計測自動制御学会東北支部　第197回研究集会(2001.10.13)
資料番号197-21

# ペトリネットによるコンピュータシステムの制御

## Petri-Net for Computer Network System

○野木　宏樹*　　渡部　慶二*　　羅　志偉**　　遠藤　茂*

○ Hiroki Nogi*, Keiji Watanabe*, Zhiwei Luo**, Shigeru Endo*

*山形大学大学院　理工学研究科, **理化学研究所

Yamagata University, Riken

キーワード：ペトリネット(Petri-Net)、ネットワーク(Network)、分散システム(Distributed System)

連絡先：山形県米沢市城南 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　野木宏樹, Tel.(0238)26-3178, E-mail:nogi@ewata.yz.yamagata-u.ac.jp

## 1. はじめに

現在、コンピュータを用いた制御システムは各界でさまざまな貢献をなしている。制御システムは一般的にはC言語などのプログラミング言語で記述されるが、このことはプログラミング初心者にとっては、簡単な制御内容の記述にも苦労を強いられることになる。また制御内容が複雑になればプログラムも長くなり、変更・修正も大変になる。

これらの問題に対して、以下の三点の解決が求められる。

- 初心者でも容易に制御内容の記述が可能である
- 一目で制御内容の理解が可能である
- 制御内容の保守・変更が容易である

これらを解決するために“ペトリネット”を取り上げる。ペトリネットには以下に挙げる特徴がある。

- グラフィカルモデルであり、図形の配置だけで制御システムを構築できる
- システムの状態遷移を一目で確認できる
- 離散事象システムの表現に的確

ペトリネットを用いると、マウスで画面をクリックし図形を配置するだけでロボットや機械を制御するシステムが容易に構築でき、制御の実行時には動作の状態遷移をリアルタイムで確認することができる。このため、エラーが発生した場合などは、どの作業段階で発生したのか容易に特定できる。以上のように、ペトリネット環境は制御システムの構築に有用である。

さらに本研究では、ペトリネット環境にネットワーク環境を加える。ネットワークによるロボットや機械の遠隔操作は日常生活にも実に便利である。本研究の目的は、インターネットを用いたTCP/IPデータ通信による複数台のコンピュータ上での制御システムを想定した“通信ペトリネットシステム”を構築することである。これによりパソコン間でのトークンのやり取りが実現され、メインコンピュータから複数台のサブコンピュータを制御することが可能となる。

また、この制御システムの構築にはJava言語を用いる。Javaはネットワーク処理を前提にして作成された言語であり、C言語などに比べ簡潔かつ明瞭にネットワーク処理を記述できるうえに、ブラウザなどに付属する“JavaVM”さえあればOSに依存しないという利点があるからである。

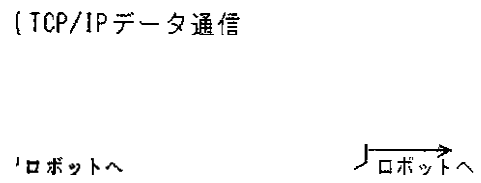


**図1．　複数のパソコン上の制御システム**

## 2. ペトリネット

### 2.1. ペトリネットの概念

ペトリネットの構造は、プレース(place)とトランジション(transition)と呼ばれる2種類の節点集合とその間のアークと呼ばれる有向枝集合で定義される2部グラフである。表1に示すように、プレースは丸○、トランジションは棒｜で、アークは矢印→で描かれる。トランジションとプレースを交互にアークで結合することによりネットは構成される。またトークンはネット上を移動し、現在の作業の進行状態を表している。実際のシステムを制御する場合には、トランジションに作業内容を割り当て、プレース→アーク→トランジション→アーク→プレースとトークンが移動していくことによって、現在の作業内容を視覚的に表現できる。

**表1．　ペトリネットの構成要素**

| 要素名 | 記号 | 役割 |
|---|---|---|
| プレース | ○ | 事象の状態 |
| トランジション | ｜ | 事象の動作 |
| アーク | → | 動作の方向 |
| トークン | ● | 現在の状態 |

### 2.2. 発火

トークンがトランジションにあるとき、トランジションは発火しているという。あるトランジションが発火するためには、そのトランジションのすべての入力プレースにトークンがなければならない。これを発火条件といい、これが満たされているとき、トランジションは発火可能であるという。


**図2．　トランジションの発火の様子**

### 2.3. 階層化

システムが複雑になるにしたがって、そのシステムの表現も複雑化する。ペトリネットにおいてもシステム状態の把握の容易さが失われ、ネット管理も複雑化してくる。この問題点の解決法の一つとしてネットの階層化が挙げられる。

ネットの階層化とは、図3に示すようにネットの一部分を一つの多重トランジションの中にまとめることによって、ネットを簡略化することである。

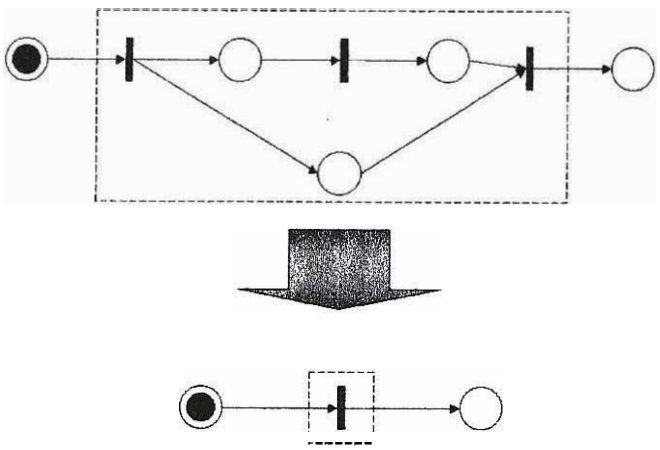

**図3．　多重トランジションを用いた階層化の例**

## 3. ペトリネット制御システムの構築

本研究では Java を用いて通信機能を備えたペトリネット制御システム Npj(Network Petri-net by Java)を構築した。Java はネットワーク処理を前提として作成された言語であり、ネットワーク記述が簡潔であるということ、またどんな OS 上でも同じプログラムが動作することができるという利点を持つ。

### 3.1. メインウィンドウ

Npj を起動すると以下に示すメインウィンドウが表示される。

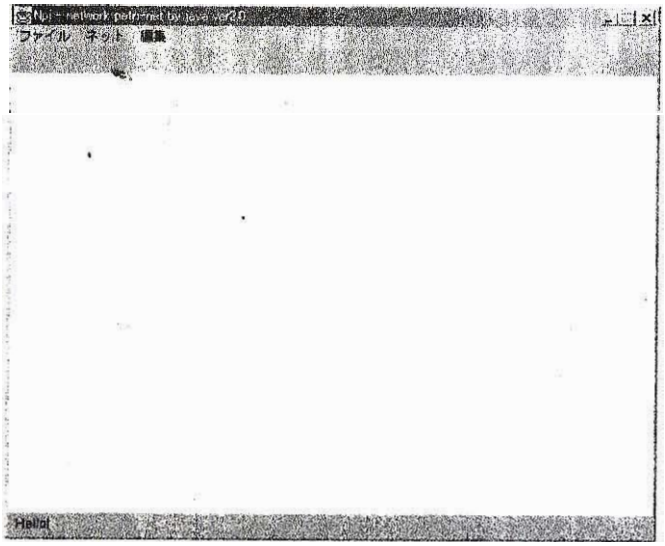


**図4． Npjのメインウィンドウ**

### 3.2. システム構成

Npj のシステム構成は、図5に示すような階層構造になっている。

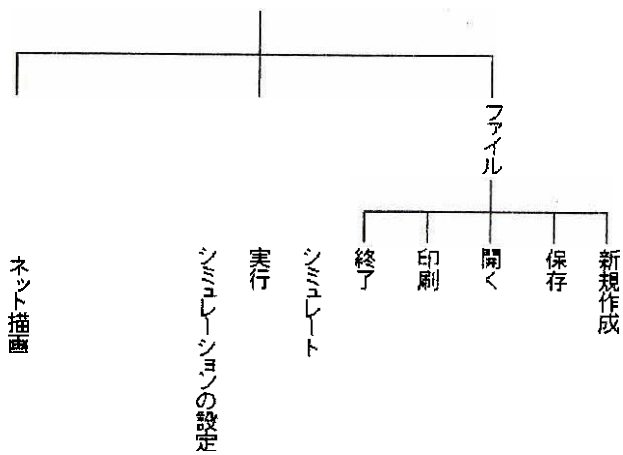


**図5． Npjの階層構造**

図5のおのおのの命令のもつ機能を以下に示す。

・**メイン**（プログラム起動時の最上位階層）

・**ファイル**

・**新規作成**
新しくネットを作成するために空白のパネルを生成する。

・**保存**
作成したネットにファイル名をつけて保存する。

・**開く**
保存されたネットファイルを開く。

・**印刷**
作成したネットをプリントアウトする。

・**終了**
システムを終了する。

・**ネット**

・**シミュレート**
作成したネット上でトークンを移動させ、システムのシミュレーションを行う。

・**実行**
Npj を用いて実際のシステムを制御する際のスタートボタン。

・**シミュレーションの設定**
トークンの移動時間など、シミュレーションに関する各種設定を行う。

・**編集**

・**ネット描画**
パネル上にネットの構成要素を描くためのネット作成ダイアログを表示させる。

### 3.3. ネットを描く

Npj でペトリネットを描くには、まず編集メニューのネット描画アイテムから、図6に示すようなネット描画ダイアログを呼び出す。このネット描画ダイアログでそれぞれの要素のボタンをクリックした後、続けて空白のパネルをクリックすることにより、要素を描くことができる。

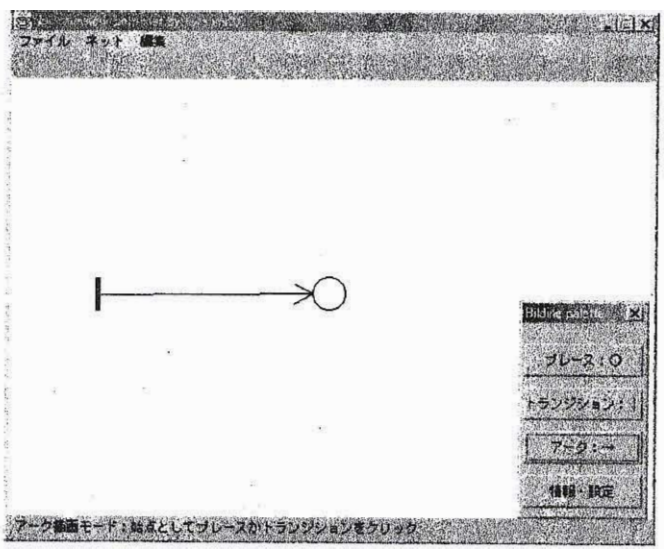


**図6． ネットを描いている様子**

### 3.4. 各種設定

ネット描画ダイアログの中に情報設定というボタンがある。これをクリックした後、続けてトランジションをクリックすると、図７に示すような情報設定ウィンドウが表示される。このウィンドウにより、システム動作のスタートとなる初期トランジションの設定、トークンを他のコンピュータのネットへと送信する際の IP アドレスの設定、またトークンを受信するトランジションの設定を行うことができる。

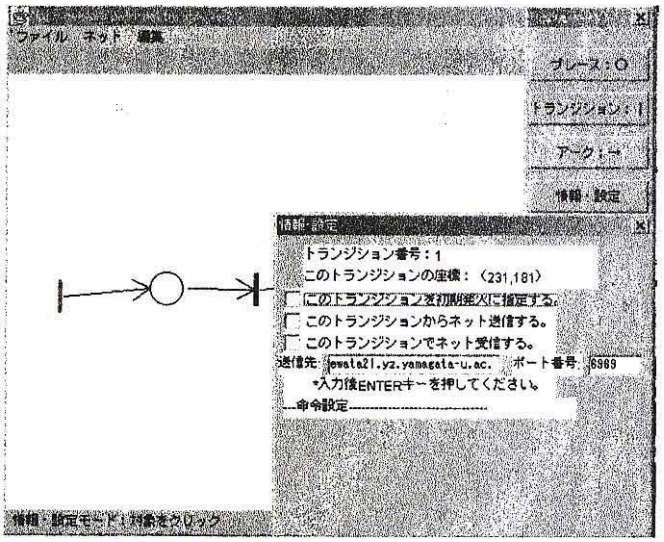


**図７． 情報設定ウィンドウ**

### 3.5. シミュレーション

ネット作成、および初期トランジションの設定が終了すれば、シミュレーションを行うことができる。ネットメニューからシミュレートアイテムをクリックすると、図８に示すようなシミュレーションダイアログが表示される。これを用いてシミュレーションの指示を行う。

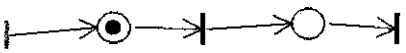

**図８． シミュレーションの様子**

## 4. Npj の応用

### 4.1. サブネットファイルによるネットの階層化

先にも述べたように、ペトリネットには階層化という概念がある。これにより複雑なシステムの表記も簡略化できる。Npj においては、図１０のように、トランジションの設定をする際に、トランジションの下にさらにサブネットファイルを設定することにより階層化を実現したいと考えている。

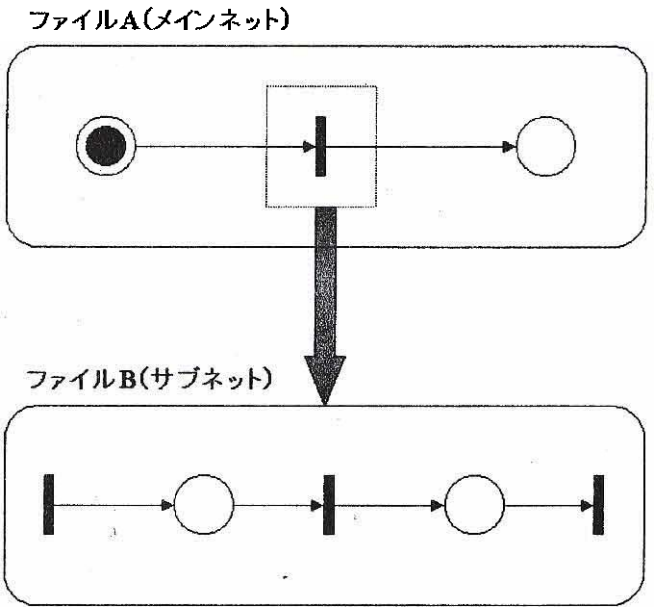


**図 9. ファイルによるネットの階層化**

### 4.2. インターネットによるトークンの送受信

Npj では、図９のモデルに表すように、インターネットを用いてトランジションからトランジションへとトークンを送受信することができる。これにより、たとえばメインコンピュータに描かれたメインネットから、複数のサブコンピュータのネットへとトークンを送信することができ、複数のコンピュータにおける作業も１台のコンピュータで制御することができる。

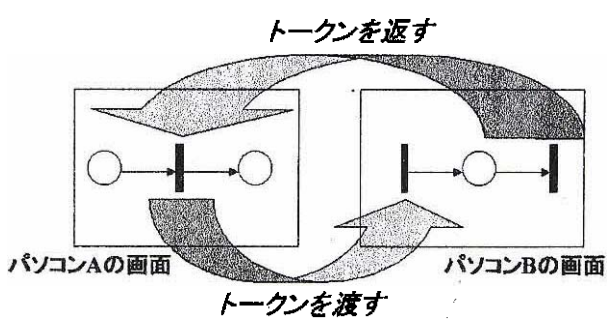


**図 10. トランジション間でのトークンの送受信**

## 5. おわりに

本研究において、コンピュータを用いたペトリネット制御システムの初期のアーキテクチャが完成した。将来的には、トランジションから動作命令を送ることにより、実際のシステムを制御することが実現できれば良いと考えている。また、エラーチェックなども実現できれば良いと考えている。

## 参考文献

1)椎塚久雄:「実例ペトリネット」 コロナ社 (1995)
2)熊谷貞俊・薦田憲久:
「ペトリネットのよる離散事象システム論」
コロナ社(1995)
3)StevenHolzner:
「Java プログラミング BlackBook」
インプレス (2000)
4) 柴田徹「Java を使ったペトリネット環境の構築」
卒業論文 (2000)