シミュレーション・モデル化の「理論」を目指して

——代替モデル／プログラムの認識——

森戸　晋

1. はじめに

離散事象型（以下，離散型）シミュレーションの研究の圧倒的多くは，出力結果の解析や分散減少法，感度分析等，統計的側面に関するものであった．この状況は今後も大きく変化することはないであろう．しかしながら，筆者の生産システムを中心とする限られた経験と情報では，実験の応用の場においては，モデル化や確率的要素を伴わない確定的分析，システムがなんらかの意味でトラブったときの打開策の検討等が，統計的側面以上に問題となっているように見受けられる．これらの問題の多くは，一言で言うならば，「モデル化」に関する問題といって差し支えないと考えられる．このような問題の検討は，始まったばかりである．この分野で，実りある研究成果が出せるかどうかは定かではないが，「指針」のようなものでもよいからほしい，というのが応用の場に携わる人々の願いではないだろうか．

ここでは，ボトムアップ・アプローチで，シミュレーション・モデル化について考えてみたい．ボトムアップとは，具体的問題に即して，実際にプログラムを作成する過程におけるモデル化／プログラム化の問題を考えるという意味である．なお，本稿では議論しないが，トップ・ダウン・アプローチでモデル化についての議論を進めているZiegler, Oren等の研究[6]もあることにふれておきたい．ただし以下に述べるような議論と，Ziegler等の抽象的議論とがどのような形で合流するのか，あるいは，合流できるのかは，今のところ明らかでない．

最近では，GPSS, SLAM, SIMAN等のシミュレーション言語の普及・性能向上によって，シミュレーション言語を活用する割合が非常に高くなっている．本稿では，シミュレーション言語の利用を想定し，モデル化やシミュレーション言語を用いたプログラミングが経るステップを考えることによって，各ステップの役割を明確にし，各ステップにおいて代替性を認識することの重要性を明らかにする．なお，最後に，モデルという枠の中に入ると考えられる他の研究のキーワードをいくつか紹介する．

もりと　すすむ　早稲田大学　理工学部　工業経営学科
〒169 新宿区大久保3—4—1

2. 3段階のモデル化，「オペレーショナル・モデリング」の位置づけ，代替性

筆者は，モデル化のプロセスを3段階に分けて考えることにしている，3段階とは，1) 概念モデル化(conceptual modeling). 2)オペレーショナル・モデリング(operational modeling), 3) プログラミング(programming) である．

概念モデル化は，概念モデルを定めるプロセスで，その最終産物は，1)モデルの境界 (boundary), 2)どの程度，細部にわたってシステムを見るかという詳細度(levels of detail), 3)システムの動きに関する基本ルール (system logic), 等の明確な定義であり，モデルの仕様，前提条件，仮定等と同義語と考えてよい．最終的な概念モデルは，数式を使わなくとも，人によって解釈が異なることがないよう，正確に定義されたものでなくては困る．

プログラム化は，基本的には，概念モデルを離散型シミュレーション言語で実現するプログラムを作成するプロセスである．なお，離散型シミュレーションの世界では，このプロセスをモデル化と呼ぶ人も少なくない．

これらの間に位置するオペレーショナル・モデリングは，概念モデルを等価変換して，離散型シミュレーションの枠組みの中で，ダイナミックなシミュレーションが実行できるようにするプロセスをさす．特に，システムが複雑な場合，概念モデルは，それが正確であればあるほど無味乾燥であることが多い．つまり，言葉で書かれたモデルの仕様，すなわち，概念モデルは，それだけを

© 日本オペレーションズ・リサーチ学会．無断複写・複製・転載を禁ず．

見ても，いかなる形で離散型シミュレーションの枠組みに落とせるかが明らかでないことが少なくない．シミュレーションを行なうためには，ダイナミックにシステムを動かす必要があるが，これが，仕様の羅列から容易に想像できないのである．そのような場合に，概念モデルとプログラミングとの間の橋渡しをするものを筆者は「オペレーショナル・モデル」と呼んでいる．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名づけたのである．

後に示す例からもわかるように，簡単なシステムでも，中間段階のモデル化が役立つ場合もあるので，与えられた概念モデルを等価変換して，プログラムに落とすまでの中間的プロセスをオペレーショナル・モデリングと考えておく．

本稿の中心的テーマは，「代替性」であるが，以上のように，モデル化を3つの段階にわけると，概念モデル化における代替性とは，現実のシステムをどのようにとらえ，どのような技法で分析するかに焦点が当てられることになり，多くの場合，完全に等価というわけではない複数の候補の中からどれを選ぶかが問題となる．多くの人は，これこそがモデル化であると主張するであろう．これに対して，概念モデルの定義が確定したあとで，オペレーショナル・モデルや，プログラムの代替性も考えられる．こちらの代替性は，等価性と言い替えてもよい．本稿で取り扱うモデル化は，とりあえず，後者の意味での代替性＝等価性である．

概念モデルで定められた仕様からは，最終的なプログラムが直ちに浮かび上がらず，中間段階のモデル化を必要とする例としては，中野他[1]を参照されたい．この例では，2つの一見全く性格の異なるオペレーショナル・モデルが，所与の概念モデルを実現する例が紹介されている．オペレーショナル・モデルは，プログラム側から見れば，プログラムの基本設計とも考えられるので，オペレーショナル・モデルの考え方が変わることによって，作成されるプログラムとその性能が大きく変わりうることは，容易に想像できよう．

オペレーショナル・モデリングの醍醐味は，複雑なシステムでこそ味わえるもののように思われるが，ここでは，その「雰囲気」を味わっていただくために，簡単なバッファを有する多段直列待ち行列システムを考える．

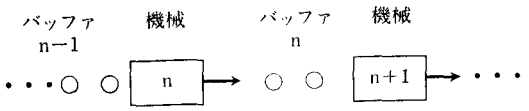

図 1 有限バッファを有する直列待ち行列システム
(バッファ・サイズ＝2を想定)

3. 有限バッファを持つ直列待ち行列システム

考えるシステム（図1参照）は，基本的には通常の待ち行列と同じであるが，バッファが有限なので，ブロッキング現象が起こる．ブロッキングには，以下の2種類のタイプがある：

生産ブロッキング(Production blocking)：機械nでの処理（加工）が終わった段階で機械nの直後のバッファ（これをバッファnと呼ぶ）が満杯のときには，処理が終わった部品を機械nから取り出すことができずに，部品が機械nをブロックし，バッファ$n-1$に部品があっても，機械nの処理を開始できない．いずれ，機械$n+1$の処理が終了し，バッファnに空きができると機械nのブロッキングが解消される．

通信ブロッキング（Communication blocking)：機械nでの処理は，直後のバッファnに空きがある場合に限って許される．したがって，機械nでの処理が終わった時点で，処理を終えた部品がゆくべきバッファの存在が保証されている．

通信ブロッキングには，

1) 機械を部品の待ち場所として使えない場合

2) 機械を部品の待ち場所として使える場合

という2つのサブ・クラスがある．ここでは，C1タイプの通信ブロッキングを考えることにしよう．つまり，これから考える概念モデルは，C1タイプの通信ブロッキングを有する直列待ち行列システムである．部品の到着やサービス時間については，適当な前提がおかれているものとするが，プログラムの基本構造に影響を与えない(しかし，実験の遂行，結果の統計的分析においては，これらの選択が重要な役割を果たすことは言うまでもない）ので，ここでの議論からはずすことにする．

シミュレーション言語は，大なり小なり，分析対象となるシステムの一般形態を想定している．分析すべきシステムがこれにピッタリ合致する場合は，簡単にモデル化ができる．ちなみに，SLAM に備えられたブロッキ

© 日本オペレーションズ・リサーチ学会．無断複写・複製・転載を禁ず．

```
 6          RESOURCE,BUF2(2),3;
 7          RESOURCE,BUF3(2),5;
    ...
10          RESOURCE,M2(1),4;
11          RESOURCE,M3(1),6;
    ...
22  AWB2    AWAIT(3),BUF2;
23  AWM2    AWAIT(4),M2;
24  FB1     FREE,BUF1;
25              ACT/2,EXP(1.,2);
26  FM2     FREE,M2;
27  ;------
28  AWB3    AWAIT(5),BUF3;
29  AWM3    AWAIT(6),M3;
30  FB2     FREE,BUF2;
31              ACT/3,EXP(1.,3);
32  FM3     FREE,M3;
```

図 2 プログラムA

```
 5          RESOURCE,M2(1),6;
    ...
22  QP2     QUEUE(4),,,,SEL2;
23  QB2     QUEUE(5),2,,,SEL2;
24  SEL2    SELECT,ASM,,,QP2,QB2;
25              ACT;
26  GON2    GOON;
27  AW2     AWAIT(6),M2;
28              ACT,,,QB1;
29              ACT/2,EXPON(1.,2);
30  FR2     FREE,M2;
```

図 3 プログラムB

```
 9  MC2      QUEUE(2),,1,BLOCK;
10               ACT/2,EXP(1.,2);
```

図 4 プログラムC

ングは，生産ブロッキングである．言語にとって都合のよいシステムが分析対象となることが多いのは当然であるが，現実が言語の枠組みに合うかというと必ずしもそうではない．言語があるあらゆる状況に簡単に対応できるようにするには，無理があり複雑なシステムに対しては，言語の提供する枠組みで簡単な表現ができずに，工夫を強いられることも少なくない．

さて，直列待ち行列システムのモデル化の結論を急ぐことにしよう．**図2**一**図4**は，部品や機械の動きを完全に同一にできる「等価な」SLAM プログラムである．もちろん，等価な動きを実現するためには，乱数系列を一致させる等の配慮が必要である．紙面の節約のため，図2以外は1ステージ（第2ステージ）分だけ示しているが，他のステージも全く同一の構造を考えればよいことは理解していただけよう．

プログラム（図2），プログラムB（図3）は，通信ブロッキングの定義を忠実に実現した2つの等価なSLAM プログラムである．プログラムAは，部品をプログラム中の要素(entity)とし，機械とバッファをリソースとしたプログラムである．各ステージともバッファ・サイズ＝2を仮定しているので，BUF 2，BUF 3のリソースを各2個宣言している．一方，プログラムBは部品とバッファを要素とし，機械をリソースとしたプログラムである．ここでは，SELECT ノードのASM（アセンブリー）オプションを用いて，部品が次に入るべきバッファを確保してから機械処理を行なうことを表現している．行23の QUEUE ノードの2番目のパラメータで指定される初期要素数2がバッファ・サイズに対応する．

これらのプログラムはいずれも短いプログラムであるが，単純というわけでもない．これは，いずれのプログラムも，1ステージで「閉じていない」ことによる．プログラムBでは，行28からQB 1へのジャンプが第1，第2ステージをつなげる働きをしており，逆に，第3ステージからはQB 2（行23）へのジャンプが入り込む．これは，機械nの開始がバッファ$n-1$の空きを生み，機械$n-1$のブロッキングを解除する，…というように，隣接するステージが独立でないことによる．

プログラムAにも，同様なステージ間の連結構造が存在するが，このプログラムにおけるリソースの「AWAIT（占有）-FREE（解放）」のループは複雑に絡みあっており，作成に苦しむばかりでなく，理解するのも非常に難しい．このシステムにおいては，同一シミュレーション時刻に発生する事象，すなわち，同時発生事象（simultaneous events）が本質的であり，同時発生事象を正しく処理するために，プログラムAでは複雑なリソースのループを設けている．

なお，同時発生事象によるモデルの複雑化は，上の例に特殊であるわけではなく，FMSやかんばん方式のように何らかの離散的要素がシステム中を動き回る生産システムではよく見受けられる現象である．ちなみに，図3のプログラム構造が，Pritsker［3］(p. 353) 他に示されているかんばんモデルのプログラグ構造と全く同じであることに気づくとおもしろい．すなわち，図3におけるバッファをかんばんとみなすことによって，後者のプログラムの基本構造ができ上がっている．こうしてみると，ブロッキングを伴う生産システムとかんばんシステムとは，後工程引き取り(pull)型か押し出し(push)型かの違いは別として，同一構造を有している部分があることが理解できる．

プログラムA，Bは，プログラム・レベルにおける等価な代替案ということになろう．それでは，オペレーシ

© 日本オペレーションズ・リサーチ学会．無断複写・複製・転載を禁ず．

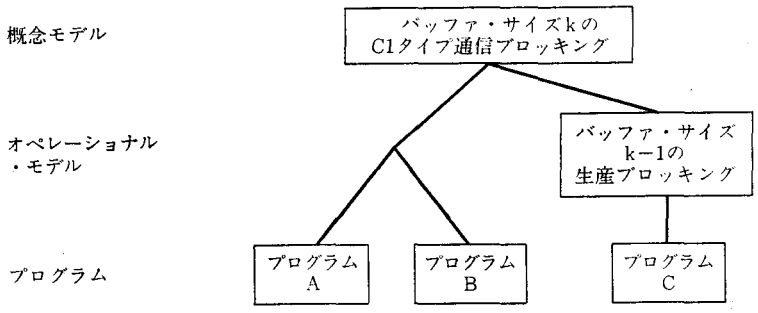

図 5 ブロッキングを伴う直列待ち行列システムにおける3段階のモデル化

ョナル・モデリングはどこに登場するのだろうか．答は，図4のプログラムCである．SLAM には，生産ブロッキングのモデル化機能が QUEUE ノードの block オプション（図4の行9参照）として与えられている．ブロッキング指定の直前のパラメータ1はバッファ・サイズの指定である．したがって，プログラムCはバッファ・サイズ1の生産ブロッキングのプログラムということになる．プログラムCがプログラムAやBと等価ということは，以下の性質にもとづく：

命題 図1のような有限バッファを持つ直列待ち行列システムにおいて，バッファ・サイズ k の生産ブロッキングは，バッファ・サイズ $k-1$ のC1タイプの通信ブロッキングと等価である．[2]

したがって，もし，この等価性に気づけば，プログラムとしては，バッファ・サイズを1つ減らした生産ブロッキングのモデルをプログラム化すれば，C1タイプの通信ブロッキングのシステムの動きと全く同じ動きを作り出すことができることになる．なお，プログラムCは，あくまで，生産ブロッキングを想定しているので，結果の分析に当っては状態の「読み換え」等，若干の作業が必要になる場合もある．

以上をまとめると，プログラムCに至るモデル化のプロセスでは，与えられた概念モデルをそのままの形でプログラム化せずに，一度，別の等価なモデルに変換し，これをプログラムに実現していることになり，図5に示すように，ここでは，「バッファ・サイズが1少ない生産ブロッキングを有する直列待ち行列システム」がオペレーショナル・モデルということになる．プログラムAやBに対するオペレーショナル・モデルは，この場合，元の概念モデルと同一と考えるのが自然のようであるが，概念モデルが複雑になるにしたがって，それを特定言語の枠組みに落とす考え方を複数思いつく可能性が高くなる．

なお，生産ブロッキングのシステムを言語に備えられたブロッキング機能を使わずに表現しようとすると，プログラムAやBの形のものを考えざるをえなくなることに注意したい．したがって，オペレーショナル・モデリングでは，概念モデルを実現する複数の代替的・等価的な考え方を模索あるいは列挙し，それらと言語の提供する機能とのマッチングを考えながら，プログラムの基本枠組み，すなわち，オペレーショナル・モデルを決定することになる．

4. プログラム・レベルの代替性

プログラムA，B，Cを比べてみると，ステージ当たりのコマンド数は，それぞれ，5，9，2である．また，これらのプログラムを，原材料の無限供給を想定した4ステージ・モデルとして実行したときの実行時間比は，概ね，A，B，C＝2.0：2.3：1であった．プログラムCが他の2つのプログラムより2倍以上早いことがわかる．

前節では1つの例を示したにすぎず，一般論としては代替性を意識しながら，適当なオペレーショナル・モデルを選択し，それにもとづき注意深くプログラミングを行なうことによって，プログラムの，A）作りやすさ，B）わかりやすさ，C）実行速度，D）記憶容量，等の側面で望ましい性質を持つプログラム開発への道が開けると考えられる．一般には，前節の例のように，C1型の通信ブロッキングと生産ブロッキングの等価性に気づきさえすれば，各尺度から見て望ましいプログラムができるとは限らず，評価尺度間のトレードオフを考慮しなければいけない場合が多いと考えられる．

プログラム・レベルにおける代替案を考えるに当っての鍵は，

「何を，プログラム中に動き回る『要素』(GPSSTransaction，SLAM の entity 等) ととらえるか」の決定にあり，これを「モデルのオリエンテーション(orientation) の決定」と呼ぶ．

システムの中に，数多くの「役者」が登場し，それらが複雑に絡み合ってサービスを進めていくようなシステムこそ，モデルによる分析が威力を発揮する舞台となるが，このような場合，どの役者をプログラムの「要素」として表現し，どの役者をリソースやその他の機能で表

© 日本オペレーションズ・リサーチ学会．無断複写・複製・転載を禁ず．

表 1　システムのオブジェクトとサービスとの関係

サービス＼オブジェクト	パーツ	パレット	機械	機械バッファ	工具	スタッカークレーン	ストッカー	ロードステーション
ローディング	1	1				1		1
機械バッファへの移動	1	1		1		1		
ストッカーへの移動	1	1				1	1	
機械加工	1	1	1		1			
パレット交換	1	1				1		1

現するかによって，プログラムの様相や性質が相当変わってくる．

現段階では，適正オリエンテーション決定のための標準的手順というようなものは見あたらないが，複雑なシステムのモデル化に対しては，少なくとも，表1のような行列を作り，これを参考にしながらオリエンテーションを定めるのよい．表1はFMSシステムの例で，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1（あるいは，実行に必要なオブジェクトの数），さもなくば，0とする．一般に，

1）シミュレーションの実行に当って，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1つがプログラム中の要素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）あるアクティティに対して，2つ以上のオブジェクトを要素とすると，同時発生事象のリスクやプログラム長大化の可能性が高まる

等の性質がある．こう考えると，オリエンテーション決定問題は，数理計画の集合被覆問題(set covering problem）に類似する問題であることがわかる．

5. 代替モデル化について…エピローグ

本稿を読んで下さった読者は，「なんだ，SLAMのプログラミングの話ではないか」と思われるかもしれないが，著者はそう思わない．第1に，ネットワーク・モデルやブロック・ダイアグラムというようなプロセス中心のモデル化機能を持つシミュレーション言語の使用は大前提としているものの，他の言語でも同様の議論ができるはずである．

第2に，プロセス中心のモデル化では，限られた「持ち駒」＝コマンドでシステムを表現することを強いられるから，でき上がったプログラムには，自ずと，システムの構造が浮かび上がってくる．しかも，オリエンテーションを変えるということは，同一のシステムをいろいろの視点，角度から見ることにほかならない．したがって，本稿で論じてきたことは，単に，より早く，よりわかりやすいプログラムを開発するという「シミュレーション言語のソフトウェア・エンジニアリング」的側面だけに留まらずに，システムの構造を明らかにするという側面を併せ持つと考えられる．

どのレベルにおいても，代替性を意識することは，モデル化，さらには，オペレーションズ・リサーチの本質ではないだろうか．

6. モデル化に関連するその他の研究

離散型シミュレーションにする年一度（12月）の集会であるWinter Simulation ConferenceのProceedingsが理論・応用・ソフトウェアに関する最新の情報源である（問い合わせ先：The Society for Computer Simulation, P.O. Box 17900, San Diego, California 92117）ので，これを見ていただくとして，以下にいくつかのモデル化に関する研究のキーワードをいくつか示しておく：

1）確定シミュレーション（[4]等）：Pritskerによれば，実際の離散型シミュレーションの25～30%は，確率要素を含まないシミュレーションであるという．また実際は，確率的変動があっても，あえてそれを除いてシミュレーションを行なうことによって，よりシステムの本質が明確になることもある．通常の待ち行列のように，確率要素を除去することによって本質が消えてなくなる場合もあるので注意を要するが，離散型シミュレーションにおいて，確定的分析の役割と進め方を明らかにすることは，実務的に重要な意味を持つと思われる．

2）他のORの技法との関連（[4]等）：シミュレーションから得られる情報を基礎に，他のOR技法（たとえば，待ち行列モデル）でのモデル化を考える．これは概念モデルでの代替性，つまり，完全には等価ではないモデル間の関係を探ることにつながる．

© 日本オペレーションズ・リサーチ学会．無断複写・複製・転載を禁ず．

新時代のコンピュータ総合誌
ComputerToday
90年1月号／発売中／定価930円
ファジィ・ニューロコンピュータ
——コンピュータがひらく新しい世界4——
ニューロとファジィの融合　合原一幸
柔らかな知的処理を目指した
ファジィとニューラルネットの融合　高木英行
ニューロコンピュータをどう思うか　向殿政男
ファジィコンピュータをどう思うか　甘利俊一
ファジィチップ　廣田　薫
ニューロチップ　堀尾喜彦・合原一幸
ファジィ制御　高木友博・深海　悟
ニューロ制御　堤　一義
μBRAIN：ニューラルネットワークを用いたAIシステム　古谷立美
ファジィニューラルネットに向けて　今崎直樹・山口　享
ニューラルネットワークによるファジィコントローラの構成　中西祥八郎・高木敏幸
音声認識　市川　熹・天野明雄
月刊誌
数理科学
90年2月号／発売中／定価960円
電子・陽電子
極微の世界のスーパースター
はじめに電子の諸相　鈴木　皇
"電子"の由来　西尾成子
自由電子磁気モーメントを中心にして　加藤正昭
いうことを聞かない電子"驢馬電子"　伏見康治
動きまわる電子"金属電子"　長岡洋介
集団で動く電子"プラズモン"　松平　升
手をつないだ電子"クーパー対"　大塚泰一郎
不思議な低次元電子　鹿児島誠一
電子の生成と消滅　兵頭俊夫
電子線ホログラフィーで見る世界　外村　彰
低速陽電子ファクトリー　谷川庄一郎
陽電子顕微鏡　兵頭俊夫
■最新刊　好評発売中
REDUCE入門
パソコンによる数式処理活用法
広田良吾・伊藤雅明共著　A5・定価2300円
►価格表示は，税込み価格となっています．
サイエンス社
東京都千代田区神田須田町2—4　安部徳ビル
☎03(256)1091　振替 東京7—2387

3) 事象グラフによる事象構造の分析（[5]等）：一般のシミュレーション言語より，さらに単純な構造を持つSchrubenの事象グラフを用いて，事象の構造やモデルの等価性を解明する．

[謝辞] 本稿執筆の場と豊かな研究環境を提供していただいたPurdue UniversityのBruce Schmelser, Jim Wilson両教授とPritsker Corporation会長のAlan Pritsker博士に辞意を表します．

参考文献

[1] 中野一夫，相沢りえ子他，「ダンプトラック運行シミュレーション・モデルの開発と走路区間のモデル化」，オペレーションズ・リサーチ，Vol. 32, No. 5, pp. 259-268, May 1987,

[2] R. O. Onvural and H. G. Perros, "On Equivalences of Blocking Mechanisms in Queueing Networks with Blocking", Operations Research Letters, Vol. 5, No. 6, pp. 293-297, December 1986.

[3] A. A. B. Pritsker, C. E. Sigal, and R. D. J. Hammesfahr, SLAM II Network Models for Decision Support, Prentice-Hall, 1989.

[4] A. A. B. Pritsker, Developing Analytic Models Based on Simulation Results, SMS E9-25, Purdue University, August 1989 (to appear in Proceedings of 1989 Winter Simulation Conference).

[5] L. Schruben, "Simulation Modeling with Event Graphs", Communications of the ACM, Vol. 26, No. 11, pp. 957-963, November, 1983.

[6] B. P. Ziegler, Multifacetted Modelling and Discrete Event simulation, Academic Press, 1984.

© 日本オペレーションズ・リサーチ学会．無断複写・複製・転載を禁ず．