

離散系シミュレーション：モデリングとソフトウェアを中心として

01603200 早稲田大学 森戸 晋 MORITO Susumu

1 はじめに

離散系シミュレーションのここ10年余の普及はめざましいものがある。技術そのものは1960年代から存在していたものの、1980年以前はソフトウェアとして事実上GPSのしか存在しなかったものが、現在では国内だけでも10を越えるソフトウェアが市場に出回り、しのぎを削っている。管理技術としての離散系シミュレーションには現在、普及の一層の拡大という明るい将来展望がある反面、気がかりな兆しも見受けられる。

1.1 普及の拡大

明るい面としては、

- ソフトウェアの普及・機能向上・低価格化
- 適用領域の拡大
- 一部分野における適用の常識化や義務づけ

等が上げられる。

適用領域では、製造ラインや工場内物流、自動倉庫、物流配送システム等、生産システムを対象とするものが圧倒的に多く、これに計算機や通信システムを対象とするものが続くという傾向はこれまで通り(梅田, 森戸 [3])であり、これらの領域ではシミュレーションの利用が定着したといえる。

一方、利用の目的という観点から見ると、従来はシステムの事前評価が圧倒的に多く、次いで改善計画のためが続く、日常業務の制御のためのシミュレーションはごく限られていた。ところが、スケジューリングが注目されるようになり、制御や運用のためのシミュレーションが急速に普及しつつある。

また、最近のスケジューリングや一部のビジネスプロセスリエンジニアリング(BPR)のソフトウェアに見られるようにシミュレーション専用ではないソフトウェアにシミュレーション機能が含まれるケースが増加している。

さらに、従来、適用対象としては上がらなかった分野、たとえば、レジャー産業(たとえば、テーマパークの設計)、イベント(博覧会の企画)、公共システムの計画、あるいは、ファーストフードチェーン等に代表される工場外のサービス産業における物流といった分野に、シミュレーションが急速に浸透しはじめていることに注目したい。

1.2 意思決定者の期待

シミュレーションの将来はバラ色かという必ずしも手放しに喜んでいられる状況ではなく、心配な面もある。膨大なコスト、リスク、時間をかけずに、「大ざっぱな見当」をつけたいというのがシミュレーションに対する意思決定者の期待である。すなわち、大ざっぱな見当(英語では、しばしば"ball-park figures"という表現が用いられる)がつけられ、さらに適当な感度分析結果が提示出来れば、意思決定者は代替案の評価・選択が可能となり、必要とあれば選択された代替案をさらに詳しく調べる事が可能となる。ところが、シミュレーションは、ほどほどの労力で大ざっぱな見当をつけるという意思決定者の期待に

応えていないという声が、普及拡大の陰で聞こえてきていることに注意する必要がある。

本発表では、離散系シミュレーションの現状と将来展望を、上述の不安をふまえながら、モデリングならびにそれを実現するソフトウェアという観点からサーベイする。

2 離散系シミュレーションで何を見ようとするのか

タイプ1: 確定的な単純大規模システムの動的挙動

人は一般に、システムの動的挙動を把握するのは得意とは言えない。不確実性が存在しないと分かっているにもかかわらず、たとえば多品種からなる多数の要素が系内を流れていたり、流れる要素によって通過する条件が異なる多くの段階を経なければいけないといった「大規模」システムの動的挙動を把握することは、人間に向けた仕事ではない。このような分析は、表計算ソフトでも可能であるが、当然、離散系シミュレーションのソフトウェアで処理可能である。現行のソフトウェアの機能(表示機能等)を有効に活かせば、表計算ソフトでは容易に出来ないことが簡単に達成可能である。

タイプ2: 不確実性を伴うシステムの挙動

例えば、M/M/1のような単純なシステムでも、不確実性が存在するとその挙動を評価することは決して容易でないし、予想に反する結果が生まれることも多々ある。ましてや、大規模なシステムが不確実性を伴うとなると大変である。適切な実験計画の下でシミュレーションを実施し、結果をしかるべき方法で分析することによって、不確実性を伴うシステムの挙動、とくに動的挙動を把握することが出来る。

タイプ3: 要素の相互作用によるシステムの動的挙動

シミュレーションは、要素間の相互作用、とりわけ、有限リソースの占有に関わる要素間の競合を扱う場合が多い。このようなシステムの動的挙動は、そうした相互作用を支配するルールに依存する。システムが確定的な場合でもこのタイプのシステムはシミュレーション分析の対象となるが、不確定要素が入る、すなわち、タイプ2とタイプ3の混合タイプは、その分析が一層難しくなる。実システムのルールは多岐多様にわたり、解析的な取扱いが困難な場合がほとんどであり、シミュレーションが唯一の分析方法であることが多い。

実際のシミュレーションは、これらすべての要因を包含した場合が多いが、離散系シミュレーションで見たいことを上の3タイプに分けることはモデリングの上からも、また結果分析という立場からも意義がある。

タイプ1とタイプ2のシミュレーション結果の解析方法は確立している。標準的な離散系シミュレーションのテキスト(たとえば、Law and Kelton [2])には、タイプ2の確率的シミュレーションを想定した実験の計画や結果の解析方法が解説されているし、ソフトウェアの中には(従来のソフトウェアより)適切な結果解析機能を組み込むものが現れたり、シミュレーション結果解析のための専用ソフトウェア(SIMSTAT)が登場している。ところが、タイプ3のシミュレーションに関しては、必ずしも結果の取り扱い方法が確立していないというのが実状である。

意思決定者が期待する大ざっぱな見当をつけるためには、より大局的なシステムの挙動や特性を把握する必要があるが、そのためには要素(とリソース)の相互作用に踏み込んだタイプ3のシミュレーションを行って詳細レベルに立ち入った分析を行わないとモデルの妥当性が評価できない、という問題がシミュレーションにつきまとう一つの頭痛の種になっている。

幸い、モデルを基礎とするORの分野の中で、シミュレーションほどその研究・開発においてモデリングの議論が活発な分野はない。具体的には、例年12月に米国で開催されるWinter Simulation ConferenceのProceedingsを参照されたい。そうしたモデリングに関する議論の中に、階層的モデリング、メタモデル、というような、上述の問題の解決に役立つような概念が生まれており、それらを実現するソフトウェアも生まれつつある。

3 階層的モデリング (Carrie)

Carrie [1] は、階層的モデリングに関わる基礎的かつ相互に関連する概念として以下の5つを上げている。

- (1) 階層
- (2) モジュール性
- (3) トップダウン/ボトムアップ
- (4) 近似
- (5) 段階的詳細化

いわゆる IDEF 型のアプローチは、生産システム工学やソフトウェア工学の分野において、システムを階層的に分析する方法としてよく用いられている方法である。IDEF 型のアプローチは、

- a. システムをいくつかの「部分」に分割して、ある部分を詳細にモデル化し、他は大ざっぱに捉える
- b. システムで遂行される一部のアクティビティに焦点を当て、これを詳しく捉え、他は大ざっぱに捉える

という型のいずれかで展開される。システムをその部分、すなわちサブシステムに分割していくという前者のアプローチはきわめて自然な発想であるが、この場合、各サブシステムの境界とサブシステム間の相互作用を明確にする必要がある。IDEF 型の表現では、自動的にアクティビティが強調されるようになるが、システムの階層的把握という意味では、後者のアクティビティ中心の捉え方がより扱いやすい。

SLAM や SIMAN 等の代表的シミュレーション言語では、システムを記述するために要素(entities)とリソース(resources)を区別している(両者を識別せず、両者を異なる種類の要素と考えるソフトウェアもある)。アクティビティの入力となり、アクティビティによって何等かの処理を受け、形をかえるのが

要素であり、そのアクティビティを遂行するために必要となるのがリソースである。

アクティビティをより詳細に見ようすると、要素とリソースの差は明白になる。たとえば、FMSにおいて部品を加工するというアクティビティは、

- ワークを載せるパレットをロード/アンロード・ステーション(LUL)に運ぶ
- LULで、ワークをパレットに固定する
- ワークを機械Aに搬送する
- 機械Aで加工を行う
- ワークを機械Bに搬送する
- 機械Bで加工を行う
- ワークをLULに搬送する
- LULにて、ワークをパレットからはずす

というような一連のサブアクティビティに分割される。システムを流れる要素、すなわち、ワークの「初期状態」と「最終状態」は、アクティビティのサブアクティビティへの分割には影響されない。

これに対して、アクティビティからサブアクティビティへの展開は、リソースに大きな影響を及ぼす。一連のサブアクティビティで使われるリソースとして、パレット、LUL、LULのオペレータ、機械A、機械B、搬送装置等が考えられ(他にも、治具、工具等が考慮の対象ともなりうる)が、必要となるリソースの組合せはサブアクティビティ毎に異なり、FMSにおいてワークを加工する、というアクティビティの遂行にあたり、これらのリソースが常時必要となる訳ではない。

そこで、FMSにおけるワークの加工というマクロなアクティビティの時間遅延を正確に評価するためには、要素とリソースの間の詳細な相互作用に着目した(タイプ3)シミュレーションが必要になる。

以上の議論は、階層的モデリングにおいて階層分割がシステムの「部分」に基づくのではなく、アクティビティに基づいて行われるのが望ましいこと、また、共通のリソースを競合するアクティビティ群はある階層の一塊にまとめられるべきであること、等を示唆している。これらを含め、Carrieは「階層的モデリングの10原則」を提案している。

4 ソフトウェア

わが国で流通している離散系シミュレーションの商用ソフトウェアは、筆者の知る限り以下の通りである(ABC音順): AutoMod II, EXTEND, PCModel, QUEST, SIMAN/Cinema/ARENA, SimKit, SIM-PL+++, SIMSCRIPT II.5, SLAMSYSTEM and FACTOR/AIM, WITNESS, WORKBENCH.

これらのソフトウェアの動向や、階層的モデリングへの対応等については発表で述べる。

参考文献

- [1] A. Carrie, Hierarchical modelling — Principles and Review, *Proceedings of WORKSIMS '94*, 24-33, 1994.
- [2] A.M.Law and W.D.Kelton, *Simulation Modeling and Analysis*, 2nd ed., McGraw-Hill, 1991.
- [3] 梅田茂樹, 森戸晋, 離散系シミュレーションを取り巻く実態と展望, *オペレーションズ・リサーチ*, 38(11), 561-565, 1993.