対象ではない。局所的な問題から学び、全体的な解を発明しようとする(learn from local problems and try to devise a global solution)とき、マルチ・エージェント型のシミュレーションには他のシミュレーション・モデルにはない可能性があるといえる。

付録: ComCom.abs について

本稿で取り上げたルールの他にも、ComCom.abs では次の 3 種類のルールを用意している。

方向性ルール「ルール 1」:各エージェントは、同じ大きさの C をもたらすポジションがいくつかある時には、できるだけ右に(右・右上・右下に)移動する。

凝集性ルール「ルール 2」:一つのクラスターを構成するすべてのエージェントは同じ方向に移動する。その際の「方向」はクラスター全体の多数意見に従う。

ランダム・ルール「ルール 3」:各エージェントは、ランダムに移動する。

本文中で取り上げているのは「基本ルール」と呼んでいるもので、これは ComCom.abs では「ルール 0」に対応している。これらのルールは、ある意味で利己的な「基本ルール」に対して、エージェントが迷ったときには、リーダーの指し示す方向・方針・戦略に従うという「方向性ルール」、あるいは、特に事情がない限りは(つまり他のクラスターと接触しない限りは)、所属するクラスターの全体の多数意見に従って、同じ方向に移動する「凝集性ルール」という組織論的な意味づけが考えられている。

こうした背景から、ComCom.abs では、基本ルールを含めて、方向性ルール、凝集性ルール、ランダム・ルールの四つのルールから二つを選んで、それぞれをもっているエージェントの間で対戦もしくは競争させることもできる。さらに、ルールを途中で変えられるので、どのようなルールの組み合わせが望ましいのかを試行錯誤しながら考えることができる。実は、どのような組織戦略が良いのか、これまでのところあまり知られていない。現場の知恵を生かす形で、創発的な組織戦略をシミュレーションで色々と実験しながら生み出していけば面白い。いままでのところ、とりあえず大きなクラスターを形成するのが目的ならば、凝集性ルールで、できるだけ多くの周辺のエージェントをかき集めて、大きな、しかし中身の詰まっていない骨格だけのようなクラスターを作り、それから基本ルールでクラスターの引き締めにとりかかるというのが、強い組織作りには向いているような感触が得られている。

ただし「情報出力画面」の読み方には注意がいる。 t 期の「エージェント・マップ」の情報は $t+1$ 期の「情報出力画面」に出力されているからである。これは、情報表示を World の Agt_Step に書いているためで、実行順番は、World 各エージェントルール 再描画、となっており、情報表示のための値を計算した後、二次元マップを表示するまでにエージェントが動くので二次元マップのみ $t+1$ 期の情報が表示されることになる。二次元マップと他のグラフ表示は別に考えた方が誤解を招かないかもしれない。

また、ComCom.abs は、現在のパソコンの処理スピードと比較すると、まだ重たいプログラムである。リアルタイムで観察していると動きが鈍くなる。そこで、一旦はログ・ファイルを作成させて、後でそれを再生する方法をお勧めする。ログ・ファイルとは計算結果

を格納するもので、これを使えば、速くシミュレーションの結果を再生させることができる。ComCom.abs はログ・ファイルを出力することを前提としているので、シミュレーションを実行する前に、次のような設定が必要になる。

ログ・ファイルを格納するためのフォルダを作成する(名称は自由)。

ABS を起動する。

[設定]-[実行環境設定]さらに[ログファイル][設定]で のフォルダを指定する。

シミュレーションを実行する。

ログ・ファイルを再生するには、[エクスプローラ]-[フォルダ]で のフォルダを開いてからファイルをクリックすると ABS が起動するので、ABS のウィンドウで[再生]-[再生]。あるいは、ABS のウィンドウで[ファイル]でファイルをクリックしてから、[再生]-[再生]。

参考文献

- Allen, Thomas J. (1977) *Managing the Flow of Technology: Technology Transfer and the Dissemination of Technological Information within the R&D Organization*. MIT Press, Cambridge, Mass. (中村信夫訳 『”技術の流れ”管理法』 開発社, 1984)
- Epstein, Joshua M. & Robert Axtell (1996) *Growing Artificial Societies: Social Science from the Bottom up*. Brookings Institution Press, Washington, D.C. (服部正太・木村香代子訳 『人工社会 複雑系とマルチエージェント・シミュレーション』 構造計画研究所/ 共立出版, 1999)
- 藤本隆宏 = キム・クラーク(1991)田村明比古訳 『製品開発力』 ダイヤモンド社.
- 桑嶋健一(2000)「ゲートキーパー」高橋伸夫編著 『超企業・組織論』 有斐閣.
- 桑嶋健一・高橋伸夫・玉田正樹(2000)「コミュニケーション競争モデル」日本行動計量学会第 28 回大会論文. 抄録集(近刊).
- 桑嶋健一・高橋伸夫(2000)「グローバル研究開発とコミュニケーション能力 製薬会社を例として」国際ビジネス研究学会第 7 回全国大会. 報告要旨(近刊).
- 原田勉(1999) 『知識転換の経営学』 東洋経済新報社.
- Myers S. & D. G. Marquis (1969) *Successful Industrial Innovations*, National Science Foundation.
- 高橋伸夫 (1989)「日本企業のぬるま湯的体質」『行動計量学』 16(2), 1-12.
- 高橋伸夫 (1993) 『ぬるま湯的経営の研究』 東洋経済新報社.
- 高橋伸夫 (1997) 『日本企業の意思決定原理』 東京大学出版会.
- Thomas, Charles R., Jr. & Fred Seibel (1999) “Adaptive cargo at Southwest Airlines,” The 4th Annual Colloquium on the Application of Complex Adaptive Systems to Business (Ernst & Young), July 25-27, 1999, The Cambridge Center Marriott, Mass.
- Thomas, Chuck R., Jr. & Fred Seibel (2000) “Adaptive cargo routing at Southwest Airlines,” *Embracing Complexity: A Summary of 1999 Colloquium on the Application of Complex Adaptive Systems to Business*. Ernst & Young, Cambridge, Mass., pp.73-80.

ワーキングペーパー・シリーズ

Working Paper Series

No. 9 (最新号)

高橋伸夫・桑嶋健一・玉田正樹 コミュニケーション競争モデル

No. 8

服部正太・木村香代子・西山直樹

ターミナル内における移動シミュレーション

No. 7

板山 真弓・田村 誠 Schelling 分居モデルを超えて 3

No. 6

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

No. 4

山本 和也 森林火災の拡大と樹木の密度

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

No. 2

板山 真弓・田村 誠 Schelling 分居モデルを超えて

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