# ペトリネットのシミュレーションへの応用 (III)

## ——待ち行列を伴うシステムのシミュレーション事例——

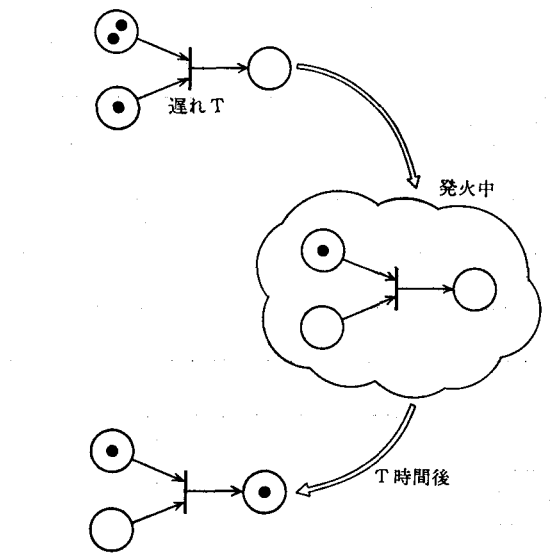
椎塚　久雄

今回は，ペトリネットの発火規則に時間遅れの概念を導入した**時間ペトリネット** (timed Petri net) を用いて，待ち行列を伴うシステムのモデル化とシミュレーションの事例を紹介する．

## 8. 時間ペトリネット

トランジションが発火する毎に，その発火ステップを1単位時間きざみの時間軸上でコントロールすることによって，ペトリネットに時間の概念を導入することができる．この場合，トランジションの発火に要する時間の最小値を1単位時間 ($U_t$と記す) とし，各トランジションでの発火時間遅れを$U_t$の整数倍とするものである．時刻$m$のときのマーキングを$M$とする．$M$のもとで$U_t$遅れのトランジション$t$が発火可能であれば，これを発火させることによりマーキングは$M'$に変わる．次の時刻$m+1$でこの新たなマーキング$M'$のもとで発火可能なトランジションを見つけ出しこれを発火させ，以下同様の方法で順次時間を進める．

一般に，遅れ$T$ ($T>1$) を持つトランジションの場合，このトランジションの発火によるマーキングの変化のさせ方には基本的に2つの方法が考えられる．その1つは，**図8.1** (a)に示すように，発火と同時にそのトランジションの入力プレースより1個のトークンを取り去り，$T$時間後にそのすべての出力プレースに1個ずつトークンを加える方法である．もう1つは，まず発火と同時にそのすべての入力プレースの中の1つのトークンを予約済みとして登録し，他のトランジションの発火に関与させないようにし，そして$T$時間後に予約を解除して入力プレースより1個ずつトークンを取り去り，すべての出力トークンに1個ずつトークンを加える方法である．この間トランジションは，図8.1 (b)に示すように，発火中という状態にあるものとする．




(a)　トランジション時間ペトリネット


(b)　プレース時間ペトリネット

**図 8.1**　時間ペトリネットの遅れの解釈

最初の方法ではトランジションの発火によるマーキングの遷移則が発火期間中は成立しないように見えるのでこれはトランジションに実際の行動を対応させる場合のモデル化に適している．また後の方法ではマーキングの変化が$T$時間後に瞬時に行なわれるのですべての時間でマーキングは遷移則を満たすように変化するから，これはプレースに実際の行動を対応させる場合のモデル化に適している．

しいづか　ひさお　工学院大学　電子工学科
〒160 新宿区西新宿1—24—2

このように時間ペトリネットは，時間に関するパラメータの導入の仕方によって大きく2つに分類され，上述した前者のように，トランジションの発火に要する時間を規定する場合を**トランジション時間ペトリネット**[11]，また後者のように，プレースに入力されたトークンが利用可能になるまでの遅延時間を導入する場合を**プレース時間ペトリネット**[12]という．さらにこれら両者をあわせたペトリネットモデルの研究もされている[13]．

一方，システムの確率的な事象の表現と解析を行なうために，この時間ペトリネットはさらに拡張され，トランジションが発火可能になってから発火を開始するまでの時間を，連続の確率分布を持つような確率変数で定義する**確率ペトリネット**（stochastic Petri net も提案されている[14]．文献[15]には，時間および確率ペトリネットに関する最近のサーベイがある．

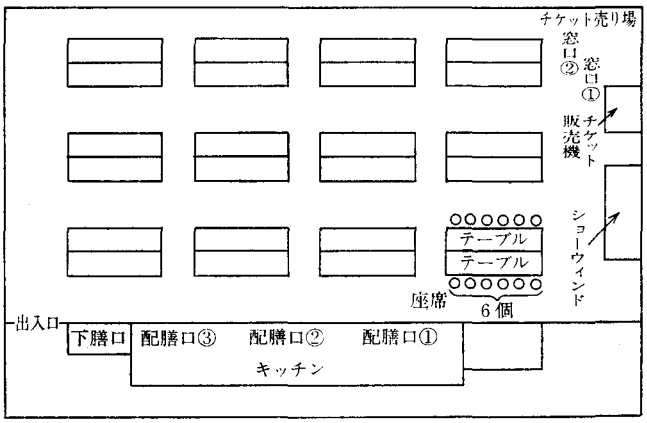



**図 9.1** モデル化する学生食堂の見取図

## 9. 学生食堂のモデル化とシミュレーション事例

時間ペトリネットの一応用例として，ここでは筆者の勤務する大学の学生食堂を取り上げ，食堂内における客の食事時間および流れを待ち行列を伴う同時進行システムとしてとらえ，ペトリネットモデルにより客の流れの状態をシミュレーションした事例を紹介する[16]．

### 9.1 時間進行とプレースタイマー

ここで考えるネットモデル上での時間は，基本的には8.で述べたプレース時間ペトリネットにしたがっている．しかし，ネットモデルの時間はトランジションが発火する毎に進めるのではなく，マーキングが$M$のとき発火可能なトランジションをすべて発火させた時点で単位時間として“1”進める．そして，発火が一周したときのマーキングを$M'$とし，$M'$の状態においても同様に発火可能なすべてのトランジションを発火させたら，また時間を“1”進める．このようにすることによって，疑似的な並列処理を行なうことができる．ただし，2つ以上のトランジションの発火が競合状態にある場合には，トランジションの番号が小さいものを優先して発火させることにする．これは競合トランジションをランダムに発火させても，ほぼ同一のシミュレーション結果が得られるためである．プレース時間ペトリネットを用いた理由は，食堂での客の食事時間を考慮するためである．したがって，使用するシミュレータの最も特徴的な点は，プレースでのトークンの待ち時間を導入したことであり，それをプレースタイマー（place timer）と名づけた．

たとえば，食堂のテーブルに対応するプレースにトークンが入ると，それは客が食事をするために席に着いたことを意味するが，このとき，その入力プレースとなっているトランジションは発火可能状態となり，これを発火させると，次の瞬間にはトークンが出ていってしまう．つまり，客が席に着いた次の瞬間すぐに席を立って出ていってしまうことになる．プレースタイマーは，この現象を避けるために，テーブルに対応するプレースに新しくトークンが入ると，指定したある一定の時間だけそのトークンを使ったトランジションの発火はできないようにコントロールするもので，各プレースにつけられたタイマーがモデルを実行するうえで有効な働きをしている．

### 9.2 ネットモデル化

**図 9.1** にはペトリネットによってモデル化しようとする学生食堂の見取り図を示す．客は出入口より入場し，ショーウインドにてメニューを選び，チケット売り場（窓口①，②，および自動販売機）で食券を買い，各自が選んだメニューの配膳口（①，②，③，…メニュー別になっている）へ進み，注文したメニュー品を受け取り，座席に着いて食事をとる．座席は1つのテーブルに12席，片側にそれぞれ6席ずつとなっている．客は自由に席を選ぶことができる．食事が済んだら下膳口にて下膳し，出入り口より退場する．

このシステムをペトリネットによってモデル化したものを**図 9.2** に示す．ただし，このネットモデルでは，食券売り場，配膳および下膳部分のモデルは簡単化するため

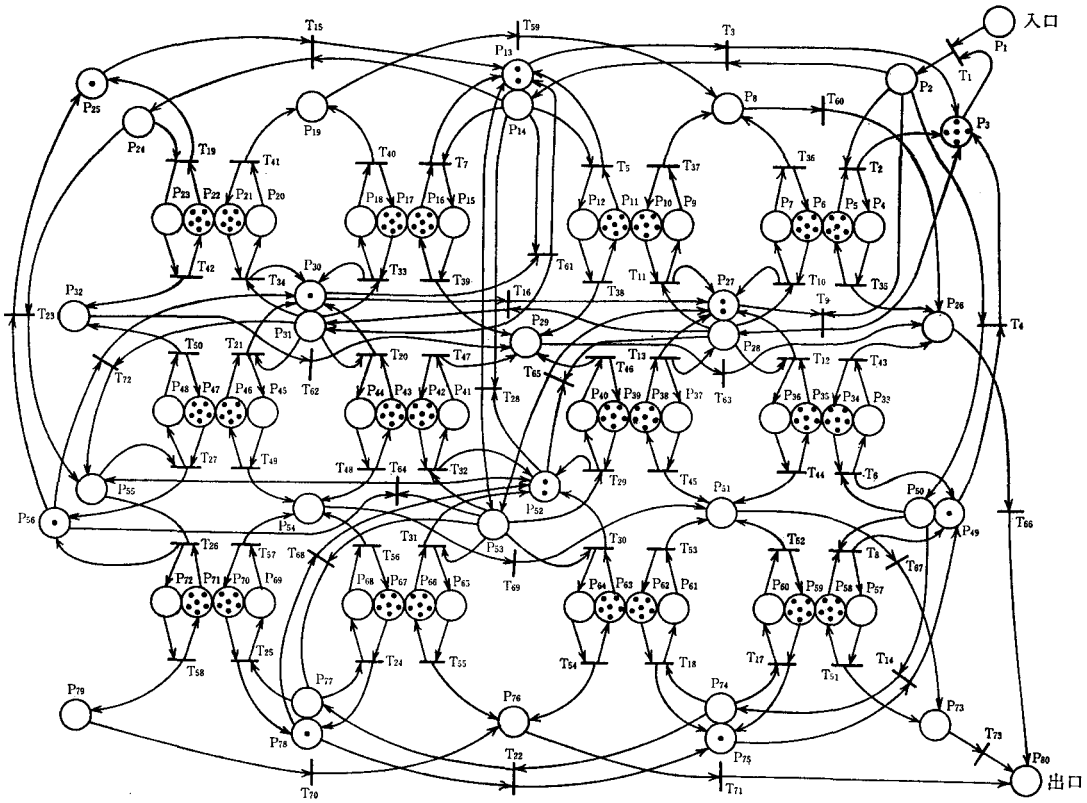



図 9.2　学生食堂のペトリネットモデル

に省略した．したがって，このモデルでは入り口にあたる場所では，客はすでに料理を受け取っているものと仮定している．

**座席のモデル化**：座席は1つのテーブルに12席あって，片側にそれぞれ6席ずつある，ここでは，立ったまま食事をすることは考えていないので，座席の数より客が多くなることはあり得ない．**図9.3**はこれをモデル化したもので，入口より客が入ってきて，トランジション $T_i$ が発火条件を満たし，それを発火させるとプレース $P_j$ の中にあるトークンはプレース $P_k$ の中に移る．すなわち，プレース $P_j$ の中のトークンの数は空いている座席の数を表わし，プレース $P_k$ の中のトークンの数は食事中の客の数を表わしている．この2つのプレースは対になって座席に役目を果たしている．したがって，プレース $P_j$ の中のトークンの数とプレース $P_k$ の中のトークンの数を加えたものはこの場合常に6に等しい．食事を終えた客は，プレースタイマーによって所定の時間になれば，トランジション $T_0$ が発火し対応するプレース $P_k$ のトークンはプレース $P_j$ に移り，同時に客は出口へと向かう．

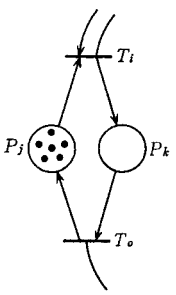



図 9.3　座席のネットモデル

**通路のモデル化**：通路をモデル化するときに注意しなければならないことは，客の動きをうまくコントロールできるようなモデルを作ることである．たとえば，一度席に着いた客が再び戻ってきて席に着いたり，あるいは席に着かずに出ていってしまうことがあっては客の流れをコントロールできない．そこで客の動ける方向を限定して，食堂内に入ってきた客は必ずどこかの席に着くようにし，席を立った客は必ず出口へ向かうようにした．したがってモデル上ではこれから食事をする客と食事を終えた客との経路は交わらないようにした．

図9.2においてプレースの対$(P_2, P_3)$,$(P_{13}, P_{14})$,$(P_{24}, P_{25})$,$(P_{30}, P_{31})$,$(P_{27}, P_{28})$,$(P_{49}, P_{50})$,$(P_{52}, P_{53})$,$(P_{55}, P_{56})$,$(P_{74}, P_{75})$,$(P_{76}, P_{77})$はこれから席に着こうとする客の通路上のある地点を表わしている．ただし，初期マ

ーキングにおいてすでにトークンの入っているプレースは，座席をモデル化した場合と同じように，その存在によってもう片方のプレースにはそのトークン数以下のトークンしか入ることができない．これによって通路上の一点に多数のトークンが溜ってしまうことを防ぐことができる．

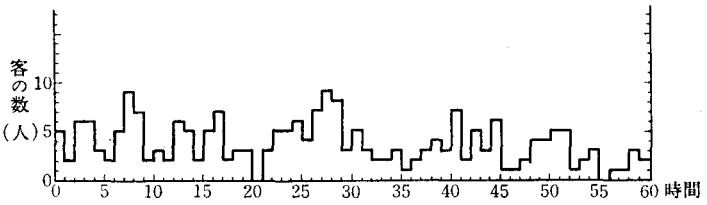



**図 9.4** 1分毎に到着する客の数

### 9.3 シミュレーションの実行

食堂内での客の食事時間およびシミュレーション開始時刻にすでに何人かが食事中であるかといういわゆる初期客も考慮し，種々の状況を想定してシミュレーションを実行することができる．食堂の入口に到着する客の待ち行列のデータは，1977年7月2日の12:00から13:00において生産機械工学科の矢部研究室が調査を行なった実測値を借用した．しかし，実際のシミュレーションでは，計算時間を短縮するために，実測データをもとにして各1分毎の客の数を計算し，それをその時間の到着客とした．したがって，客は全くランダムに到着するのではなく1分毎に何人かがまとまって到着することにした．**図9.4**に示すのは1分毎に到着する客の人数である．それ以外のデータ，たとえば客の食事時間等はすべて推測値を用いる．初期客というのは12:00現在に食堂内ですでに食事をしている客のことである．また，実行時間とは時計で計った実時間ではなく，発火可能なトランジションをすべて発火し終えたとき時間を1進めることにしている．モデル上での時間のことを意味する．プレースタイマーは，各プレースに6個設けた．食堂の片側のテーブルには6人まで座ることができるので，6個のタイマーはその6人に対応している．あるプレースのタイマーが0ならば，そのプレースを入力プレースとするトランジションはすぐに発火可能であるし，タイマーが3ならそのプレースを入力プレースとするトランジションはモデル上での時間が3進まなければ発火できない．

ここでは，次のような2つの場合のシミュレーション実行例の一部を示す．

**実行例1**：出口での混乱を避けるために出口へのアークの重みは3とした．テーブルのプレースにまだトークンの入る余裕があるのに，通路のプレースにトークンが溜ってしまうことがあるため，通路に関する初期マーキングを増加させ変更した．また，各テーブル毎のプレースタイマーを**表9.1**に示すように，10分から30分までとランダムに設定した．これは客の食事時間をみな一定とするより，客毎に食事時間が異なるようにした方がより現実的であるからである．初期客の食事時間は**表9.2**に示すように，客1と客2で時間1分から24分までとした．

シミュレーション結果を**図9.5**に示す．この結果からみるかぎりにおいては，この食堂にはまだ余裕があるように思える．

**実行例2**：実行例1の結果から，この食堂にはまだ余裕がみられるため，各時間の入口での到着客をすべて2人ずつ増加してみる．トークン数では約5割増しである．ただし，12:00の到着客だけは図9.4と同じである．

**図9.6**に示すように，モデル上の時間20以降の客はほぼ

**表 9.1** 実行例1のプレースタイマーのセット時間

| プレース No. | タイマー | プレース No. | タイマー | プレース No. | タイマー |
|---|---|---|---|---|---|
| 4 | 20 | 33 | 20 | 57 | 15 |
| 7 | 15 | 36 | 15 | 60 | 20 |
| 9 | 10 | 37 | 10 | 61 | 10 |
| 12 | 20 | 40 | 20 | 64 | 15 |
| 15 | 25 | 41 | 15 | 65 | 20 |
| 18 | 15 | 44 | 30 | 68 | 30 |
| 20 | 20 | 45 | 20 | 69 | 25 |
| 23 | 30 | 48 | 25 | 72 | 20 |

**表 9.2** 実行例1の初期客の残り食事時間

| プレース No. | 客1 | 客2 | プレース No. | 客1 | 客2 | プレース No. | 客1 | 客2 |
|---|---|---|---|---|---|---|---|---|
| 4 | 1 | 24 | 33 | 9 | 16 | 57 | 17 | 8 |
| 7 | 2 | 23 | 36 | 10 | 15 | 60 | 18 | 7 |
| 9 | 3 | 22 | 37 | 11 | 14 | 61 | 19 | 6 |
| 12 | 4 | 21 | 40 | 12 | 13 | 64 | 20 | 5 |
| 15 | 5 | 20 | 41 | 13 | 12 | 65 | 21 | 4 |
| 18 | 6 | 19 | 44 | 14 | 11 | 68 | 22 | 3 |
| 20 | 7 | 18 | 45 | 15 | 10 | 69 | 23 | 2 |
| 23 | 8 | 17 | 48 | 16 | 9 | 72 | 24 | 1 |

110から120の間で安定している．言い換えればこの時間で客の数は飽和している．したがって，だいたいこのあたりが許容人数の限界ではないかと考えられる．

以上示したモデル化とシミュレーションは，時間ペトリネットのみを用いて基本概念だけを例示した．さらにシステムの性能評価等を行なうには，時間・確率ペトリネットを導入することが望まれる．また，このシミュレーションはパソコン上で専用シミュレータを開発して実行した．しかし，パソコンでは規模が大きくなると処理速度が問題になる．システム性能評価用ソフトウェアパッケージの代表的なものとして，最近，Chiola らが ワークステーション上で開発した Great SPN があげられる〔17〕．

ところで，本稿で示した例でもわかるように，ペトリネットモデルはシステムをブラックボックス的に扱うのではなく，かなりホワイトボックスに近い状態で扱うことができる利点を持っている．　（つづく）

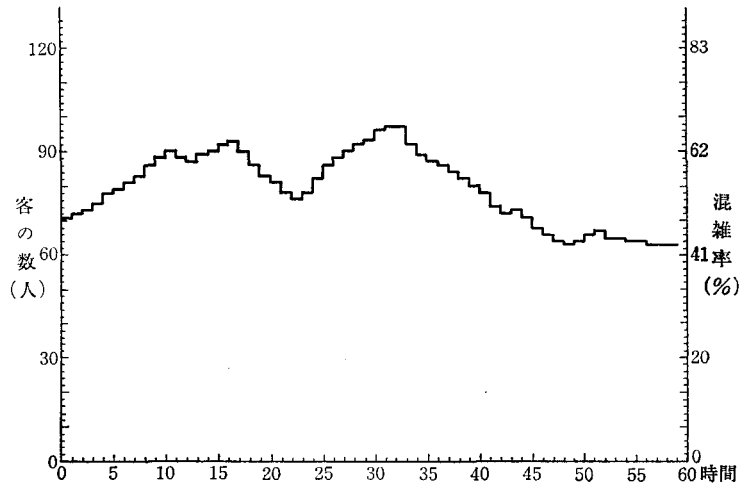



図 9.5　実行例 1 の結果

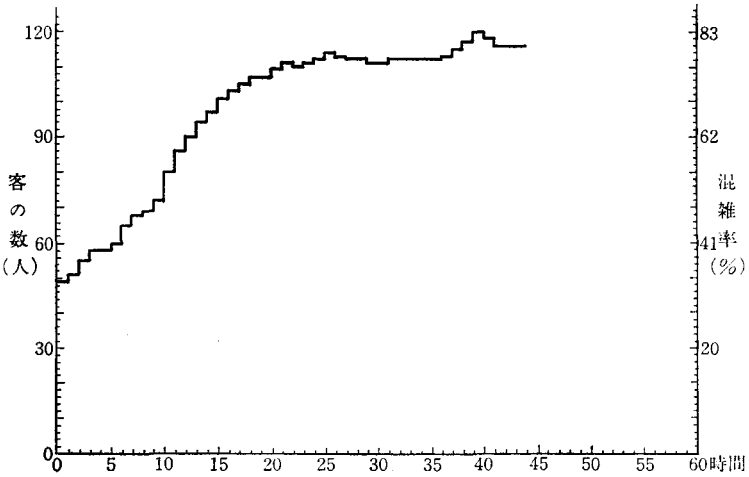



図 9.6　実行例 2 の結果

## 文　献

（前回までにあげたものは省く）

〔11〕 C.Ramchandani: "Analysis of Asynchronous Concurrent Systems by Timed Petri Nets", Ph. D. Thesis, Department of Electrical Engineering, MIT, 1974.

〔12〕 J. Sifakis: "Use of Petri Nets for Performance Evaluation", in Modelling and Performance Evaluation of Computer Systems,(H. Belner and E.Gelenbe(ed.)), North-Holland,pp.75-93, 1977.

〔13〕 R.R. Razouk and C.V.Phelps: "Performance Analysis Using Timed Petri Nets", in Protocol Specification, Testing and Verification VI, (Y. Yemini et al. (ed.)), North-Holland, pp.75-93, 1985.

〔14〕 M. Ajmone Marsan, G. Balbo and G.Conte: "A Class of Generalized Stochastic Petri Nets for the Performance Evaluation of Multiprocessor Systems", ACM Trans. Comput, Syst., Vol.2, No.2, pp.93-122, 1984.

〔15〕 西尾章治郎："システム性能評価のための時間および確率ペトリネット"，計測と制御，Vol.28, No.9, pp. 760-769, 1989年9月.

〔16〕 椎塚久雄・末永亮・中田努："ペトリネットによる待ち行列システムのモデル化"，電子通信学会ネット理論研究会論文集（第1回），pp.1-14, 1986年5月.

〔17〕 G. Chiola: "Great SPN User's Manual, Ver. 1.3", Dipartimento di Informatica, Universita di Torino, 1987.