計測自動制御学会東北支部 第238回研究集会(2007.10.18)
資料番号 238-11

フラクタルペトリネットシステムに関する研究

A Study on Fractal Petri Net Systems

○渡部公樹*, 渡部慶二**, 村松鋭一**, 有我祐一**, 遠藤茂**

○ Koji Watanabe*, Keiji Watanabe**, Eiichi Muramatsu**, Yuichi Ariga** , Shigeru Endo**

*山形大学大学院理工学研究科, **山形大学工学部

**Yamagata University

キーワード：ペトリネット(Petri Net), 離散事象システム(Discrete Event Systems),
フラクタルペトリネット(Fractal Petri Net)

連絡先： 〒992-0037 米沢市城南 4-3-16 山形大学 工学部 応用生命システム工学科 渡部研究室
渡部公樹, E-mail: dkd91751@dipfr.dip.yz.yamagata-u.ac.jp

1. はじめに

近年，コンピュータを用いた制御システム構築において，社会の要請に伴い，システム構築の大規模化，複雑化が進んでいる一方で，制御システム構築期間の短縮化，開発コスト削減，高品質化が求められている現状がある．

しかし，現在の制御システム構築言語として，C 言語や C++言語などで記述されているのが一般的であり，プログラム作成には，専門的知識が必要となり，プログラミング初心者にとっては困難な作業となる．またプログラミング知識を有した者にとってもシステムが複雑化・大規模化になるほど，その内容を記述するプログラムも複雑になるため，相当な時間と苦労を強いられてしまう．さらに，どんなシステムでも潜在的なエラーを持っており，システム構築直後はもちろんのこと，運用時にもそのエラーが顕在化し運用停止となることが度々起こる．復旧させるまでの時間を可能な限り短縮するためには，保守と改変が容易にあることが望まれる．

上記の問題点を対処するためには，以下の 3 点の解決が求められる．

1) 容易に制御システム内容の記述が可能であること．
2) 視覚的に制御システム動作を確認可能であること．
3) 制御システムの保守・改変が容易であること．

これらの解決法の１つとして，ペトリネット理論をアルゴリズムとして動作するソフトウェアを開発し，コンピュータネットワークシステム構築をすることを本稿の目的とする．ペトリネットは，離散事象システムの並列的，非同期的，分散的なシステムを表現する数学的モデリングする最適なモデル化手法であり，システム

の状態遷移を視覚的に把握することができる．このため，制御システム内容の記述・把握が容易となり，また，視覚的に制御状況を確認できるため，何処かでエラーが発生した際，迅速に把握でき，保守・改変が容易になる．これより，上記問題点が解決可能となる．

本研究では，著者らがこれまでペトリネット理論を用い構築してきたネットワークペトリネット環境Network Petri-net for Java(NPJ)[4]を拡張し，ペトリネット図にフラクタル性を持たせることを提案したFPNJ(Fractal Petri-Net for Java)を構築した．本システムでは，制御系が複雑・大規模化した際にも，制御系を容易に表現し，状況把握しやすいシステムを構築することが可能になり，また，ペトリネット（離散事象システム）で起こりうる問題点（競合，デットロック）の解消を図る．

2. 離散事象システムとペトリネット

本研究では，システムを離散事象システムとして扱いペトリネット理論を用いている．ここで，離散事象システムとペトリネットについて説明する．

2.1. 離散事象システム

離散事象システムとは，以下の特徴を持つシステムの総称である[5]．

1)システムが離散的状態で表現される．

システムがもつ状態の集合が，離散集合として与えられる場合をいう．（たとえば，各状態変数のとりうる値が離散値であり，それらの値の組み合わせによりシステムが定義できる場合(スイッチのON／OFF)や，モデル化の場合に本来連続値をとる状態を離散的に扱う場合がある(センサの値は連続的に変化するが，閾値を設定した場合，状態変数を考えればよい)）．

2)システムが事象駆動である．

システムが，任意のタイミングにより発生する事象によって状態変化するシステムを時間駆動であるという．

また，上記2点から，下記4点があげられる．

1) 初期状態と遷移可能な状態の間には線形性が成立しない．
2) 状態遷移は並行的に生起しうる．
3) 状態遷移は非同期的に生起しうる．
4) 状態遷移は非決定的に生起しうる．

このような離散事象システムは，計算機システム，通信システム，生産システムなどの多岐にわたり存在している．

2.2. ペトリネット

2.2.1 概要

ペトリネットとは，並行的・非同期的・分散的なシステムを表現するための数学的モデルであり，離散事象システムのモデル化に広く用いられる．離散事象システムの解析設計に必要なモデルを考える際に大事なことは，1)直感的に理解しやすいモデルと2)検証・性能評価などの解析シミュレーションが可能な数学的モデルであり，上記2項をペトリネットにおいて実現可能である．

2.2.2 ペトリネット

ペトリネットとは、以下の四要素で構成されているグラフィックモデルであり、トークンの移動によるプレース配置組み合わせの変化で状態遷移を表すモデルである。

つまり、トークンのプレース配置組み合わせが、一つの状態を表している。

① プレース:条件 ○
② トランジション:事象 |
③ アーク:要素の流れの方向 →
④ トークン:要素 ●

あるトランジションに対して、入力アークがつながっている入力プレースの全てにトークンが入っている時(すべての条件が満たされている時)、トランジションは「発火可能状態」と言い、トランジションが発火するとトークンはトランジションに奪われる。その後トークンは出力プレースに移り、次の条件を満たしている状態になる。

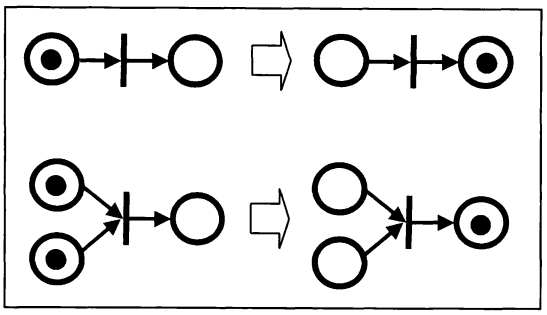

Fig.1: Movement example of the Petri-net

3. FPNJ の概要

本ソフトウェアは，ペトリネット理論の特性である並列処理表現が必要であり，また，汎用性の高いソフトウェア構築を目標とする．これらの要件を満たす言語として，マルチスレッド処理記述が容易あり，またマルチプラットフォームで動作する Java 言語にて構築した．

Fig.2 , 3 にプログラム起動時の初期画面を示す．起動時に edit window （Fig.2） と option Window (Fig.3)が生成され，option window のペトリネットの各要素をクリックし，edit window をクリックすることにより，ネット図の記述が可能となる．

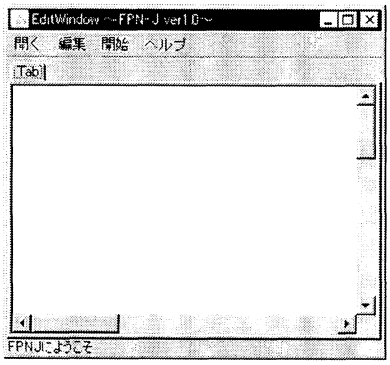

Fig.2: initial screen of FPNJ(1/2)

Fig.3: initial screen of FPNJ(2/2)

以下に FPNJ で拡張した機能について説明する．

3.1. セーフティペトリネット

ペトリネット理論では，プレースが保有するトークンの個数には上限が無い．しかし，プレースの個数に上限がないため，競合が起こる可能性が高くなる．その解決策の一つとして本システムでは，競合が起きにくいセーフティペトリネットの構成を提案し，下記の点を定義した．

- プレースに最大１つのトークンの存在しか許さない．
- プレースからトランジションに接続するアークを１つしか許さない．

しかし，セーフティペトリネットでネット図を表現した場合，従来のペトリネットの幅広い表現性が失われてしまう．その解決策として，以下の点を定義した．

- トークンにデータ保持機能と Tag 機能を拡張
- トランジションに Tag 認証機能

このように，セーフティペトリネットとトークン，トランジションに Tag の概念を導入することにより，ネット図の安全性の向上が可能となる．

3.2. フラクタル表現

制御システムを通常のペトリネット理論で表現した場合，ペトリネット図は視覚的に表現できるが，プレース，トランジション，アークのみの表現となるため，直感的にわかりにくいものとなる．また，描画したネット図に誤りがある場合，誤りが発生している箇所を特定する事はとても困難となる．その解決策として，ネット図にフラクタル性を持たせ表現することを提案する．以下に本ソフトウェアのフラクタル表現について説明する．

3.2.1.フラクタル構成定義

ある仕事要素に対するペトリネット図のフラクタル表現を行うため，以下3点を定義した．

1) ネット図の最小構成により，フラクタルの自己相似性を表現する．
2) トランジションの階層ネット図は，1)で定義したネット図を用い，その階層のネット図も同様とする．
3) 仕事要素部分を表現するネット図は，1)で定義したネット図で構成する．ただし，構成が不可ならば，システムにあったネット図で構成する．

上記定義をもとに，Fig. 4にフラクタル性を持たせたネット図の例を示す．Fig. 4のパネル描画部に配置したトランジションは，図中階層レイヤ1のような自己相似性を持つペトリネット図をトランジションの階層として埋め込んでおり，また階層レイヤ2には，実際の仕事要素を表現するネット図を埋め込む．このとき，レイヤ2のネット図は仕事要素内容よりネット図の構成は異なるが，レイヤ1はどのトランジションでも同じ構成をとる．
このように，フラクタル表現を導入することで，ネット図設計がより容易になり，また，描画トランジションの階層ネット図の構成は，レイヤ1の構成のように，どのネット図も同じ構成要素から成り立つため，ネット図が複雑化，大規模化した場合の保守において，ネット図のすべての構成を確認せず，エラーが発生しているトランジションの階層内容を確認することで対処でき，迅速にエラー修正が可能となる．

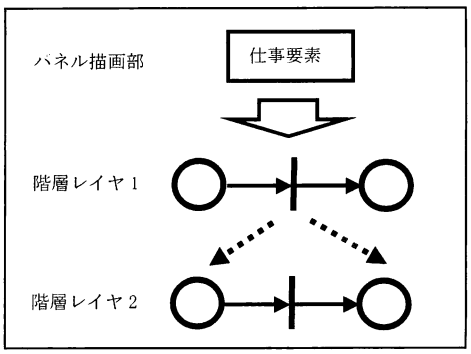

Fig.4: Fractal expression of the Petri-net

3.2.2.システムへのフラクタル表現適用

実際のシステムを構築する場合，仕事要素が集合しシステム構築がなされる．ペトリネットを用いてシステムを表現した場合，上記同様に考えることができる．本ソフトウェアでは，仕事構成をオブジェクト化として表現することを目標にし，そのオブジェクト化にフラクタル表現を利用して表現することを提案する．仕事のオブジェクト作成を行うため，以下に定義を示す．

1) 2.2.1.の定義1)，2)を満たすこと．
2) 仕事オブジェクトを作成したいネット図において可達性問題が解決できること．

上記定義をもとに，Fig.5に仕事をオブジェクト化したペトリネット図の例を示す．図中の階層レイヤ2のP1，P2は，プレースの配置番号を示しており，P1からP2の可達性問題は解決しているものとする．ここで，P1からP2まで

のネット図を仕事オブジェクトに格納することを目的とする.

前項同様, 図中パネル描画部に配置したトランジションの階層に階層レイヤ 1 を構成し, レイヤ 1 のトランジションの階層には, レイヤ 2 が埋め込まれている. 実際の描画では, 図中のパネル描画部のようなトランジションの配置のみで, システムを作成できる.

このように, 仕事オブジェクトにフラクタル表現を導入することにより, ネット図の作成は容易になる. また, 仕事オブジェクトを生成する仕事要素にもフラクタル表現を導入しているため, ネット図の構成に統一性ができ, エラー発生時にも容易に処置することが可能になる.

また, 仕事オブジェクトは仕事要素の変更はあるが, ネット図自体に変更がある可能性は低い. そのため, 一度作成したオブジェクトをライブラリとして保存する.

ライブラリ機能に関しては, 次節で説明する.

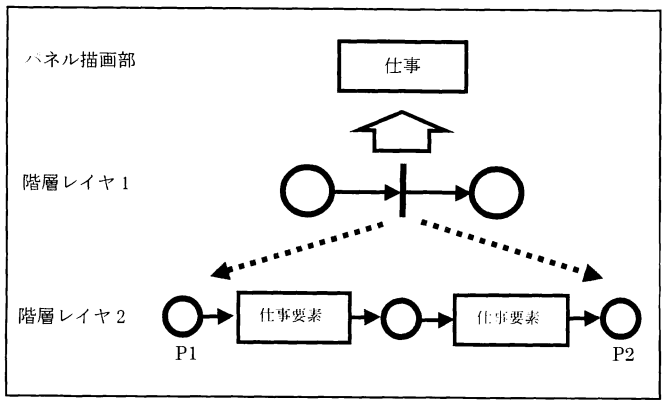

Fig.5: Object expression of the fractal Petri-net

3.3. ライブラリ機能

一般的に, システムが複雑化した場合, システムを描画したペトリネット図もまた複雑な構成をとる. その為, 視覚的にシステム状況把握しやすさが失われ, システムの管理が困難となる. その解決策の一つとして, 本ライブラリ機能と前項のフラクタル表現により解決する.

本機能では, ペトリネット描画をする上で頻繁に使用するネット図（フラクタル表現）やAND・OR・分岐機能をトランジションの階層ペトリネットとしてライブラリとして保存し, Fig. 6 のように, ネット描画時にはトランジションを 1 つのタスクとして描画し (図左), その階層には図右のネット図が格納されている.

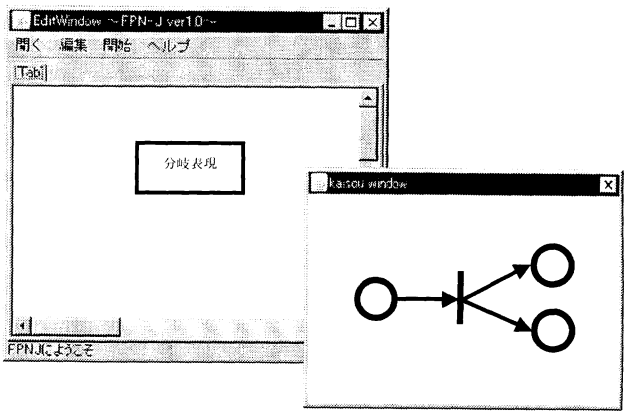

Fig.6: drawing example of library

これにより, 描画上は階層構成を持つトランジションの描画となり, システムが複雑になった場合において, ネット図の複雑化を緩和できる. 以下に, 初期定義した AND, OR, 分岐機能について説明する.

1) AND 機能

Fig.7 にペトリネットの AND 表現示す. 1 つのトランジションに対して, n 個（n:2,3,4…,n）のプレースがアークでつながれていおり, 全てのプレースにトークンが入った場合のみトランジションを発火させる動作を作成したいとき, AND 機能を使用する.

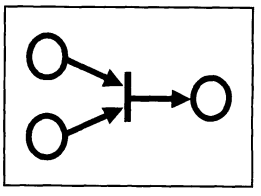

Fig.7: AND expression of Petri-net

2) OR機能

Fig.8にペトリネットのOR表現示す．1つのプレースに対して，n個（n:2,3,4…,n）のトランジションがアークでつながれており，n個のプレースのうち少なくとも1つのプレースにトークンが入った時点で，トランジションを発火させる動作を作成したいとき OR 機能を使用する．

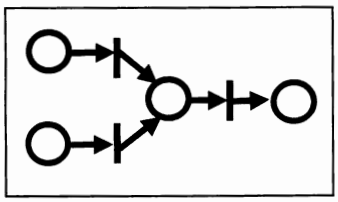

Fig.8: OR expression of Petri-net

3) 分岐機能

1つのトランジションでプレースを 2 個以上接続されている場合，トランジション発火後，接続されているプレース全てにトークンを渡す．しかし，ネット図を構成する上で，条件分岐表現が課題となり，従来のペトリネットにて条件分岐を表現することは困難となる．その解決策として条件分岐を行うトランジションを設置することを提案する．分岐トランジションは，トークンのデータ保持機能，Tag機能とトランジションに持たせた分岐条件により実現できる．

Fig.9に分岐トランジションの動作例を示す．まずトランジションの設定情報として，Tag情報，分岐条件，行先プレース番号がある．トランジションが発火するとトークンが取得しているTag情報/データを参照し，以下の順序で処理を行う．

1) Tagの有無を確認．（Tagが無い場合，分岐処理を終了し，設定されているプレースへトークンを渡す．）
2) トークン Tag とトランジションに設定したTagの認証．
（Tag 不一致の場合，分岐処理を終了し，設定されているプレースへトークンを渡す．）
3) トークンのデータを参照し，分岐条件に従いプレースへ移動．

このように，分岐トランジションを設置することにより，容易に条件分岐を表現することが可能となる．

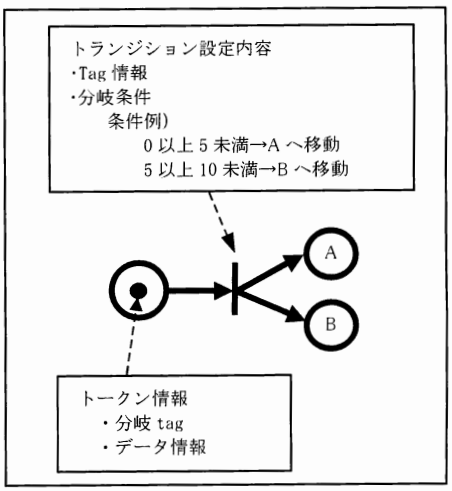

Fig.9: example of branch transition

4. 可達性問題解決

本ソフトウェアでは，前章で述べたペトリネットのフラクタル表現とペトリネットの拡張を用い，制御システムの視覚的表現を行うが，記述したネット図が開始プレースから終了プレースまで可達でなければ，システムを動作させることは出来ない．そこで，ペトリネットの可達性判定を行うための問題解決法を下記に示す．

本ソフトウェアでの可達性問題解決方法として，ネット図に描画されたトランジションとプレースの各写像に対してドットマトリックス作成し，可達木を作成し問題解決を行う．Fig.10に一般的なネット図の例を示し，Fig.11にFig.10のドットマトリックスを示す．Fig.11△はプレースからトランジションの写像を示しており，□はトランジションからプレースの写像を示している．このドットマトリックスを用い，Fig. 12のフローチャートをもとに可達木を作成し，その可達木より指定したプレース間の可達性を確認する．

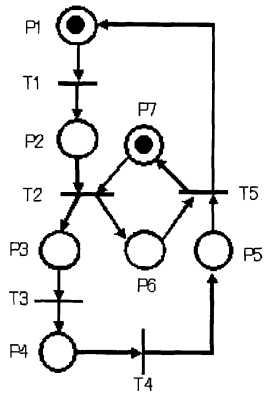

Fig.10: example of figure of Petri-net

	T1	T2	T3	T4	T5
P1	△				□
P2	□	△			
P3		□	△		
P4			□	△	
P5				□	△
P6		□			△
P7		△			□

Fig.11: Dot matrix

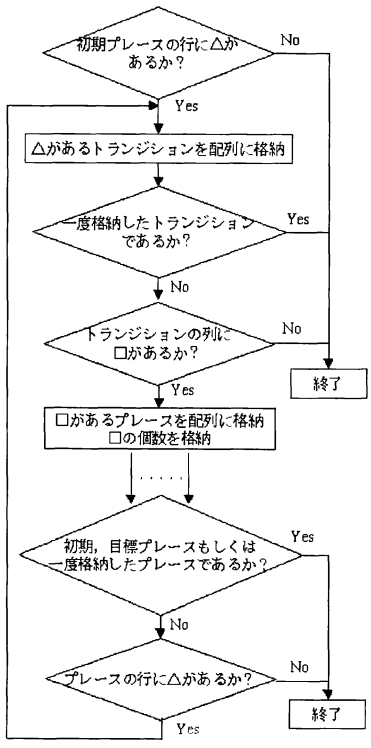

Fig.12: flow chart of Reachability Tree

5. おわりに

本研究では，制御システムの視覚的表現できるソフトウェアの構築を目的に，ペトリネット理論を用いた制御システム NPJ を拡張し FPNJ(Fractal Petri-Net for Java) を構築した．本システムでは、ペトリネットのフラクタル表現・視覚表現、可達性にて離散事象システムでの問題点（大規模化、複雑化、動作把握が悪いなど）を解消することが可能になると考えられる．FPNJ では，ペトリネットにフラクタル表現やライブラリ機能を導入し，制御システムの視覚的表現が可能になった．さらに，セーフティペトリネット導入とトランジションとトークンへの Tag 概念の導入により，ペトリネットで想定される問題（競合など）が緩和され，安全性を向上できた．また，可達性問題の解決により，システムの検証を可能にし，設計ミスを事前に把握できるシステムになると考えられる．

今後の課題として，エラー発生に対応する機能の追加，スーパバイザの導入および実機適用による機能検証があげられる．

文　献

1) システム制御情報学会編　青山幹雄・内平直志・平石邦彦，ペトリネットの理論と実践，朝倉書店
2) 離散事象システム研究専門委員会，ペトリネットとその応用，計測自動制御学会
3) 椎塚久雄，実例ペトリネット-その基礎からコンピュータツールまで- ，コロナ社 (1992)
4) 津藤洋明，渡部 慶二，村松 鋭一，有我 祐一，遠藤 茂：ペトリネットによるコンピュータネットワークシステムの構築，計測自動制御学会東北支部　第 214 回研究集会（2005.10.14）資料番号 224-12
5) 日本機械学会編，機械工学便覧，デザイン編 β 6 制御システム