離散事象シミュレーションの技術動向

Trends in Discrete Event Simulation Technologies

米田　清　K. Yoneda
田村　信介　S. Tamura

システムは構築が済んでから問題が発生しても応急的な対策しか施せない。システム開発に伴うリスクを最小化するには上流工程でのシステム評価が有効である。しかし，そのシステムはこれから実現するのでまだ存在しない。そのため，シミュレーションによるシステム評価が重要になる。
ここではシステム構築の上流工程ツールとして定着しつつある離散事象シミュレーションの技術動向を概観する。

Problems uncovered only after a system has been built allow no more than cursory repairs. Evaluating a system at an early phase of its development is therefore useful because it minimizes the risks involved in actually building the system. Simulations are important in this context, since the system being planned or designed does not yet exist.
This paper outlines trends in discrete event simulation, which is now a standard tool in upstream systems construction methodology.

1 まえがき

計算機シミュレーションは，プログラムによって実物の動作や性質を模擬・再現する方法であり，模擬訓練やアーケードゲーム，仮想現実感，気象予測，半導体の動作予測など，多岐にわたる。ここでは対象の動的な変化を模擬する技術，つまり初期状態と因果関係とを与えてシステムの時間的な挙動を予測する技術としてとらえる。数学的には，広い意味での微分方程式を解くことに当たり，関数や確率過程を時間に関して数値的に積分する方法である。ここでいう関数は必ずしも数式ではなく，因果関係を形式的に記述できればよい。

数値的な積分法には時間駆動と事象駆動の二つがある。微小時間ずつ，時刻が進むごとに新たな状態を計算するのが時間駆動であり，運動方程式に従って連続的に状態が変化する現象の記述に適している。しかし，「10時に部品の搬送が始まって搬送中となり，30分間で搬送が終了するとその加工が始まる」というように，ある事象が起こってから次の事象が起こるまでは同じ状態を持続するような場合には効率が悪い。状態変化がないにもかかわらず，状態の再計算を繰り返すことになるからである。事象駆動はこのような場合に適しており，なんらかの事象が生起するときにだけ状態を更新する。

この方式は離散事象シミュレーションとよばれている。図1に示すように状態変化が階段関数で表される場合に対応する。実用上はこの方式に適するシステムが多い。待行列網で表現できるようなシステムがその典型で，コンピュータ，通信，交通，生産などの性能予測が含まれる。

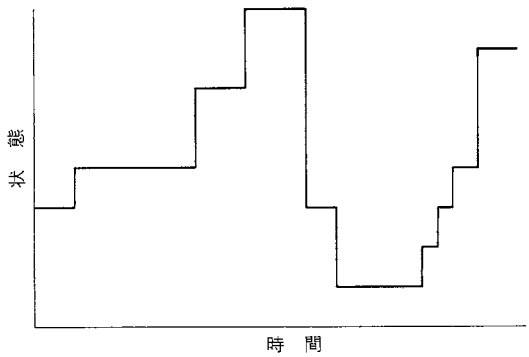

図1．離散事象シミュレーションの状態変化　離散的な時点でシステムの状態が階段状に変化する。
State transition in discrete event simulation

離散事象シミュレーションの扱う領域に対応した数学理論は，マーク点過程とかジャンプ過程という名で呼ばれている。その理論では階段関数で記述したシステムを数学的に扱いやすくするため，連続関数に書き直すことがよくある。それが流体近似や拡散近似である。この立場では，図1の階段関数を連続関数で近似していることになる。

2 原理

離散事象シミュレーションは階段関数の積分なので，その

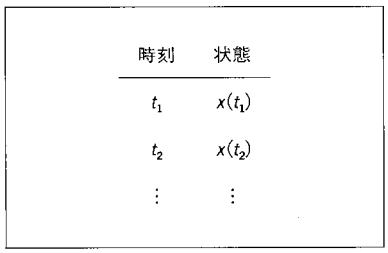

図2．状態の時系列に沿った列挙　シミュレーション開始時点から終了時点まですべて列挙する。
Time series enumeration of states

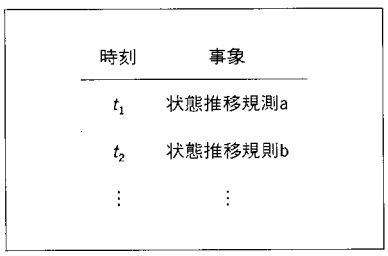

図3．事象リスト　離散事象シミュレーションでは，状態の時系列を得るためこの構造を使う。
Event list

実行には各事象の生起時刻とそのときに取る状態を決めることが必要である。それは，**図2**のようにシミュレーション開始時点から終了時点までにとる状態を時系列的にすべて列挙することにほかならない。

状態の時系列を得るために，離散事象シミュレーションでは**図3**のような事象リストという構造が使われる。これは個人が使う予定表のようなもので，将来起こることになっている事象が書いてある。第1行が最初に起こることである。ある予定を実行すると，それによって次の予定が発生し，予定表が更新される。例えば予定されていた会議に出席すると，その行為を実行することによって次回の会議の日程や報告の提出日が決まる。事象リストでも，ある事象が起こるとその事象に依存して起こることになった派生事象がリストに加わって，済んだ事象はリストから抹消される。例えば製品の搬送が始まると，搬送に必要な時間が経過した後に搬送終了という派生事象がリストに加わり，搬送開始という事象はリストから抹消される。

初期状態が与えられ，図3の事象リストの指定するとおりに状態を推移させれば，図2のすべての状態の一覧表が得られ，積分ができる。これが離散事象シミュレーションの原理である。

3 シミュレーション作業

離散事象シミュレーションの作業手順は，およそ次のとおりである。

(1) シミュレーションしたい対象のモデルを作成して，それが対象の調べたい性質を正確に反映していることを確認する。
(2) モデルのパラメータを決めるために，入力データを収集する。
(3) シミュレーションプログラムが，ソフトウェアとして信頼できることを確認する。
(4) シミュレーションを行う。
(5) シミュレーションの結果を解釈する。

以下では，これに沿ってそれぞれの技術を概観する。もっとも重要なのは最上流にあるモデル作成の工程であって，時間的にも費用的にも負担が大きい。

3.1 対象のモデル化と確認

モデル化では，調べたい性質を反映するようにシミュレーション対象の状態変化の因果関係を記述する。記述は最終的には離散事象シミュレータとなる。したがって，初めは抽象的な記述であっても，最後には事象，すなわち，いつ何が起こるという時刻と状態で表現する必要がある。

システム開発でシミュレーションを利用するには，設計情報を変換してシミュレーションモデルが記述できると都合がよい。設計情報をモデル情報に変換するには，両者で共通の言語が必要であり，そのような言語をモデル記述言語と呼ぶ。コントロールグラフ，ペトリネット，事象グラフ，形式的条件仕様記述，オブジェクトによる抽象化などがいろいろ試行されている段階で，標準的なものはまだない。

形式的条件仕様記述には，ある種のモデル化の誤りを自動的に検出する機能が簡単に付加でき，コントロールグラフは，よりシミュレーションプログラムに近い事象グラフへの変換が容易である。また，ペトリネットなら，アイコンを操作してグラフィックにモデルを作れるようにできる。当社でも目的に応じて種々の記述言語を利用，試行している。最近はシミュレータをプロトタイピング的に逐次開発する目的でオブジェクト指向モデリングを利用する例が多い。オブジェクトは独立性が高いモジュールなので，モデルを部分的に改造してもその影響が全体に波及しない。この特集でもオブジェクト指向に基づくシミュレータを紹介している。

シミュレーションモデルがその使用目的に耐えると判断を下すのは難しい。しかし，逆に使用目的に耐えないという判断の材料は比較的得やすい。例えば，モデルの動作をアニメーションにしてながめ，おかしなことが起これば誤っているのは明らかである。また，常識に合わない結果が出ればモデルが誤っていることが疑われる。したがって，誤りが明らかになりやすい状況をいろいろ設定して，それでも矛盾が出なければ，正しいと判断する。確認法の重要な部分は，誤りが明らかになりやすい，いじわるな状況の設定のしかたである。モデルをなんらかの極限的な環境，例えば異常に負荷が高い

とか，複数の事象が同時刻に生起するとかの条件で動作させてみると，誤りを検出しやすい。確認法のもう一つの重要な部分は，誤りを認識しやすいヒューマンインタフェースで，先に触れたアニメーションがもっとも注目されている。

3.2 データの収集

シミュレーションの対象システムがある環境のもとでどのような動作をするかの因果関係は，設計情報から記述できる。しかし，シミュレーションを行うには，そのシステムがどのような環境で動作するのかを表す設計情報にはないデータの収集が必要になる。データ収集の方法には二通りの考えかたがある。

一つは負荷の特徴付けと呼ばれ，システム動作のモデル化と同じように，システムの動作環境をモデル化しようというものである。例えば単位時間当たりの製品到着数を調べて，到着間隔の分布を想定し，シミュレーションにはその乱数を用いる。もう一つはトレース駆動と呼ばれ，モデル化を低いレベルにとどめておいて，徹底した実測で補完しようとするものである。例えば，類似の実働システムで製品の到着間隔を長時間にわたって実測し，その記録をそのままシミュレーションの入力とする。この方法は，トレース駆動シミュレーションと呼ばれている。

負荷の特徴付けのほうがデータ収集は容易で，トレース駆動のほうが信頼性は高い。

3.3 ソフトウェアの作成と確認

シミュレーションソフトウェアは，データの収集に続いて，あるいはそれに平行して開発する。開発には市販ないしフリーソフトウェアとして流通している離散事象シミュレーションのパッケージを利用することが多い。しかし大きなモデルを扱う場合は，ソフトウェアの信頼性だけでなく実行時間が問題になり，流通ソフトウェアが利用できない場合も多い。

実行時間はモデルの作りかたとプログラムの作りかたの両方に依存する。事象リストはモデルの大きさにほぼ無関係な時間で操作できるアルゴリズムが知られているので，問題は状態推移にかかる時間である。これはモデルに依存し，シミュレーション対象のシステムがうまく設計されていれば，モデルの大きさの対数オーダで収められることが多い。この特集で紹介している半導体生産工程シミュレータ DEUS はそのような設計である。

ソフトウェアの信頼性の問題は，一般にソフトウェアエンジニアリングの分野で扱われており，シミュレーションでもその成果が援用できる。なかでもオブジェクト指向が有力視されている。また，単にシミュレーション専用言語を使うだけでも，低レベルの汎(はん)用言語を使うよりも誤りをかなり減らせる。信頼性を上げる正攻法は，モデル化でうまい抽象化を行って，コード量を少なくすることである。

3.4 実行と結果の解釈

シミュレーションは計算機実験とも呼ばれるとおり，理論解析というよりは実験解析の技術である。必要な情報をどのように効率よく取るかが問題になり，実験計画法が有用である。

自然や組織が相手の実験よりも，計算機実験は実験条件を制御しやすい。例えば，システムの制御方法を他の方法と比較するとき，工場実験では制御方法以外の環境条件を同じに保てないのが普通であるのに対して，シミュレーションならまったく同じ条件下で比較ができる。このような，シミュレーションに固有な実験条件の制御しやすさを利用して，情報を効率よく取る技術が分散減少法や出力解析といった統計学の一分野として発達している。これらの技術は，まだ日常的に使うまでには普及していない。当社では特に大規模な実験には，共通乱数法やラテン超立方体法が使われている。

4 あとがき

シミュレーションは，設計工程の一部として位置づけるべきである。設計工程で評価の手順まで見通していれば，シミュレーションがスムーズに進む。システムを構築してしまってからどう評価するのかを考えるのでは手遅れである。

システムの設計工程にシミュレーションを組み込みにくい主な原因は，モデル化にシミュレーション特有の技術が必要であることと，シミュレーションの実行に多大な計算が必要になることである。負荷や性能のデータが整備され，設計情報からシミュレーションモデルが構成でき，高速なシミュレーションが可能になるにつれて，システム設計者自身がシミュレーションを実行できる環境は整いつつある。

この特集では，シミュレーションを設計工程に組み込む努力をしている当社の現状を紹介する。

文 献

(1) 特集：オブジェクト指向ソフトウェア開発，東芝レビュー，**48**，1 (1993)
(2) S. Morito, et al : eds. Proc. New Directions in Simulation for Manufacturing and Communications, Operations Research Society of Japan, Tokyo, 1994
(3) Tew, J.D., et al : eds. 1994 WSC Proceedings, ACM, IEEE, SCS, Lake Buena Vista, Florida, 1994

米田 清 Kiyoshi Yoneda

1974 年入社。性能評価などのシステム分析業務に従事。現在，研究開発センター システム・ソフトウェア生産技術研究所主査。
Systems & Software Engineering Lab.

田村 信介 Shinsuke Tamura, D.Eng.

1972 年入社。分散システム技術の研究・開発に従事。現在，研究開発センター システム・ソフトウェア生産技術研究所部長，工博。
Systems & Software Engineering Lab.