

ペトリネットモデル・シミュレーション・システム

椎塚 久雄

1. ま え が き

現実のシステムはかなり複雑な振舞いをする場合が多いので、ペトリネットを用いたシステムの解析や性能評価およびシミュレーション等を行なうときには、単に紙面上のネットモデルだけでなく、適当なペトリネット・ツールがあれば、人間の手を煩わすことなくネットモデルの動的挙動を容易に把握することができることは言うまでもない。ペトリネットの理論的な側面の研究もさることながら、このようなペトリネット・ツールの開発に関する研究も、最近特に重要視されている [1, 2]。しかし、これらのツールの使用環境はワークステーションクラスの機種がほとんどであり、パソコンクラスのものは少ない [2]。パソコンクラスのツールの開発は、ペトリネットをより身近なものにすることができ、またネット理論の普及とともにその教育的効果にも大きく貢献できる道を開くものである。

本稿は、第4回日本オペレーションズ・リサーチ学会事例研究奨励賞（ソフトウェア部門）を受賞した「ペトリネットモデル・シミュレーション・システム」について述べたものである。

さて、本稿で紹介するペトリネット・ツールは、その取り扱いの容易さに重点をおいて開発したものである。本システムは使い勝手をよくするためにメニュー選択方法を採用した。これは、対話形式にした場合入力が非常に複雑になるという欠点があるためである。本システムはMS-DOSのもとでC言語を用いて開発し、画面上でネットグラフおよびネットの論理的な記述が自由な修正機能とともに可能にした。このことは、本システムの入力はほとんどマウスから行なうことができるので、紙面上に書くのと同じような感覚でネット構造を入力でき

ることを意味している。本システムが扱うことのできるネットの型は、プレース/トランジションネット（通常のペトリネット）と時間付きペトリネットである。本稿ではこのシステムの種々の機能の概略を順に述べるが、紙数の関係でソフトウェアに関する詳細は省略する。また、ペトリネットの基礎的事項については、たとえば文献 [3] を参照されたい。

2. システム構成

システム構成は、図1に示すように階層構造になっている。階層構造のプログラムは、作成が容易にできるのでこの方式を採用した。各部の機能等については5.および6.で述べる。

3. 画面設計

画面上には次のようなコマンドを表示させている。

- ペトリネットグラフ
- ポインティング用カーソル
- 命令選択用の文字
- 現在実行している命令
- 命令実行中の補助表示
- メッセージ表示
- ファイル一覧表
- ファイル名等入力欄

以下これらの機能について順に概略を述べる。

ペトリネットグラフ：画面上におけるペトリネットグラフは、なるべく多くのプレースやトランジション等を表示させなければならないので、図2に示すように、画面のほぼ大部分を占めている。また図3に示すように、プレースとトランジションは8×8ドットの単位で場所を指定する。これは、仮想画面という考えを取り入れたためである。画面上に表示されているプレースやトランジションの座標を記憶するための配列変数を仮想画面と名づけた。仮想画面の概念の導入は、きわめて簡単な操

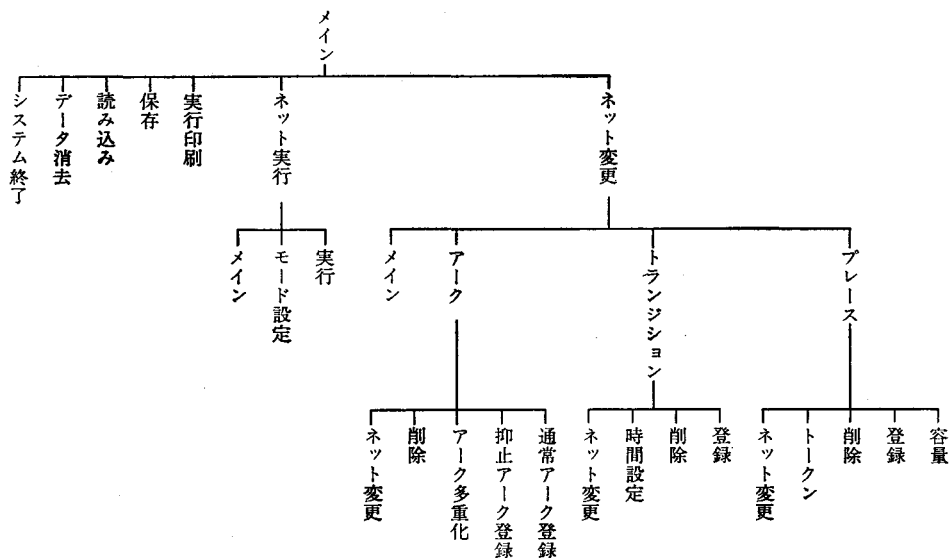


図 1 本システムの階層構造

作だけで、ネット構造の記憶ができ、即座にそのネットを稼働することができるという利点をもたらしている。

ポインティング用カーソル：カーソルは一般的な形で、白い矢印で表示する。

命令選択用の文字：命令を選択するための文字は画面の右端に表示し、その一番上には1つ上の階層の命令を表示した。

現在実行中の命令：実行中の命令というのは、階層の一番下であるので、それ以上表示するものはない。そこで、命令の文字の背景の色を赤色に変えて利用者にわかるようにした。

命令実行中の補助表示：実行中に補助的な情報が必要な命令がいくつかあるので（たとえば、トランジションの登録の時の直線の傾きなど）、この情報を命令選択用

のメニューの下に表示する。

メッセージ表示：エラーメッセージや利用者を与えなければならない情報を表示するために、画面の最下部にメッセージ表示領域を設けた。

ファイル一覧表：ファイル名の入力時にファイルの一覧表示があれば使いやすいため、ペトリネットグラフ表示領域を利用してその部分に表示させた。

ファイル名等入力欄：ファイル名、パス名、見出し文を入力するためにファイル一覧表の下の部分に、ファイル名等入力欄を設けた。

4. データ構造

ブレース、トランジション、アーク、入出力関数、仮想画面等のデータ構造の構造体はポインタでつながれて

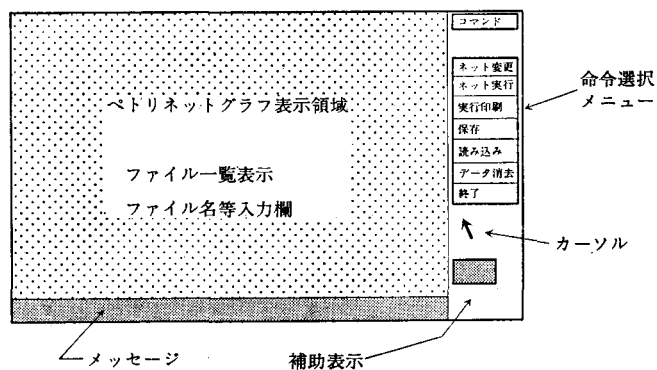


図 2 画面構成

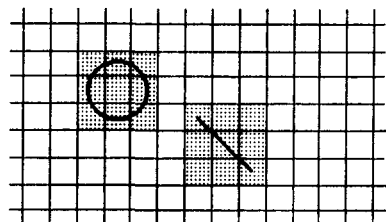


図 3 仮想画面と登録位置

いる。たとえば、プレースの場合は、プレース番号はプログラム中では存在せず、プレースの構造体へのポインタとして処理している。また、これらすべての構造体について共通していることは、データはリスト構造で形成されていることである。

プレース：プレースに必要なデータは、そのプレースの画面上の座標とトークンの数である。トークンはプレースの構造体の中に入れずにマーキングという構造体を別に作る方法と、構造体の中に入れる方法の2つが考えられるが、プレースとトークンを結びつけるにはプレース番号かポインタが必要であり、メモリ効率が悪くなるので、プレースの構造体の中に入れることにした。この他には前後の構造体へのポインタがある。

トランジション：トランジションにおいて必要なデータは、プレースに比べるとかなり多くなっている。まず、プレースと同様に画面上の座標と前後の構造体へのポインタに加えてトランジションの直線の傾きが必要となる。また、時間ベトリネットの機能を持たせるため、各トランジションには2つの時間 τ_1 、 τ_2 が対応づけられている。トランジション $\#$ は発火可能となってから少なくとも τ_1 時間たっちはじめて発火でき、 τ_2 時間内に発火しなければならない。この他に、トランジションに対応した入出力関数へのポインタ、およびネット実行時に時間付きベトリネットのための時間と発火可能フラグがある。

アーク：アーク1本のデータは何本かの線分で表現している。これは、曲線で表現しようとするアークを、いくつかの部分的な直線で折線近似をしているためである。このためアーク1本がそれぞれ大きさの違うデータを持っていることになる。そこで、はじめにアーク1本へのポインタをもつ構造体を定義し、アーク1本の中では、任意数の線分へのポインタをもつ構造体を定義する。すなわち2重のリスト構造になっている。

入出力関数：入出力関数に必要なデータは、あるトランジションに入出力しているアークの本数とそのプレースのポインタである。入出力関数に関するデータは、他のデータとは違い利用者が直接指定するデータではない。また、このデータが必要なときはベトリネットを実行するときだけであり、実行時にはアークのデータは必要としない。そのため、入出力関数のデータはネットを実行する直前にプログラムが作成するようにすれば通常必要なデータの数が減りメモリ効率を高めることができる。

入出力関数のデータ作成は、まず、すべてのアークについてその始点がプレースであればそのアークは入力関数を表わしており、その終点で指定されているトランジションの入力関数のデータに加える。逆に、始点がトランジションであれば、出力関数を表わしているのので、そのトランジションの出力関数のデータに加える。このようにすると入出力関数のデータ構造は同一のものを使用することができ、そのデータを入力と出力とで区別したリストを形成すれば良いことになる。

仮想画面：仮想画面はプレースとトランジションの検索時間の短縮をはかったもので、検索時間は登録された順に関係なく一定時間でしかも短時間に検索できる。また、プレースやトークンの画面上での重なり判定にも使用し、それらが重ならないようにしている。メモリ容量の関係から、仮想画面は 8×8 ドット単位としたが、パソコンクラスではこの単位が最も適していると考えたからである。しかし、この仮想画面はアークにはその性質上適用できず、アークの検索時間は登録されたアークの数によって時間を要することがある。

5. 実行アルゴリズム

本システムのプログラムの中で重要な部分のアルゴリズムの概要、すなわち、プレース、トランジション、アークの登録と削除、ネット実行時の入出力関数の作成や発火トランジションの選択などについて述べる。

プレースの登録と削除：初めにマウスのボタンが押されるまで待ち、左ボタンであればその座標を計算して仮想画面からその位置の情報を取り出してきて登録であれば何もないときに、削除であればプレースがあるときの各々の処理を行なう。

トランジションの登録と削除：トランジションの登録と削除も全体的な流れはプレースと同じである。登録の場合はマウスの右ボタンが押されたときにトランジションの傾きを変更する処理が入っているだけである。

アークの登録と削除：アークの登録はマウスの左ボタンが押されたときには、始点、中点、終点の各登録モードに応じて副関数を呼び出している。また右ボタンの時には登録モードを中点と終点で交互に変更している。始点の登録は座標を計算後、座標上にプレースかトランジションがなければならぬ。また抑止アークの場合はプレースに限られるのでこのチェックを行なう。中点の場合は単にアークの線分リストに座標を追加していくだけである。終点の場合は、始点の座標上にあるものによ

てブレースであればトランジションに、トランジションであればブレースになるようにチェックを行なう。

アークの削除は指定された点から各アークまでの距離を計算し、最も近いアークを削除する。これによって自然な操作が可能になっている。アークまでの距離の計算は、アークを構成している各線分までの最短距離を計算してそれをアークまでの距離とした。線分までの距離の計算は、指定点から線分に垂線を引きその交点が線分上にある場合とない場合で2通りの計算方法がある。

ネットの実行：まずはじめに入出力関数のリストをトランジションの構造体に追加し、時間を無視した状態で発火可能なトランジションがあるなら実行は終了しない。なぜなら、時間を進めていけばいつかは発火可能な状態になるからである。この条件をプログラム中では発火条件1と呼んでいる。そして条件2は、時間付きベトリネットで使用するもので、普通のベトリネットでは常に条件2は満たされている。これは、時間付きベトリネットにおける τ_1 、 τ_2 の条件を満たしているときに成り立つ。そして条件1と条件2ともに満たしているトランジションを実行モードにしたがって選び出し、このトランジションが競合していれば、実行モードにしたがって再び発火させるトランジションを選び出す。次にこのトランジションを発火させて1単位時間が終了する。

入出力関数はアークの始点と終点、仮想画面から作成する。すべてのアークについて、はじめに始点がブレースであればこのアークは入力関数を表わしているので終点で指定されるトランジションの入力関数のリストに始点のブレースを追加する。また、始点がトランジションであれば出力関数を表わしているのでこのトランジションの出力関数のリストに終点で指定されるブレースを追加する。

発火可能トランジションを捜すのは、すべてのトランジションについてまず入力ブレースのトークンの数が入力関数のリストで定義される数を満足していれば条件1を満たしているフラグを立てる。次に前回から条件1を満たしているトランジションであれば時間を進め、そうでなければはじめて条件1を満たしたので時間を0に設定する。次に、時間が τ_1 と τ_2 の間であれば条件2を満たしたフラグを立て、 τ_2 を越したら条件2を満たさなくなったのでフラグを降ろす。

6. システムの操作

本システムは特別な操作をしなくても使用できるように、なるべく複雑な操作は避けた。したがって、ネット

構造の入力から修正、実行等のほとんどの操作はマウスを使用して行なうことができる。マウスの左ボタンは実行や指定等の時に使用し、右ボタンは取り消しや変更等の時に使用する。命令を指定するときはその命令の表示されている文字上にカーソルを移動してマウスの左ボタンを押すだけでよい。もちろんキーボード上からの操作も可能である。

プログラムを起動した時の状態がメインで最上位の階層となり、図1に示すように、この状態では以下の6命令が使用でき、さらに階層を下げるによりすべての命令を使用できる。以下にそれらを示す。

(1) ネット変更：①ブレース、②トランジション、③アーク、の中から変更したいものを指定する。

①ブレースの変更では、登録、削除、トークンの数の変更、ブレースの容量の変更等ができる。登録の場合は、ベトリネット表示領域でカーソルを移動して登録したい位置で左ボタンを押す。削除の場合は、削除したいブレース上にカーソルを移動して左ボタンを押す。

トークンの数を変更する場合は、設定したいブレース上にカーソルを移動して、左ボタンを押すとトークンの数が増え、右ボタンを押すとトークンの数が減る。設定できるトークンの数は最大999個まで可能。

ブレースの容量を変更する場合は、上のトークンの数を変更する場合とほぼ同じ操作でできる。設定できる最大容量は999で、容量設定をしないブレースには最大999個までのトークンが入ることができる。

②トランジションの変更では、トランジションの登録と削除、および時間設定ができる。登録の場合は、ベトリネット表示領域でカーソルを移動して登録したい位置で左ボタンを押す。このとき補助表示領域に表示されるトランジションの傾きを右のボタンで変更できる。削除の場合は、削除したいトランジションにカーソルを移動して左ボタンを押す。

また、時間設定を選択すると、トランジションの上下に2つの数字(τ_1/τ_2)が表示される。トランジション i は発火可能となってから少なくとも τ_1 時間たつてはじめて発火でき、 τ_2 時間内に発火しなければならない。設定したいトランジション上の数字にカーソルを移動して、左ボタンで時間の増加、右ボタンで減少させる。

③アークの変更では、通常のアークと抑止アークの登録、アークの多重化、およびそれらの削除ができる。

アーク登録の場合は、通常のアークと抑止アークの登録方法は同じである。アーク多重化を行なうと、その失

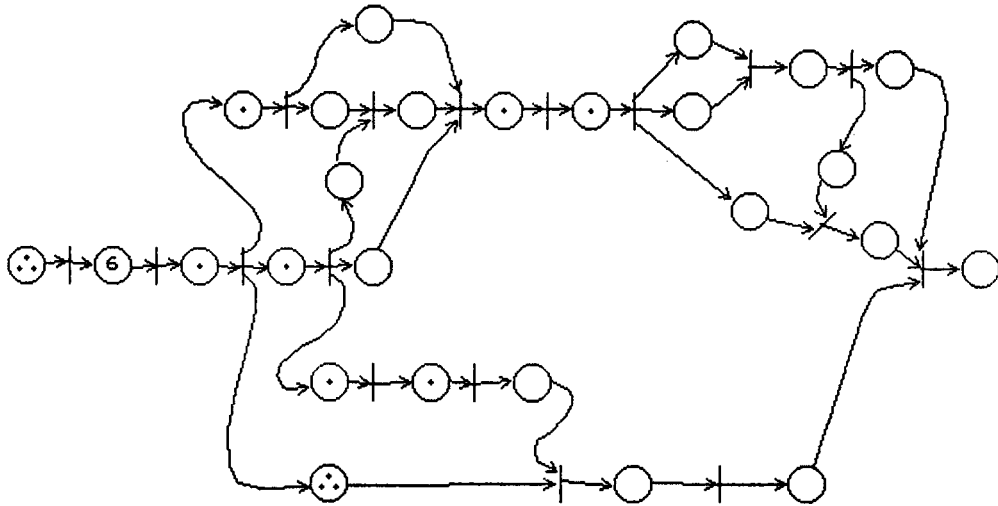


図 4 グラフィックエディタによるモデル作成例

印の近くに多重度を表わす数字が表示される。多重化したいアークの線分であればどこでも良いので、カーソルを移動する。左ボタンを押すと多重度が増加し、右ボタンで減少する。アークを削除する場合は、削除したいアークの線分であればどこでも良いので、カーソルを移動し左ボタンで削除する。

(2) ネット実行：ペトリネットモデルのシミュレーションを実行する「実行」と、実行中に発火させるトランジションの選択条件を設定する「モード設定」がある。このモード設定では、実行するときの条件、すなわち、発火後の待ち時間、発火トランジション、競合トランジションを指定できる。パラメータの種類は次のものがある。

- 発火後の待ち時間：無し／短め／普通／長め／ボタン指定
- 発火トランジション：番号順／ランダム／ボタン指定
- 競合トランジション：番号順／ランダム／ボタン指定

(3) 実行印刷：ペトリネットグラフと実行の各過程のマーキングを印刷する。グラフの印刷は2種類あり、初めのグラフはプレースとトランジションの番号との対応が分かるようにその中心に番号を印刷し、次に実際の初期状態のペトリネットグラフを印刷する。

(4) 保存：作成したペトリネットグラフの構造のデータをフロッピーディスクに保存する。

(5) 読み込み：保存したペトリネットのデータを読み込む。

(6) データ消去：現在のペトリネット表示領域にあるペトリネットのデータを消去する。この命令は、画面上から

消去するだけでなくデータも消去する。

7. 応用が期待される分野

ペトリネットモデルは、並列処理のいろいろな性格について検証することができる[3]。たとえば、安全性、有界性、保存性、活性、可達性と被覆性、およびデッドロックの検証等である。本システムはこれらの問題に対する支援ツールであることは言うまでもないが、特に、オペレーションズ・リサーチに関連したシミュレーション応用分野として主に次のようなものが挙げられる：

- (1) FMSのモデル化とシミュレーション。
- (2) 待ち行列システムのモデル化とシミュレーション。
- (3) オフィス・オートメーションの最適運用のための解析とシミュレーション。
- (4) 通信プロトコルの解析と設計およびシミュレーション。
- (5) スケジューリングの解析とシミュレーション。

たとえば、パート線図を用いたスケジューリング(文献[3], p.71)の問題を本システムを使ってモデル化したものを図4に示す。これは時間付きペトリネットでモデル化されている。本システムを使うことにより、種々の条件のもとでシミュレーションができる。

8. むすび

本稿では、ペトリネットモデル・シミュレーション・システムの概要について紹介した。さらに改良すべき点はいくつか挙げられるが、パソコンクラスによるペトリ

ネット・シミュレータとしては、初期の目的を満足しているのではないと思われる。

ペトリネットは、近年注目されつつある離散事象システムをモデル化する有力なツールである。本ソフトウェアによって、離散事象システムに興味を持たれる方が1人でも多く増えることを願っている。

文 献

- [1] K. Jensen: "Computer tools for construction, modification and analysis of Petri Nets", W. Brauer, W. Resig and G. Rozenberg (ed.), Petri Nets: Applications and Relationships to Other Models of Concurrency, Lecture Notes on Computer Science 255, Springer-Verlag 1987, pp. 2-19.
- [2] F. Feldbrugge and K. Jensen: "Petri net tool overview 1986", *ibid*, pp. 20-61.
- [3] 市川, 小林共訳: "ペトリネット入門", 共立出版(昭59).
- [4] 椎塚, 秋葉: "ペトリネット・ツール: SANET", 電子情報通信学会ネット理論研究会論文集(第3回), pp. 26-33, 1988年2月.
- [5] 椎塚, 秋葉: "ペトリネットモデル・シミュレーション・システム", 日本オペレーションズ・リサーチ学会秋期研究発表会アブストラクト集, X-4, pp. 218-219, 1988年9月.
- [6] 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—システム概要と使用説明書— Ver. 3. 21", 1988年12月.

「事例研究」の原稿募集

ORの特徴は実践にあるといわれています。実際的な応用をぬきにした理論ということはORでは考えられません。本誌でも以前から会員の皆様からの事例研究の報告をお願いしてきましたが、まだ十分な成果をあげているとはいえません。

もっと気軽に、「この問題はこう処理したが、もっとよい方法はないか」、「やってみただけでなかなかうまくいかない」というような事例や問題提起をどしどししていただきたいと思います。会員同士の知恵の交換というつもりでこの欄へのご投稿をお願いします。

投稿要領：学会原稿用紙36枚（25字×12行）以内（図・表を含む）

投稿先はOR学会事務局OR誌編集委員会宛。

なお、原稿の他コピーを2部添付してください。

（OR誌編集委員会）

「論文・研究レポート」の原稿募集

ORの実践をわかりやすい事例を中心に紹介してほしいという会員からの要望がある一方で、OR理論の展開あるいは手法の開発など学術的な研究報告も忘れないでという注文も根強くあります。

本誌では「論文・研究レポート」という審査論文欄を設けております。この論文・研究レポートでは、特に、経営の実践に役立つ理論研究、手法あるいはシステムの開発、概念フレームおよび方法論等を扱った研究のご寄稿を歓迎いたします。

投稿要領：学会原稿用紙36枚（25字×12行）以内（図表を含む）

投稿先はOR学会事務局OR誌編集委員会宛。

（OR誌編集委員会）

なお原稿のコピーを2部添付してください。