

シミュレーションモデルのアート性と標準化

相澤 りえ子

1. はじめに

以前、シミュレーションモデルはアートだと教科書の前書きに書いておられた先生がいらした。確かにシミュレーションモデルを構築すると、作り手によってモデルが違うことが多い。同じ主題でも作る人の感性が出てきてしまうシミュレーションモデルは工芸なのだろう。工芸とは経験と勘と感性の生産物である。しかし、シミュレーションモデルは、表現、形式、構造が違っていても、目的やシステム条件が明確に定義されており、その上で作られたモデルであれば、そのモデルを使ってシミュレーションした結果は、統計的な見方からすると同じ結果になるはずであるし、そうでなくてはならない。モデルはアートであっても良いが、分析は科学でなくてはならないからだ。

シミュレーション分析をビジネスとしている身としては、対象システムを分析する時にチームで同じモデルを使うこともあれば、システムを部分に分割して何人かが部分的なモデルを作り連結して全体モデルを構築することもある。また、出来上がったモデルをクライアントに説明をする必要がある手前、モデル化方法として、いつもある一定の規則が存在する方が都合が良いし、モデル化の生産性も上がる。一方、シミュレーション分析の過程において、モデルを構築すること自体が、対象システムを知り抽象化することで真の構造を理解するという意味で、最もOR的な工程であることは言うまでもない。そこで、シミュレーションのモデル化を行う時には、常にアート性と標準化の両面から考えることになる。

本稿では、モデルに関しての話題として、モデルの代替性、モデルの自動生成、エージェントベースモデルとシステムシミュレーションモデルの違いについて考察することにより、モデル化の標準化可能な側面と、

アートの側面、言い換えると標準化できない側面について述べる。

2. モデルの代替性について

モデルと一言で言っても、モデル化プロセスから見ると、一般的に、概念モデル (conceptual model) とプログラミング (programming) モデルがあると言われているが、森戸[1]によると、さらに二つのプロセスの間にもう一つのプロセスがあり、それをオペレーショナル・モデル (operational model) と呼んでいる。

概念モデルとは、モデルの境界、システムをどの程度細部まで見るかという詳細度、システムの動きのロジックの定義であり、まさにモデルの仕様となるものである。この概念モデルは、人によって解釈が異なることがないように正確に定義されているべきである。

プログラミングモデルは、シミュレーション専用言語を用いて、概念モデルを実行させるためのプログラムを作ることである。

プログラミングモデルは、概念モデルの構成や機能を表示している。しかし、概念モデルを理解したからといって、どの様にプログラミングモデルに変換するか、言い換えれば、いかなる形で離散型シミュレーション (ここでのシミュレーションは待ち行列モデルを記述する離散型シミュレーションである) の枠組みに落とせるか、が読み取れないことが多い。そのような場合に、概念モデルとプログラミングモデルの間の橋渡しをするオペレーショナルモデルが存在する。森戸は文献[1]の中で、この中間のモデルの事を、無味乾燥な仕様を離散型シミュレーションの枠組みの中で運用可能 (operational) な形にするという意味で、オペレーショナルモデルと定義している。

まさに、筆者が本稿で述べているモデルはこのオペレーショナルモデルのことである。同じ動きをするシステムでも、捉える人によって離散シミュレーションモデルの枠組みが違って来るからである。

あいざわ りえ子

(株)構造計画研究所

〒164-0012 中野区本町 4-38-13

例えば工場を運営するのにいくつもの部署が絡んでいるが、生産計画を立てる部署では、製品の工程図を作り、いくつの半製品がどのような順番で、どの機械で加工され、いつ完成品になるかをスケジュールする業務を行っていたとする。それに対し制御に関わる部門では、各加工機械にどのような製品が投入されたならどのような処理をして、次にどのバッファに送り出すかといった指示書を作成する業務を行っていたとする。このどちらの場合でも、設備改善計画を立てる際には、スループットを最大化させたい、機械の手待ちを少なくしたい、バッファの仕掛量を少なくしたい等といった目的で、どのような設備にしたら良いかをシミュレーション分析することになるだろう。概念モデルはシステム仕様として存在するし、目的とその詳細度も一意に決まる。しかし、オペレーショナルモデルは、生産計画を作る人と作業指示を作る人とで違ったものを作るのではないと思われる。少なくとも、各々の部署の人が分かりやすいオペレーショナルモデルは違う。

前者の部署の人は、このモデルを「もの中心モデル」として捉え、システムの中を動き回るもの（この場合は半製品）を中心に、半製品がどの機械で加工を行っていくかの順路を書き下し、さらにこのシステムに存在するリソースとその前の待ち行列を機械とバッファとして整理することになる。

後者の部署の人は、このモデルを「事象中心モデル」と捉え、システムに存在する機械の処理手順を表にまとめるだろう。その表とは、A という機械は、何工程目の a 製品が投入された場合にはどのような処理をし、処理が終わったならどのような処理をするかを定義するものである。

この違いは、プログラミングモデルにおいて用いるシミュレーション言語（またはシミュレータ）によっても決まってくる。現在市販されている生産・物流専用シミュレータの中には、工程図を書いてモデル化するものや、機械制御ルールを記述しモデル化する言語があるので、そのどちらを使うかは、何を基準にしてモデル化すると自分や仲間が分かりやすいかで決める、つまり好みで決めることになろう。

人間の嗜好でモデル化を決めるところは一見アートのようだが、そこには離散シミュレーションの枠組みが規則として存在する。

オペレーショナルモデル構築プロセスとしては、最初に、モデルを「もの中心モデル」として表現するか、「事象中心モデル」として表現するか、またはその併

用かを定める。この表現の選択は、モデルの世界観 (world-view) の決定と呼ばれている[2]。

「もの中心モデル」の場合は、まず、何をシステムの中を動き回る「もの」と捉えるかを定めることが必要である。これをモデルのオリエンテーション (orientation) の決定と呼んでいる。大きなシステムになると、数多くの要素が複雑に絡み合っており、機能が成り立っている。その場合、どの要素をシステムの中を動くものとして捉え、どの要素をリソースやバッファといった他のものとして捉えるかを定めることが、そのモデル化の鍵となる。実はこのオリエンテーションの決定においても、モデル化の岐路がある。中を流れる「もの」を、製品と捉えるか、機械と捉えるか、作業者と捉えるか等により、モデルが違ってくる。このどれを「もの」として捉えるかは、モデルを作る技術者にゆだねられている。そして、モデルの作りやすさ、モデルの分かりやすさ（他人が見る時の分かりやすさ）、その所属グループ内での常識等により、決まってくるものであるが、文献[1]では、その決定に際しての一つの方法を提示してくれているので引用させていただく。

2次元の表で行列を作り、これを参考にしながらオリエンテーションを定めると良い。行列の行は、システムに存在するアクティビティを、列には、システム内に存在するオブジェクト、すなわち役者を配し、アクティビティ i の実行にオブジェクト j が関与するときには (i, j) 要素を実行に必要なオブジェクトの数とする。関与しなければ0とする。一般に、

- ① シミュレーションの実行に当たって、各アクティビティに関与するオブジェクトの少なくとも一つがプログラム中の動き回るものとして表現されないと、シミュレーションは実行できない。
- ② あるアクティビティに対して、二つ以上のオブジェクトを動き回るものとする、同時発生事象のリスクやプログラムの長大化の可能性がある。

等の性質がある。

例として、単一窓口システムのオリエンテーション決定のための行列を表1に示す。

「事象中心モデル」の場合は、まず、システムの中での状態変化とは何であるかを洗い出す。そして、それは何がトリガとなって起こるのか、状態変化のタイミングでシステムはどのような状態になるのかを決める。ここで、トリガを起こす原因となるものが、「ものの中

表1 オリエンテーション決定のための行列

オブジェクト	半製品	機械バッファ	機械
アクティビティ			
バッファへの投入	1	1	
機械への投入	1		1
機械加工	1		1

心モデル」の「もの」に対応する。

このように、モデルには、世界観の決定やオリエンテーションの決定等の一定の標準があり、その下で、ある程度環境に応じた分かりやすいものが求められる。

3. シミュレーションモデルの自動生成

オペレーショナルモデルを作ることは、シミュレーション分析の専門家にとっては、一定の規則下であっても自分の好みで作り方を考え、システムを掘り下げていくことであり、興味深く、楽しいプロセスである。しかし現場で対象システムを設計している人にとって、シミュレーション分析をしようとした時にその前に立ち塞がるモデル化のプロセスは大きなハードルであるに違いない。設計者は、早く、自分のプランが効率的か、どこかに問題を抱えていないか、を具体的に見たいのである。それなのに、離散シミュレーションの枠組みを知ってからでないとシミュレーション分析ができない、どんな目的でシミュレーションをしたいのか、どの程度の詳細度で解を得たいのか、システム内のロジックがどの様になっているのか、が分かっているならば、概念モデルは作ることができる。しかし、離散シミュレーションの枠組みを知らないとオペレーショナルモデルはできないので、シミュレーションはできない、対象システムの分野に籍を置く人にとって、通常は離散シミュレーションの枠組みは未知のものである。習得するか、コンサルタントに依頼するかしかなかく、時間や費用がかかる。

そこで、筆者等はコンピュータネットワークシステムといった特定の分野の評価に使うツールとして、モデルの自動生成機能を持ったシミュレータを開発した。これは、コンピュータネットワークシステムを設計販売するインテグレータのシステムエンジニアや、ネットワーク運用監視をおこなうカスタマーエンジニアが、システム設計情報や実際のLAN/WANで流れているパケットのログデータをシミュレータの入力として与えるだけで、離散シミュレーションのモデルが定義でき、シミュレーション分析ができるツールである

[3].

モデルの自動生成機能とは、あらかじめオペレーションモデルを提供しておき、それになぞられてプログラミングモデルを作る機能である。したがって限られた概念モデルに対してのみ適用可能なため、通常的分析目的を明確にして概念モデルを作り、オペレーションモデルを作るといったプロセスとは逆行したプロセスをたどることになる。つまり、オペレーションモデルとプログラミングモデルが先にあり、適用できる概念モデルと現実システムが、そのモデルによって規定されることになる。

しかし、LAN/WANシステムの世界では、各要素には標準規格があり、さらにその要素を繋げたネットワークの中でも標準プロトコルがあるので、システムの動きのロジックの定義は既知である。また、コンピュータネットワーク性能評価業務の中で評価すべく目的は一般的には限られている。その目的とは、

- ・ボトルネックの追求とその解決策の発見
- ・期待するレスポンスを保証するための構成の検討
- ・システム性能限界の予測

というものである。したがって、例えば研究者が目的とするような、新しく設計したプロトコルの性能検証等といったことはできない。

シミュレーションのモデル化こそがORそのものであるのに、それを自動化するとはORではないと思われる方もいるだろう。実際に、OR学会の研究発表会でこのモデル自動生成について発表した時には、会場からその様に指摘された。確かに、前にも述べたが、モデルを構築すること自体がシステムの構造を理解することであるが、この自動化プロセスでは、その工程はツールを使う人の知識の前提であるとしている。そして、シミュレーションによる実験分析そのものを、シミュレーションになじみのない人に手軽に利用してもらうことができるツールをと考えたのである。1分野のごく限られたモデルであるが、自動生成されるモデルとは、標準化の最たるものであり、誰が使ってもこのシミュレータでは標準化されたモデルを使うことになる。

4. エージェントベースモデルとシステムシミュレーションモデル

最後に、モデルの話題として、最近特に研究者を中心に、盛んに使われているエージェントベースモデルをシステムシミュレーションモデルと対比してみよう

と思う。

最近、同一システムでも、エージェントベースモデルが使われたり、システムシミュレーションモデルが使われたりしている。

例えば、避難シミュレーションや、従来、待ち行列系のシステムとして離散システムシミュレーションで行っていたような、窓口業務に人が並ぶ様なシステムのシミュレーションなどである。

元来、複雑系システムの挙動を理解するためのシミュレーションとしてエージェントベースシミュレーションは生まれた。複雑系システムとは、「システムを要素に分解し、要素の持つ特性を解明していれば、もとのシステムの挙動が理解される」という要素還元主義が通用しないものと言う[4]。物理・化学系よりもむしろ生命系・社会科学系に適用されている。システム内の主体が持つ適応動作がシステムの複雑性を生んでいて、その特性を解明するためにシミュレーションを行う。

システムの目的に応じた目標があり、その目標を達成するための個々の要素の挙動をどの様に規定したらよいかをシミュレーションにより求めようとするシステムシミュレーションモデルに対し、システム内に内在される要素の特性が規定されていて、その要素どうしの相互作用により、どの様な挙動を示しシステムとしてどのような結果をもたらすかをシミュレーションにより理解しようとするエージェントベースモデルは、互いに全く逆のアプローチを前提としている。

例えば工場のラインで生産する生産量が月産所要数になるように、個々の機械の性能をどのくらいにすべきかとか、機械への投入ルールをどの様にしたら良いかとかをシミュレーションで見出そうとしているのがシステムシミュレーションモデルである。しかし、最近では自律分散制御されている機械もあるので、その様な機械を配置する工場では、その立ち上がり期には、どの様な過程で定常状態を作り出していくのかをエージェントベースモデル的アプローチで観測し、その過渡期の生産量はどのくらいになるかを知ることも必要であろう。

この様に、エージェントベースシミュレーションとシステムシミュレーションでは分析目的がちがうので、概念モデル自体が違う。概念モデルを作成する段階で、この二つのシミュレーションモデルのどちらが適当であるかは決まっているとあって良いであろう。

同じ避難シミュレーションでも、建築計画の際、許

認可のために所要の時間で建物の中からの避難が完了するかをシミュレーションし、所要時間内に避難可能な様に、通路の幅や扉の位置を適正に設定するためのシミュレーションは、システムシミュレーションモデルを使うべきである。しかし、災害時の避難の際には、人間の心理状態により予期せぬ事態が起こりえる。他人の影響や自己の判断により挙動が決まって行くような過程を理解して、避難の際の体制を考えていこうとするシミュレーションであれば、エージェントベースモデルで考えることになる。窓口業務に人が並ぶ時の窓口数を決める問題でも、システム設計の際にはシステムシミュレーションで考え、人の到着を確率として捉えることになるが、社会事象として、公共の場への人の到着過程が自己の特性や他人の影響によって左右されるのであれば、その過程を理解するためのエージェントベースモデルが有効となる。

エージェントベースモデルの構築に関しては、オペレーションズ・リサーチ 2004 年 3 号の特集で取り上げられている。社会科学の立場でのシミュレーションの利用に関する話題は、システムシミュレーションを主に使用している者にとってモデリングに関わらず、その分析のアプローチ方法においてとても興味深い。エージェントベースモデルは未知の結果をもたらすためのモデルを構築するという意味では、アートの、想像力を必要とし、その標準化もまたむずかしいと思われる。しかし、個々のモデルは、社会科学で実証されてきた現象と検証され、現実的な解が得られている。

最近、世界中でいろいろな災害が発生し深刻な被害が起こっているが、日本の社会学者は、津波に対する避難のシミュレーションの研究等を、もう何年も前から、エージェントベースモデルを使い行っている[5]。今後は、こういった研究に基づいた災害対処計画が策定されていくことになり、エージェントベースモデルに対する期待もさらに大きくなるだろう。

5. おわりに

本稿ではシミュレーションのモデル化の、標準化できる側面とできない側面について考えてきた。

概念モデルが同一でも、複数の代替モデルが存在し、そのモデルの選び方は、システムの特性等で決まってくる。しかし、それが絶対ではなく、時として分析者が属する環境や、分かりやすさ、好みに依存して決めることも可能であるとの見解を述べた。また、ある分野を特定し、取り得る概念モデルを規定したならば、

そこでは標準モデルを作ることができ、モデルの自動生成も可能であるという例を示した。

最後に、同一システムを扱うシミュレーションモデルでも、目的や概念モデルによっては、システムシミュレーションモデルではなく、他のモデルで考えるべきであり、社会科学の分野ではエージェントベースモデルが活用されていることを説明した。

以上から、モデル化プロセスの中に、モデルを決定する岐路がいくつかあり、それをある標準で選び、モデルを作り上げることによって、適切で実行可能なモデリングが可能であること、また分かりやすさや好みも一つの標準であることをまとめたい。

参考文献

[1] 森戸晋：シミュレーション・モデル化の「理論」を指して—代替モデル/プログラムの認識, オペレーション

ズ・リサーチ, 1990年2月.

[2] 森戸晋, 相澤りえ子, 貝原俊也: VisualSLAM によるシステムシミュレーション〈改訂版〉, 共立出版, 2001年.

[3] 相澤りえ子: ネットワーク性能評価専用シミュレータのモデル化技法, 1998年度日本オペレーションズ・リサーチ学会 秋季研究発表会アブストラクト集, pp.156-157, 1998年.

[4] 貝原俊也: マルチエージェントシミュレーションと分散シミュレーション, 平成13年度第2回ORセミナー 分散シミュレーション—その最前線と事例—, pp.23-38, 2001年.

[5] Masaki Fujioka, Kenichi Ishibashi, Hideki Kaji, Isao Tsukagoshi: A Multi-agent based simulation Model for evacuees escaping from Tsunami disaster, SeoulSim (The Seoul International Simulation Conference) 2001, pp.306-312, 2001.