3E1-06

エージェントと非ノイマン型プログラム

Agents and non-Neumann type programs

間遠伸一郎

Shinichiro Mado

[*1] 宇都宮大学

Utsunomiya University

Non-Neumann type program is a new term we defined here. Although similar word “Non-Neumann type computer” is familiar to computer scientists, the word “non-Neumann type program” is at all new to them. And we also defined here the word “Neumann type program” which is the opposite category to the notion of “Neumann type program”. There are some important relations between non-Neumann type programs and Agents. Agents are different from simple automata in the point that those have not only interior states but also interior programs. The interior programs of agents are a kind of non-Neumann type programs which can be written in IF-THEN rules. Non-Neumann type programs can be a kind of logic programs because those are Boolean functions. Boolean functions can also be represented by IF-THEN rules. Non-Neumann type programs and IF-THEN rules have very important relations with agent based models of complex systems.

1. 問題の所在

本論は，エージェントシミュレーション Agent Based Simulation におけるエージェント Agent の設計方法として，エージェントの内部プログラムを非ノイマン型プログラムという観点から統一的に理解する方法を提示し，それが従来 IF-THEN ルールとして知られる形式で記述できること，したがって，IF-THEN ルールには，ある意味の一般性があることを明らかにする。このことは，複雑系科学の方法としてエージェントシミュレーションを活用する際にとりわけ重要性をもつ。

20世紀の最後の20年間は，国際関係の激動によって，経済や社会が複雑系であり，その制御は複雑系の制御という大変難しい一般的問題の特殊ケースであり,簡単には解決できない難問であるという認識をもたらした。

複雑系研究は，もともと生命や人間の精神のような，従来の科学の射程を超えた神秘に科学的解明の足がかりを提供する研究として，20世紀の半ばごろより始まった。複雑系研究の裏付けは情報科学によってなされた。つまり，情報科学の発達が複雑系研究を可能にしたのであって，情報の概念が明確にされ，情報科学の基礎が確立されるのが，複雑系研究の始まる20世紀半ばであることは偶然ではない。複雑系を解明するための鍵となる概念が情報科学的に定式化されたオートマトンの概念である。

複雑系の研究にエージェントシミュレーションが有効であることは，多くの研究者によって認められている。私の固有の研究分野は経済学だが，経済学もまた複雑系科学として生まれ変わることを求められているので，経済学においても，エージェントシミュレーションが期待されている。

しかし，エージェントシミュレーションのエージェントをどのように設計するかについては，確立された方法はまだない。

本論では，IF-THEN ルールというひとつの方法について論じる。この方法を，非ノイマン型プログラムという着想にしたがって見直すことによって一般性をもった見通しの良い方法が得られることを示す。

IF-THEN ルールのような単純な構造の集積が非ノイマン型プログラムとしての一般性を持つことは，自然発生的に形成される複雑系の研究に重要な足がかりを与えるものと思われる。

このように，非ノイマン型プログラムという概念は，大変重要な概念である。

連絡先：宇都宮大学,国際学部，madoshin@utsunomiya-u.ac.jp

2. これまでの研究

2.1 ノイマン型プログラム

オートマトンを一般的な形で科学の対象として研究し始めたのはフォンノイマンである。従来の数学は関数概念を中心としていた。オートマトンは関数概念を基礎としながらも，関数概念とは本質的に異なるものである。オートマトンは数学の対象として本質的に新しい要素を含んでいた。オートマトンによってはじめてプログラムというものを考えることが可能となった。

オートマトンを発明したのはノイマンではない。チューリングのチューリング機械はノイマンに先行するオートマトンの研究である。しかし，より一般性のある数学上の概念としてのオートマトンを明確に取り上げたのはノイマンが最初である。

ノイマンはまた，ノイマン型計算機またはプログラムストアド方式と呼ばれる電子計算機の基本設計を考案したことでも知られる。この方式は，チューリング機械のテープの部分を電子的なメモリーで実現し，そこにプログラムとデータを格納するもので，チューリング機械を有限化して電子計算機上に実現したものと考えられる。この方式の計算機をノイマン型計算機，チューリング機械のテープにあたるメモリーを主記憶と呼ぶ。

通常，計算機のプログラムと言えば，このノイマン型計算機の主記憶に格納されたプログラムを指す。筆者は，このようなタイプのプログラムをノイマン型プログラム Neumann type program と命名する。

2.2 非ノイマン型プログラム

それに対して，本論では非ノイマン型プログラム non-Neumann type program と呼ぶべきものを考察する。筆者が非ノ

イマン型プログラムという言葉を最初に使用したのは[浅利・間遠 2003 年]である。このアイデアにたどり着いたきっかけは，[塩沢 1990 年]で述べられているチューリング機械を用いた新しいタイプのモデルで,チューリング機械の解釈の仕方が奇妙なことに気づいたことであった。塩沢は，チューリング機械のテープを環境に割り当てて，ダニが環境を読みながら行動するというAgent モデルを考案している。しかし，チューリング機械のテープは，そこにプログラムやデータを記憶しておくべきものであり，プログラムは特定のアルゴリズムを実現するように周到に作られたものである。それに対して，環境から与えられる信号は偶然性を伴うものであり，もとより，環境は何のアルゴリズムも意図してはいない。塩沢の「チューリング機械」は本来のチューリング機械の解釈とは著しく異なる。しかも，塩沢は，プログラムの格納場所を環境に割り当ててしまった後の「チューリング機械」に，まだ「プログラム」があると言うのである。この「チューリング機械」には，テープが 2 本あると解釈することもできるだろう。そうすれば，一本のテープを環境に割り当てても，残りのもう一本のテープにプログラムを格納できる。実際，非ノイマン型計算機の中には，それに近い構造のものもある。しかし，この構造では，2 本目のテープに格納されているプログラムは小さなノイマン型プログラムということになるだろう。非ノイマン型プログラムという考え方には至らないのである。

非ノイマン型プログラムにつながるヒントはダニエル・ヒリスの超並列計算機 CM-1 の設計にある。ヒリスの CM-1 は，それぞれにテープが 1 本ずつあるチューリング機械が多数ネットワークで結合されたような構造で，そのうえ,全体に対してもう一本のテープが用意されているような構造である。この点で，ヒリスの CM-1 にも非ノイマン型プログラムは存在しないのであるが，CM-1 のプロセッシングユニットには，算術論理演算装置に組み込まれているブール関数に最大限の自由度が与えられており，命令によってどのブール関数を用いるかを指定できるようになっている。ただ，残念なことに，このブール関数の指定を，個々のプロセッシングユニットに組み込まれた小さなメモリーに書き込まれたプログラムで行うことができない。しかし，ここに重要なヒントが隠されている。つまり，このブール関数の指定をプロセッシングユニット内部のメモリーで行えば，そのメモリーに格納されているものは一種のプログラムであるということである。このプログラムは手続き型プログラムとは異なる。一種の宣言型プログラムだと考えられる。

2.3 オートマトンネットワーク

塩沢の「チューリング機械」は経済モデルに用いられるべきものであるが，この書[塩沢 1990 年]の中ではダニのモデルや人間行動のモデルが例として考察されていても,経済モデルの例は明示的には挙げられていない。塩沢の研究を継承しオートマトン（チューリング機械もオートマトンの一種）で経済モデルを構築しそれを計算機上で実現したのは[浅利・間遠 2001 年]である。浅利・間遠は，その際に，オートマトンネットワークというモデルを考案しているが，これは，塩沢[塩沢 1997 年]による「オートマタネットワーク」と基本的に同じものである。[浅利・間遠 2001 年]の「オートマトンネットワーク」は[塩沢 1997 年]の「オートマタネットワーク」の具体化の一つと考えられる。[Boccara 2004 年]では，オートマトンネットワークがセルオートマトンを含む Agent Based Model の典型的なものとして扱われている。[Garzon 1995 年]では，オートマトンネットワークは，セルオートマトンやニューラルネットワークなどの超並列モデル massively parallel model の一種として扱われている。この意味では，超並列モデルと agent based model やオートマトンネットワークは同種の議論と見ることが出来る。

2.4 エージェントベースシミュレーション

オートマトンネットワークは，超並列計算機や非ノイマン型プログラムと結びつく必然性を持っている。計算機上に実現されたオートマトンネットワークは，エージェントベースシミュレーションまたはマルチエージェントシミュレーションと呼ばれることがある。

エージェントベースシミュレーションでは，エージェントの環境が自然環境なのか社会環境なのか（あるいはその両方）で違いが生じる。[塩沢 1990 年]では，エージェントの環境は自然環境であるが，[塩沢 1997 年]では，エージェントの環境は社会環境になっている。オートマトンネットワークによるマルチエージェントシミュレーションは本来，社会環境に適した表現手法だと考えられる。[間遠 1998 年] では，エージェントシミュレーションにおけるエージェントの環境が社会環境であることが述べられている。

2.5 定常過程

経済モデルに限らず，従来のモデルは関数によって表現され，関数グラフの交点あるいは連立方程式の解で均衡を表現し，その均衡がモデルの主役を演じていた。それに対して，オートマトンで主体を表現し，多数のオートマトンの相互作用によって現象を表現するエージェントベースシミュレーションでは，均衡に替わって定常過程が主役を演じる。定常過程は，均衡のように静止した一点ではなく，常に変化する一連の状態であり，動的なシミュレーションによってしか捉えられないものである。塩沢を継承した経済モデルにおける定常過程の一例が [浅利・間遠 2002 年]に述べられている。

2.6 揺らぎを伴う定常過程

塩沢の定常過程は正確に繰り返される単なる定常過程ではなく，揺らぎを伴う，すなわち,毎回の繰り返しに一定でない差が生じる，広義の定常過程である。オートマトンネットワークによるエージェントベースシミュレーションの定常過程が揺らぎを伴うということは，オートマトンネットワークの中でのオートマトン同士の相互作用の複雑さを反映していると考えられる。言い換えれば，システムの定常過程の揺らぎはシステムの複雑さの反映である。定常過程が正確な繰り返しではなく，揺らぎを伴うということは複雑系では普通の現象である。したがって，複雑系では，定常過程を，揺らぎを許容する広義の定常過程として理解する必要があるのである。

3. エージェントの基本構造

エージェントを設計する際に，エージェントを，オートマトンを基礎として設計することが重要である。

つまり，オートマトンに内部プログラムを付加して拡張したものがエージェントである。

また，オートマトンはブール関数に隠れたフィードバックループを付加して拡張したものであるから，エージェントの設計の基本はブール関数の基礎であるブール代数から説く必要がある。

3.1 Partial Ordered Set

集合 L 上の関係 $\leq$ が次の性質を満たすとき L は半順序集合であると言い，構造 $(L,\leq)$ を半順序構造と呼ぶ。

3.2 Lattice

$(L,\leq)$を半順序構造とするとき，join $\vee$ および meet $\wedge$ を次のように定める。

(1) Join

$a \in L, b \in L$ に対して $a \vee b = LUB(a,b)$

ただし，$LUB(a,b)$ は $a \leq c$ かつ $b \leq c$ であり，$a \leq x \leq c$ または $b \leq x \leq c$ となる $x \in L$ が存在しないような，$c \in L$ のことである。

(2) Meet

$a \in L, b \in L$ に対して $a \wedge b = GUB(a,b)$

ただし，$GLB(a,b)$ は $c \leq a$ かつ $c \leq b$ であり，$c \leq x \leq a$ または $c \leq x \leq b$ となる $x \in L$ が存在しないような，$c \in L$ のことである。

3.3 Boolean Algebra

0 と 1 だけからなる集合 $B = \{0,1\}$ の要素を各桁とする，長さの n の記号列に join と meet と complement がつぎのようにして定められるとき，このような記号列の全体を B_n であらわす。いずれかの自然数 n に対して B_n と同型となる有限束 L をブール代数 Boolean Algebra と呼ぶ。

(1) Join

$$c_k = \max(a_k, b_k), k = 1, \cdots, n$$

(2) Meet

$$c_k = \min(a_k, b_k), k = 1, \cdots, n$$

(3) Complement

$$c_k = \begin{cases} 1 & if\ a_k = 0 \\ 0 & if\ a_k = 1 \end{cases}, \quad k = 1, \cdots, n$$

3.4 Boolean Polynomial と Boolean Function

ブール関数とはブール多項式で表現される関数である。

3.5 Automaton

オートマトンはブール関数に内部状態を介したフィードバックループを加えたものである。

3.6 オートマトンと意味・作用

オートマトンの状態変化には作用を連動させることができる。特定の入力信号と内部状態からの隠れた入力の組に対して，出力信号（すなわち次の状態）と共に特定の作用を連動させることによって，状態変化に意味を与えることができるのである。例えば，チューリング機械のテープを左右に動かす操作を，それが指令される時の状態変化の意味または作用と考えることができる。[塩沢 1990 年]では，チューリング機械を表現する 4 つ組み $qSS'q'$ において，入力信号 S に対する状態変化 $q \to q'$ の意味または作用を S' で表している。

3.7 Agent と Automaton

エージェントはオートマトンに内部プログラムを加えたものである。

3.8 Automaton と内部状態

オートマトンは内部状態を持つと共に，その内部状態からの隠れた入力を持つ点でブール関数とは区別される。しかし，オートマトンは内部にブール関数を持っているのだから，ブール関数を基礎として成り立つ概念だとも言える。

3.9 CD 変換と IF-THEN ルール

[塩沢 1990 年]で述べられているチューリング機械を表現する 4 つ組み $qSS'q'$ の一つ一つがそれぞれ一つの意味または作用を付加した IF-THEN ルールを表現している。この IF-THEN ルールは認知科学でいう **CD 変換**に当たるものだと理解することもできる。つまり，IF-THEN ルールの前提 qS を満たすかどうかが C すなわち**認知 Cognitive** であり，その結果状態が $q \to q'$ のように変化することがこの IF-THEN ルールの示すところであるが，同時にその状態変化の意味として作用 S' が引き起こされる。この作用 S' が D すなわち**指令 Directive** である。IF-THEN ルールは，その意味まで考えに入れると，C を D に変換する規則でもあるので，CD 変換に当たると考えることができる。

3.10 Agent と内部プログラム

エージェントの内部プログラムは，エージェントの行動規則を決めるものである。

チューリング機械では，オートマトンの行動規則はオートマトンの内部プログラムではなく外部プログラムとして与えられる。と考えることもできるが，この内部プログラムが可変となり，初期設定や学習によって書き換わるものである点が重要である。

3.11 Agent の内部プログラムと IF-THEN ルール

エージェントの内部プログラムはブール関数であるので，これを IF-THEN ルールで記述することができる。

3.12 Agent の内部プログラムと非ノイマン型プログラム

エージェントの内部プログラムが非ノイマン型プログラムに他ならない。

3.13 非ノイマン型プログラムと IF-THEN ルール

エージェントの内部プログラムは，IF-THEN ルールで注目できるが，これはこの種の IF-THEN ルールが非ノイマン型プログラムであることを意味している。

4. 結論

エージェントは，内部状態だけでなく，内部プログラムを持つことによって単純なオートマトンとは区別される。エージェントの内部プログラムは非ノイマン型プログラムであり，IF-THEN ルールで記述できる。

このことの重要性は，IF-THEN ルールの基本構造が簡単なので，IF-THEN ルールが自然発生的に出来やすいため，自然界にしばしば見ることができるということである。つまり，人間の精神を構成するニューロンのネットワークも IF-THEN ルールの集積であるし，生命を司る DNA の二十螺旋も IF-THEN ルールの集積である。

非ノイマン型プログラムと IF-THEN ルールを研究することが，精神と生命という 2 大神秘を科学的に解明するために重要な役割を果たすことが期待される。

精神も生命もいわゆる複雑系の典型例である。複雑系の研究にとって，非ノイマン型プログラムと IF-THEN ルールが本質的で重要であることがわかる。

参考文献

[Hillis・Kitsuregawa 1990 年] ダニエル・ヒリス著，喜連川優著訳:『コネクションマシン・・・65,536 台のプロセッサから構成される超並列コンピュータ』, パーソナルメディア ,1990 年.

[塩沢 1990 年] 塩沢由典:『市場の秩序学・・・反均衡から複雑系へ』, 筑摩書房 ,1990 年.

[Garzon 1995 年]Garzon, Max : ***Models of massive parallelism : analysis of cellular automata and neural networks***, Springer-Verlag , 1995 年.

[塩沢 1997 年] 塩沢由典:『複雑さの帰結』, NTT 出版 , 1997 年.

[間遠 1998 年] 間遠伸一郎:「経済システムの階層性と複雑さ」,『社会・経済システム』第 17 号, pp.15-20, 社会・経済システム学会 ,1998 年.

[浅利・間遠 2001 年] 浅利一郎・間遠伸一郎:「オートマトンネットワークによる市場の自己組織化の計算機シミュレーション」,『経済研究』5 巻 4 号,pp.1-49, 静岡大学法経学会 ,2001 年.

[浅利・間遠 2002 年] 浅利一郎・間遠伸一郎:「自己組織化としての市場システムの形成---オートマトンネットワーク理論の応用とシミュレーション---」,『社会・経済システム』第 21・22 合併号,pp.152-160, 社会・経済システム学会 ,2002 年.

[浅利・間遠 2003 年] 浅利一郎・間遠伸一郎:「限定合理性と Agent Based Modeling の方法」,『経済研究』7 巻 3・4 号,pp.239-271,静岡大学法経学会 , 2003 年.

[Boccara 2004 年] Boccara, Nino: ***Modeling complex systems***, Springer-Verlag , 2004 年.

[Kolman 2004 年] Kolman, Busby, Ross: ***Discrete Mathematical Structures (5th ed.)***, Pearson Education , 2004 年.