

ペトリネットのシミュレーションへの応用

[I]

—離散事象システムとペトリネット—

椎塚 久雄

1. はじめに

離散的な事象の発生で特徴づけられる離散事象システムの性質やシミュレーションに関する研究は、最近とくに注目されつつある。その1つの背景には、離散事象システムとしてとらえることのできるシステムがきわめて広範囲にわたっていることにある。離散事象システムシミュレーションは事象駆動型のプログラミングであり、システムにおける事象の発生がその中心的な役割を成している。

本講座では、このような離散事象システムをモデル化するのに最も適しているとされているペトリネットを用いたモデル化について解説し、さらにそのシミュレーションへの応用について、具体的な事例をもとにして、離散事象システムシミュレーションの問題に対するペトリネットの有用性を示そうとするものである。

事象駆動型システムの典型的な例としては、シーケンス制御で代表される機械加工・組立ライン、ロボット動作、化学、鉄鋼、電力プラントの運転過程などがあげられるが、このほかに計算機のソフトウェア、ハードウェア、オペレーティングシステム、通信プロトコル、作業計画・工程管理、およびオフィスシステムなども離散事象システムとしてとらえることができる。また、最近では銀行をはじめとする金融機関では、電子的な手段（オンライン）で預貯金・送金などの処理を行なっているが、これは現代が生んだ最も新しい離散事象システムであろう。

ところで、このような離散事象システムのシミュレーション言語としては、従来から GPSS, SIMSCRIPT などのいわゆる高級シミュレーション言語が使用されて

きた。しかし、これらの言語を用いたモデルの構築とシミュレーションは熟練を要し、一般のプログラミング言語（たとえば、FORTRAN, BASIC, PASCAL, C など）に熟達していても、現場の技術者がこのような高級シミュレーション言語を用いたプログラミングを行なうことは容易ではない。

ペトリネットの最大の利点はグラフィック言語であるということである。つまり、グラフィックに描くことによって、その構造がそのままモデルの言語に変換され、また変更、修正も自由に行なうことができることである。ペトリネットが単に従来の高級シミュレーション言語と比較して優劣をつけるというのであれば、ペトリネットをシミュレーション言語として使う価値は半減する。ペトリネットはより正確に離散事象システムをモデル化することができるということから [7]、その離散事象システムシミュレーションへの応用は自然な姿であろう。

ペトリネットをシミュレーション言語として使うには、適当なペトリネット・ツールが必要である。本講座では、最近筆者によって開発されたパソコンによるペトリネットモデル・シミュレーション・システム [9] を使用して、具体的な事例を交えながら、離散事象システムシミュレーションに対する新しい方向を探ってみたいと思う。

2. 離散時間システムの時間進行

シミュレーション言語としてのペトリネットの動作を理解するには、離散事象システムの時間進行の概念を把握するとよい。ここでは、離散事業システムの時間進行の概念を把握する上での鍵となる、離散時間システムの基本的考え方について述べる。

離散時間システムにおけるモデルの時間進行は、一定量で増加する時間のジャンプ列で、そのつど段階的に時

間が進む。一定増加離散時間システムは、映画フィルム動きを考えるとよい。すなわち、離散時間システムの時間進行は映画フィルムと同じ方法でそれをシミュレートしている。映画はあたかも連続的な動きをしているかのようにわれわれの目の錯覚をうまく利用しているが、実は毎秒24コマの割合でスクリーン上に投射する静止場面の系列を持つフィルムから成っているのである。

多くの設計システムは一定時間の増加で処理される。たとえば、銀行の利子は毎月、3カ月毎、半年毎、1年毎のいずれかで定期的に複利計算される。また、時間が一定増加で進む離散時間システムは、電子工学、制御理論、および通信などの分野における動的システムを記述するのによく用いられている。時間の関数は連続的な変化をするが、コンピュータ制御を行なう目的で、時変関数はサンプリング処理をして離散値のみが知られていることになる。

離散時間システムの最も簡単な例は、基本原理を定義したプロダクションシステムであろう。基本原理はプロダクションの集合 $A \rightarrow B$, $C \rightarrow D$ など定義され、矢線 \rightarrow の左側は右側に置き換え（あるいは書き換え）られることを意味する。規則を適用することはプロダクション列における各段階での時間進行と考えることができる。少し複雑なプロダクションシステムの例としては、チューリング機械があげられよう。

ボードゲームは近傍要素からの影響による2次元のプロダクションシステムである。ここで、細胞の群団の成長をシミュレートする生存ゲームと呼ばれるものを紹介しよう[6]。プレーヤーは2次元のグリッドに對面し、細胞の初期形状を定めるよう要請される。この形状は、各細胞の近傍に関して、新しい細胞の誕生および孤立細胞または過密細胞の死を支配する規則の集合の条件下で成長する。各細胞は生あるいは死（1あるいは0）の状態をとることができ、それらは周囲を取り囲む8個の細胞の近傍に位置することができる。与えられた細胞の近傍は、辺が共通の場合と角が共通の場合の2つの型がある。状態遷移規則は、

- (1) $0 \rightarrow 1$: 近傍に3個の細胞がある場合、
- (2) $1 \rightarrow 1$: 近傍に2個あるいは3個の細胞がある場合、
- (3) $1 \rightarrow 0$: 近傍に < 2 あるいは > 3 個の細胞がある場合の3つである。図2.1には生存ゲームの振動子パターンの進化の例を示す。これらの細胞の生成消滅のふるまい

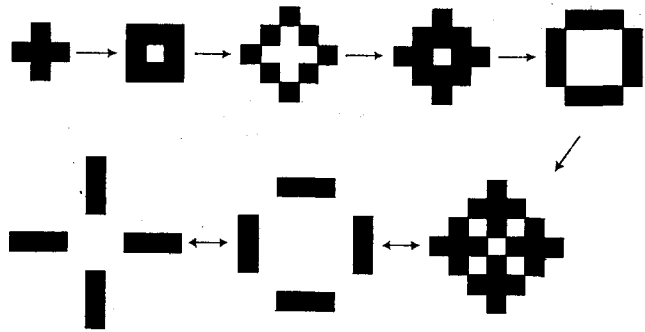


図 2.1 離散時間システムのふるまいを示す一例 (Conway [6] の生存ゲームの振動子パターン)

は、セルオートマトンと見なすこともでき、離散時間システムの概念を把握する好例であろう。

3. 離散事象システムとその特徴

離散事象シミュレーションでは、時間の増分をあらかじめ規定しないで、モデルの構成要素の活動にもとづいて、各時間ステップ毎に個々に決定される。事象は活動を開始したり終了したりする要素の順に発生する。すなわち、シミュレーションの全体的な活動は、それぞれの同時進行要素の流れに対するこれら事象の発生順にしたがって実行される。時間の離散化はシミュレータに依存するというよりは、むしろシステム自体に含まれていると考えるのがよい。したがって、離散事象シミュレーションプログラムは“活動の間にはさまれた事象”に関心があり、“活動的な仕事の期間”については比較的無関心である。たとえば、ロボットアームで部品のある位置Aから他の位置Bへ移動するシステムでは、部品がAに到着→部品をつかむ→移動→Bで離す、という事象が発生する。これを詳細なレベルで見ると、ロボットアームが部品を移動するさいにはその位置が連続的に変化するように、1つの事象もその発生期間中は連続的な変化を呈するかもしれない。しかし、離散事象システムでは、単に事象が生起するかしないかというレベルのみが問題であって、それ以下のレベルは考慮に入れない。

ここで用いる“事象”とは日常的な意味の特殊化したもので、それは単にきわだった出来事に関係している。瞬間の出来事概念と、かなりの時間持続期間を伴った概念を区別する。前者は“離散事象”(discrete event)あるいは単に事象と呼ばれ、後者は“活動”(activity)と呼ばれる。これらの違いは、1つの事象から次の事象へ変化するとき時間を進め、1つの活動から次の活動に

いくつかの要素が流れる、というように考えることができる。「活動」という用語は、要素が特別な方法でふるまっているということを含ませる必要はない。瞬間事象の量的な制限は、シミュレーションをしようとするシステムにおいて、モデル作成者がどの観点に重要度をおくかによっている。それに対する厳格な規則があるわけではない。

離散事象シミュレーションは、瞬時的な出来事あるいは事象の系列に依存する動的システムのモデルである。事象それ自体は、前の事象で実行されたシミュレーションの活動の中から発生し、状態が変化する全系列のシミュレーションを駆動する基本要素として用いられる。未来事象は未来事象集合に蓄えられる。時間の進行は未来事象集合に関する2つの基本的な操作で行なわれる：1つは、その集合が次の事象（すなわち、最小発生時間を持つ未来事象）を発生させるよう働きかけ、もう1つは、新しく導出された事象が未来事象の発生を待ち受けるために直ちに備えられる。したがって、離散事象シミュレーションは事象駆動型（event-driven）のプログラミングであり、基本的には活動に関する資源によって相互に関係する要素の並列的な流れから成っている。

いま、簡単のためにシステム内には2つの事象（それらを E_1, E_2 とする）が発生しようとしよう。 E_1 と E_2 の発生の仕方およびその相互関係として考えられるのは次のように要約できる。

- 発生の仕方⇒非同期性
- 相互関係⇒
 - 並行性（同時進行性）
 - 干渉
 - 先行関係
 - 競合

E_1 と E_2 の発生の仕方には非同期性がある。すなわち、 E_1 と E_2 は共通の時間軸に同期しているとは限らない。事象は条件さえ成立すれば生じうるものであり、 E_1, E_2 になんらかの理由により同期を強制しない限り、一般には時間軸に関して非同期であると考えてのが自然であろう。 E_1 と E_2 の間には干渉が存在する場合もあれば、あるいは干渉はなくどちらも独立に並行して発生する場合もあろう。 E_1 と E_2 の間の相互干渉にはいろいろな形が考えられる。先行関係とは E_1, E_2 のどちらが先に発生するかというものである。 E_1, E_2 のどちらも同時に発生しうる状況のとき、一方が発生すれば他方は発生できなくなるという競合性もよく見られる関係である。

以上のことから離散事象システムの特徴は、事象生起

の並行性、非同期性、および非決定性にある。このうち非同期性は上に述べた事象駆動型のプログラミングから明らかであろう。非決定性は、競合関係にある事象群のいずれを実際に生起させるかの意思決定をシステム管理者にまかせるものである。ペトリネットは、このような特徴をうまく表現し数学的解析を可能にする離散事象システムのモデルの1つである。

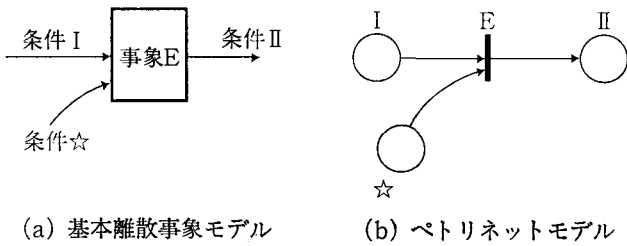
4. ペトリネットとは何か

システムの中の情報の流れを表わす一般的なグラフィックモデルは、1962年ドイツのC. A. Petri [1]によって提案された。その後、1960年代の後半、Holt [2]によってこのモデルは拡張され、彼はこのグラフィックモデルをペトリネット(Petri Net)と名づけた。ペトリネットは、非同期的でかつ並列的にふるまうシステムに対して、その中の情報の流れや制御を記述し解析するために考え出されたもので、いくつかの事象が並列的に発生する中で、それらの発生の順序、頻度などにある制約が与えられているようなシステムをモデル化するために主として用いられてきた[3]。

文献 [5] には、ペトリネットの初期段階の研究から最近のものまでの膨大な数の論文がリストアップされている。ペトリネットは、現在かなり拡張され、カラーペトリネット、確率ペトリネット、時間つきペトリネットなどのいわゆる高いレベルネットと称されるネットの研究が進みつつある。また、Reisig [4] は、条件/事象ネット、プレス/トランジションネット、述語/事象ネット、および関係ネット、というようにペトリネットに対して4つの解釈を与えている。

4.1 事象と条件の2つの側面からシステムをとらえる

連続事象システムのモデルの根本には、システム特性を入出力（原因→結果）関係においてとらえるという見方があり、よく知られているように伝達関数や状態方程式などはすべて入出力関係の表現である。したがって、離散事象システムにおいても、この入出力関係からシステム特性をとらえようとするのは自然な見方であろう。このような観点から見ると、離散事象の基本モデルは図4.1(a)に示す入出力モデルとしてとらえられる。これは、入力条件（条件I、条件☆）が成立すると事象Eが生起し、その結果新たに出力条件（条件II）が成立するというものである。この入力条件はもちろん単一であっても、さらに多くあってもよい。条件☆は外部システムからのなんらかの制御条件と考えてもよい。図4.1(b)は



(a) 基本離散事象モデル (b) ペトリネットモデル
 図 4.1 離散事象システムの基本モデルとそのペトリネット表現

基本離散事象モデルに対するペトリネット表現である。ペトリネットでは、条件をプレース (place) と呼ばれる丸印“○”で表わし、事象をトランジション (transition) と呼ばれるバー“|”で表わす。したがって、ペトリネットグラフにはプレースとトランジションの2種類の節点がある。これらの節点はアーク (有向枝→) によって結ばれているが、それはプレースからトランジションに向かっているか、あるいはその逆である。したがって、ペトリネットは有向枝を持つ2部グラフ (bipartite graph) の構造を有している。

4.2 ペトリネットの動的な性質

ペトリネットはグラフィックモデルであり、それがわれわれの視覚に訴え、複雑なシステムのあるまいをわかりやすくしようとするものである。フローチャートが計算機のプログラムの静的 (static) な性質をモデル化しているように、ペトリネットをグラフィックに表現することはシステムの静的な性質を表わしている。図 4.1 (b) はペトリネットの静的構造を表わしている。これに対して、ペトリネットは、それを実行して得られる動的 (dynamic) な性質も持っている。フローチャートによって表わした計算機のプログラムの実行を表示するために、実行中の命令を示すマークをフローチャートにおき、実行が進行するにつれて、そのマークをフローチャートに沿って動かすことを考える。これと同様にして、ペトリネットの実行はプレースに置かれたマーク [これをトークン (token) と呼ぶ] の位置とその動きとによって制御される。トークンは黒丸●で表わし、プレースの中に置く。プレースにトークンを割り当てることをマーキング (marking) という。一般にペトリネットでは、システムの初期の状態を表わすのに初期マーキングが割り当てられている。トークンの動きは発火規則 (firing rule) にしたがっている。図 4.2 にはペトリネットの要素と発火規則の適用例を示す。トークンがペトリネット

内を動きまわるようすはボードを使うゲームに似ていて、それは次のような規則にしたがっている。

- ① トークンは、ペトリネットのトランジションを発火 (firing) させることによりネット内を移動する。
- ② トランジションを発火させるためには、トランジションが発火 (enable) 可能でなければならない。トランジションのすべての入力プレースにトークンがあるとき、そのトランジションは発火可能である (図 4.2 (b))。
- ③ トランジションが発火すると、その入力プレースからトークンを取り除き、新しいトークンを生成してそれを出力プレースに置く [図 4.2 (c)]。

ところで、条件が成立しているということは、すべての入力プレース内にトークンがあり、そのトランジションは発火可能な状態、すなわち事象が生起する状態になっていることを意味する。このように、トークンがペトリネット内を発火規則にしたがって動きまわるようすはペトリネットの動的な性質を表わしている。(つづく)

文 献

[1] C. A. Petri : “Communication with Automata”, Supplement to Tech. Rep. RADC-TR-65-337, 1, Rome Air Development Center, Criffiss Air Force Base, N. Y., (Translation) 1965.
 [2] A. W. Holt and F. Commoner : “Events and

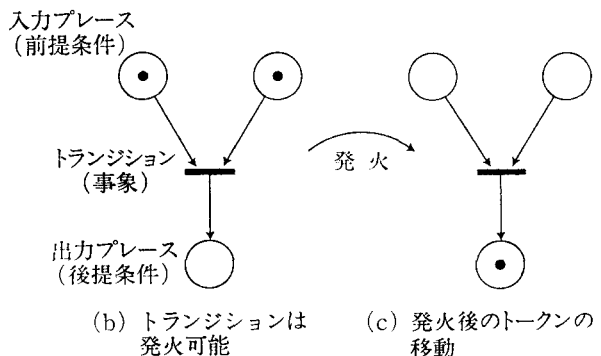
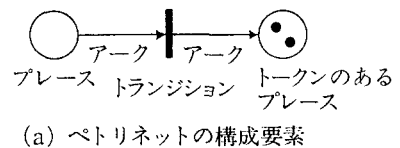


図 4.2 ペトリネットの構成要素と発火規則の適用

- Conditions”, New York: Applied Data Research, 1970; also in Rec. Project MAC Conf. Concurrent Syst. and Parallel Computation, pp. 1-52, New York: ACM, 1970.
- [3] J. L. Peterson 著, 市川惇信・小林重信訳: “ペトリネット入門”, 共立出版, 昭和59年.
- [4] W. Reisig 著, 長谷川健介・高橋宏治訳: “ペトリネット理論入門”, シュプリンガー・フェアラーク東京, 昭和63年.
- [5] S. Dreef, D. Gomm, H. Plunnecke, W. Reisig, and R. Walter: “Bibliography of Petri Nets”, G. Rozenberg(ed.). Advances in Petri Nets 1987, Lecture Notes in Computer Science 266, Springer-Verlag 1987, pp. 309-451.
- [6] M. Gardner: “The Fantastic Combinations of John Conway’s New Solitaire Game ‘Life’”, Scientific American, Vol. 223, No. 4, pp. 120-123, 1970.
- [7] J. B. Evans: “Structures of Discrete Event Simulation”, Ellis Horwood Limited, 1988.
- [8] 児玉慎三・熊谷貞俊: “離散事象システム”, 計測と制御, Vol. 24, No. 7, pp. 41-50, 昭和60年7月.
- [9] 椎塚久雄: “ペトリネットモデル・シミュレーション・システム”, オペレーションズ・リサーチ, Vol. 34, No. 8, pp. 433-438, 1989年8月.

APORS の論文誌 “APJOR” へのご投稿とご購読の依頼

皆様ご案内のとおり, 1985年から太平洋地区のOR学会連合 (APORS=Association of Asian-Pacific Operational Research Societies) が IFORS の下部機関として発足し, 日本のOR学会がその幹事役を努めることとなり, 若山邦紘教授 (法政大学) が事務局長に就任されています。

APJOR (Asia-Pacific Journal of Operational Research) は, その Official Journal という性格から, APORS 加盟各国から Board of Editorial Advisers へ参加することが求められており, 日本OR学会からは若山氏のほか森村英典会長, 茨木俊秀教授 (京都大学) が参加されています。これからも同誌を一層もり立ててゆくため, 論文の投稿・雑誌の購読についてご協力をお願いいたします。

1990年購読料 2,000円 (5月・11月発行予定)。

雑誌はシンガポールOR学会から貴殿宛直接送られます。

お申込みは学会事務局へ。(申込締切11月30日)

雑誌 EJOR 購読者募集

European Journal of Operational Research (EJOR)は, Association of European Operational Research Societies (EURO) と North-Holland 出版社との共同出版によるもので, 1990年は Vol. 43-47 が発行されます。個人購入もできますが, 当学会では割引価格でお取り扱いしています。

発行回数: 年15回 (5巻, 15冊)

使用言語: 英語

内容: あらゆる分野におけるORに関する優れた論文連絡事項として, letters や新刊書 (最近1年間のもの) の批評, 短評 (紹介)。

1990年購読料: 23,000円(予定) (送料込)

お申し込みは当学会まで。 (締切 11月30日)