

生産システム解析のための並行と競合を考慮したペトリネットツール An Advanced Petri net for the Manufacturing System

有馬 澄佳, 齋藤 和之

Sumika Arima, Kazuyuki Saito

会津大学大学院 コンピュータシステム学専攻
The University of Aizu

キーワード：生産システム(Manufacturing system), ペトリネット(Petri net), 並行性(Concurrency),
競合(Conflict),

連絡先：〒965-8580 福島県会津若松市一箕町鶴賀 会津大学 コンピュータ素子学講座
有馬澄佳 Tel.: (0242)37-2568, FAX.: (0242)37-2597, E-mail: m5021202@u-aizu.ac.jp

1. はじめに

消費者のニーズが細分化された現代では、生産形態は、単一品種大量生産から多品種生産への移行しつつある。また、短期間で製品の移行が必要とされることも多い。このように生産現場に自由度の高さが要求されるため、生産者にとっての最大の鍵は、生産の効率化を検討することによって、出来るだけ無駄の少ない製造をすることである。そのためには、先ずその生産システムの構成・挙動を分析し、数値計算やシミュレーションを用いた解析が必要となる¹⁾。

システムの効率化を検討するためには、まずそのシステムの構成を明確に定義し、主要な構成要素とその属性をまとめ、各属性変数の変化による影響を数値計算やシミュレーションで解析する必要がある。数値計算を用いた解析では、各変数の確率的なばらつき変動などをまとめて平均値として扱うだけなので、より複雑なシステムの構成や挙動をより詳細に解析するためにはシミュレーションが必要となる。

そこで本研究では、生産システムの効率化解析法の検討・提案を行っている。システム記述・解析には、離散事象システムによく用いられる「ペトリネット」を使用し、生産システム用ペトリネットツールを開発中である。また、そのツールの妥当性を検討することで、より高度なシステム解析を目指す。

本報告では、既存のペトリネットツールを用いたシミュレーション結果を検討し、生産システム記述・解析のためにペトリネットツールに必要とされる並行性や競合などの機能を考慮した上で、新しく試作したペトリネットツールについて述べる。

第2章では、まず生産システムとペトリネットを構成要素と挙動の2点についてまとめ概要を述べる。それをもとにして生産システムを離散事象システムとしてペトリネットと対応付け、単工程のペトリネットモデル化を行う。次に、第3章では、既存ツールを用いた簡単なペトリネットモデルのシミュレーション結果とその問題点を記す。そして、第4章,第5章では、その問題点を解消するためのペトリネットツールの改良・開発経緯を述べ、第6章で簡単なモデルを用いたシミュレーションによるツール機能の確認を行う。最後に、第7章にまとめを記す。

2. 生産システムとペトリネット

生産システムのペトリネットシミュレーションを行うには、先ずシステムの構成をまとめ、ペトリネットモデルとして記述しなければならない。そこで本章では、一般的な生産システムとペトリネットとの対応付け、ならびに単工程のペトリネットモデル化を行う。

2.1 生産システムとは

生産システムとは、生産のための拠点、工場において製品を製造するシステムである。製造は、レシピ(Method)に従い、原材料(Material)や設備(Machine)、労働力(Man)を用いて行われる。これを一般に生産システム工程の4Mという¹⁾。

生産システムは、装置またはオペレータによる製品の加工処理や搬送を主要イベントとする離散事象システムといえる。

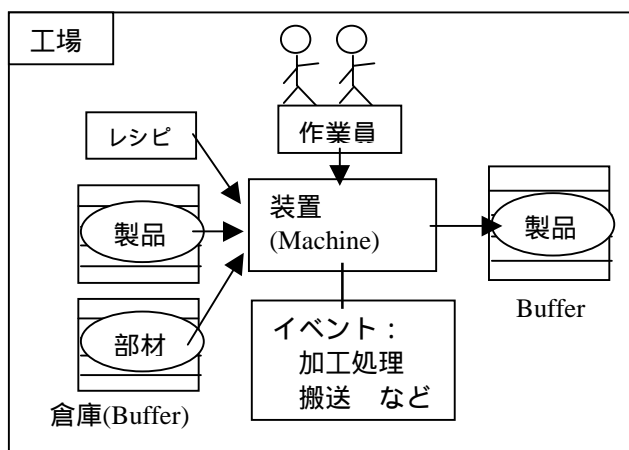


図1 生産システムの構成

製品は、需要(受注)という供給要求が起こってからその供給を実行するまでの時間に制約、納期を持つのが一般的である。この納期に従って、各製品の製造計画を立てられなければならない。製造は上記4Mを用いて行われるので、受注計画は4Mの能力に依存する。

以上により、生産システムの主要構成要素は表1としてまとめることができる。

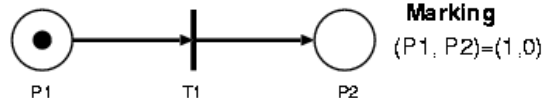
表1 生産システムの構成要素

構成要素4M	要素詳細
設備(Machine)	装置(Machine)、 倉庫(Buffer)
部材(Material)	製品(原料、WIP)、 リソース(部品、材料)
労働力(Man)	作業員(Operator)、 管理者
製法(Method)	製造レシピ(処理手順、納期)

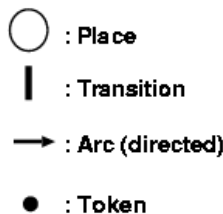
2.2 ペトリネットとは

ペトリネットは、離散事象システムを記述・解析するためのグラフィカルツールである。ペトリネットは、図2に示すように、システム構成記述のためのトランジション(Transition)、プレース(Place)、アーク(Arc)の静的な3要素、トークン(Token)と呼ばれる動的な要素の計4要素で構成される。この構成によって、システムの記述ならびに挙動の可視的シミュレートが可能となる。事象の発生はトランジションの発火(Firing)で表し、システムの状態はマーキング(Marking)と呼ばれるトークンの配置で表される。

< A Petri net Illustration >



< Elements of a Petri net >



< Making change by firing T1 >

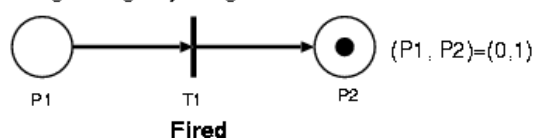


図2 ペトリネットの基本構成

ペトリネットは、その記述能力の高さから、並列的、非同期的、分散的、非決定的、確率的な動作を特徴とする情報処理システムなどの記述・研究に有効とされ、広い分野で用いられている²⁾。

さらに、ペトリネットには、論理的イベントの生起のみだけでなく時間付きペトリネットのようなイベント発生に遅延時間を付けた表現もあり、時間的な表現も可能となっている。

時間付きペトリネットには、様々なものがある。一般的に、決定的または指数分布

的な発火時間を持つものの大きく2種類に分けられる。ここでは簡単に決定的発火遅延時間を持つ時間ペトリネット(Deterministic timed net)を基本に考えていく。これは、各トランジションの発火時間が固定値で、時間の経過に対して図3のように状態が変わる。まず、入力揃ったときに、資源的発火可能(Enable)になり、その後スタンバイ時間 t_1 (時間ペトリネットでは通常0)が経過すると、発火決定(Fire decision)・開始となる。そして、最後に発火時間 t_2 が経過すると、発火実行完了(Fire execution)となり、出力側に処理を終えた資源が出力される。資源の入力と出力のタイミングは、現実の生産システム表現に即し別々に表現される方がよいので、トランジションタイマーを用いる。これは、トランジションの発火開始 t_1 時に入力プレースからトークンを取り除き、発火時間経過 t_2 後に出力プレースに再びトークンを出力させるもので、発火開始から終了までの間、使用される資源がトランジションの中にあるかのように表現される。

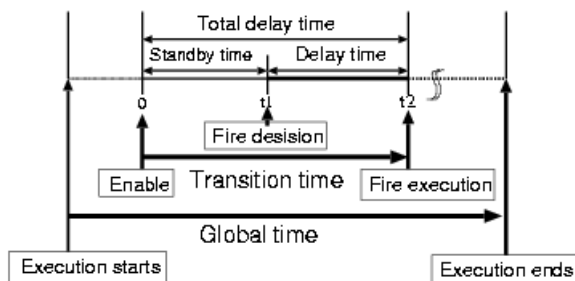


図3 時間ペトリネットの定義

2.3 生産システムとペトリネットの対応付け

生産システムのペトリネット記述のために、まず両者の構成要素の対応付けを行う。その基礎として、生産システム単工程の表現に必要なとされる要素とそれに対応するペトリネット要素を表2にまとめる。ここでは、生産システムを、装置処理(イベント)を中心とする離散事象と考えてまとめを行った。また、各処理にはある期間が必要となるため、時間つきペトリネット

を採用した。この対応関係から、単工程は図4のような4部構成で表すことができる。4部とは、中心となる1. プロセスメイン部、2. オペレータ要求制御部、3. リソース要求制御部、および4. 装置状態チェック部で、各部に含まれる要素や機能は表3に示す。

表2 生産システムとペトリネットの対応付け

生産システム		Petri net
要素	装置(処理時間)	Timed Transition
処理に 要求される リソース	仕掛かり品	Token
	仕掛かり品置場	Place
	部材	Token
	部材置場	Place
	作業員	Token
	作業員詰め所	Place
	装置とリソースとの関係	Arc
状態	故障(修復時間)	Timed Transition
	可動/非可動(メンテナンス)の別	Place 中の Token の有無

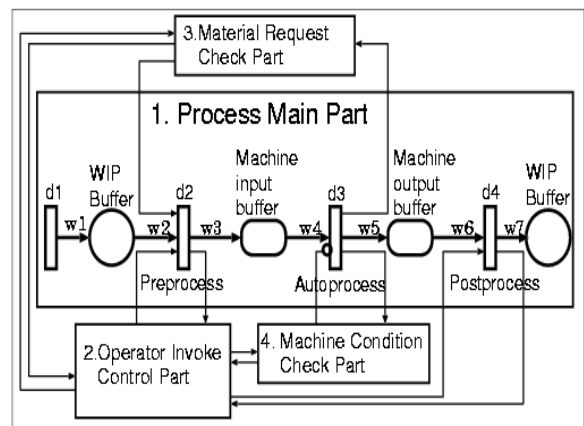


図4 生産システム単工程のペトリネットモデル

表3 各部の構成要素

部番号	構成要素
1	入出力バッファ(有限容量), 処理装置
2	作業員, 作業員詰め所,
3	部材, 部材倉庫,
4	装置可動/非可動チェック, 修理・メンテナンス・切り替え状態

生産システムの単工程は、この各部の資源の種類や量、あるいは能力によって特徴付けられる。

本報告では、一番中心となるプロセスメイン部に注目して、話を進める。プロセスメイン部は、単純には入力・出力バッファ、装置、および製品の3要素で表せる。

プロセスメイン部での処理イベントは主に、前処理(Preprocess)、自動処理(Autoprocess)、後処理(Postprocess)の3段階に分けられ、これらの各処理時間は、装置の種類(性能)やその状態、さらに処理する製品の種類によっても異なる。システムの構成によっては、時間的な近似によって前処理と自動処理だけのもの、自動処理だけのものも考えられるが、ここでは一般形として3段階を持つものとして話を進める。

単工程は、この3段階のイベントを持つ複数の並列ラインによって構成されている(図5)。製品は、この複数ラインのいずれかに割り付けられて処理される。

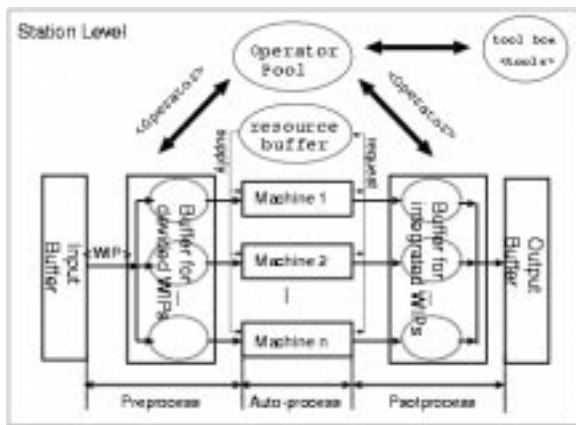


図5 単工程の構成

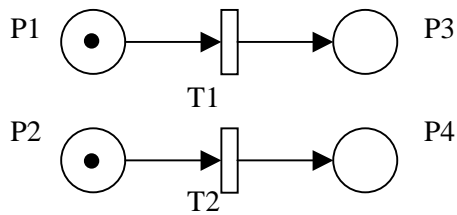
次の章では、プロセスメイン部の単純モデルを用いたシミュレーションについて述べる。

3. 既存ペトリネットツールを用いた生産シミュレーションにおける問題点

2つの既存のペトリネットツールA、Bを用いて、簡単な生産システム単工程のシミュレーションを試みた。

3.1 簡単な単工程のシミュレーションとその実験結果

まず、単工程を入力・出力バッファと装置のみとし、全3段階処理イベントを1段階に近似した単純構成ネット(図6)でシミュレーションを行った。このモデルは、他と独立な2つの並行処理ラインを想定したもので、各トランジション発火の並行性が要求される。トランジションT1とT2に等しい遅延時間を与えれば、これらは同時発火するべきである。しかし、両ツールともこの並行性の確認は困難であった。



Initial Marking: P1=P2: Enough Tokens, P3=P4=0

図6 並行性シミュレーションネット

また、製品処理の複数の装置へ割り付け(図5のPreprocess)を考えるために、図7のネットモデルを用いて競合表現の確認を試みた。しかし、期待通りの結果は得られなかった。

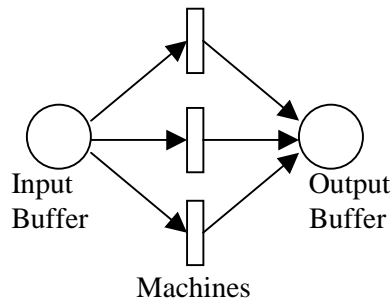


図7 競合シミュレーションネット

この競合問題は、バッファ容量が有限である生産システムにおいて、非常に重要な問題である。なぜならば、バッファを有限に保つためには式1(リトルの公式)が満たされなければならない⁴⁾、この条件下では競

合が起こるシステムが大半となるためである。

$$\lambda / N\mu < 1 \quad (1)$$

ここで、 λ は到着率、 μ はサービス率、 N は装置台数である。

以上2つのペトリネットシミュレーション結果から、生産システムシミュレーションのためには次に述べるような問題点があることが明らかになった。

i) 並行性(Concurrency)

同時発火の確認が不可能である。並行動作である確認をするための機能が必要であることが明らかになった。

ii) 競合(Conflict)表現

- ・ 動的競合の判定：時間ペトリネットでの競合は、構造的競合、資源的競合、時間的実行競合の3つの条件が判定され、最終的に動的に競合かどうか決定されなければならない。しかし、今回用いたツールではその判定が完全にはなされていなかった。
- ・ 動的競合時の解法：生産システムへの適合性を考えた際、実行競合になったトランジション群から発火トランジションを選択する条件設定法が不十分である。

これらの並行性と競合は、生産システム解析で重要な鍵となる表現であり、これらが正確に表されるペトリネットツールが望まれる。

次章では、この並行性と競合に重点をおいた生産システムのペトリネット記述とシミュレーションツールの改良を考えていく。

4. 生産システム記述・解析のためのペトリネットツール改良

3章で挙げた並行性と競合の2点の問題について、生産システムにおけるその重要性を述べる。

4.1 並行性

並行とは、事象が互いに他と独立に存在することのできるシステム構造または状態をいう。

生産システムは、複数の装置や部材、作業員などで構成される。そのシステムの挙動は、処理を行う・受ける等のイベントの組み合わせで表現され、イベントが複数並行して起こることがある。例えば、単純な一工程でも複数の装置や作業員が存在すれば、複数の製品が並行に処理されるなど並行性を持つシステムとなる。各々の装置や作業員が他と独立に処理を行えるのであれば、同時刻に複数の装置が稼働できるからである。

ある処理が他と独立に起こり、そこに並行性が存在することは、共通する一つの基準を付けなければ評価できない。そこで、時間的基準としてシステム標準時計の必要性を挙げる。意識的な並行状態(同期)はもちろんのこと、無意識にでも並行性を議論するには、共通の時間軸を用いているのが常である。それに対応するグローバルタイムをペトリネットに導入することで、これによって、ある時刻における並行な事象発生の確認ができるようになる。この実現には、図8に示すような実行の流れで、1.同時発火トランジションチェックと6.グローバルタイムを付加することで実現できる。

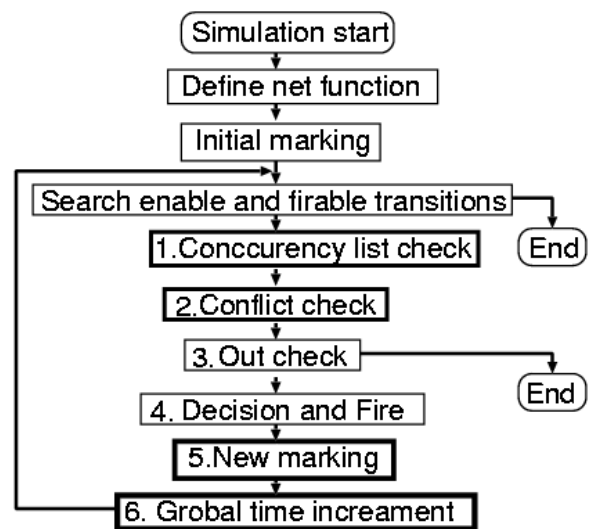


図8 ペトリネットツール改良版の実行フロー

4.2 競合

4.2.1 生産システムにおける競合

生産システムにおける競合問題とは、(1)どの装置や作業員がどの製品を処理するかの割り付け選択、または(2)作業員がどの製品の処理に従事するか、などの共有資源の取り合いである。装置や作業員を効率良く割り振ることにより、製品処理を効率化し生産量を上げる、または生産期間を短縮することが製造現場での重要な課題となっている。このことから、競合問題の解法は生産システム解析の重要項目である。通常、生産計画は、各製品の納期を守った上で一番効率的になるように考えられなくてはならない。製造現場では、必ずしも計画が予定通り進むとは限らず、時間軸に対して可変なので動的な解析が必要である。よって、製造の流れを工程毎またはある時間間隔毎に区切って、各段階(時刻)で最適な割付を行ってやるのが望ましい。

そこで、ここでは基本となる単工程中での割り付け問題について考える。それを各工程毎に行うことでシステム全体としての効率の解析につながる。製品と装置の割り付けおよび作業員の仕事の選択は、ペトリネットではトークンのトランジションへの割り付け、つまり競合と考えることができる。

次節では、生産システムの装置割り付け問題の解析に適したペトリネット競合表現の記述・解法を検討していく。

4.2.2 ペトリネットによる生産システム割り付け問題の取り扱いと競合ネット

生産システムでは、処理の流れを時間の流れに対応したシステム状態変化として確認することが必要である。そのため、生産システムの記述には、時間または確率ペトリネットが適当である。

しかし、時間ペトリネットには並行性や競合等の確認に関して多少の注意が必要である。

時間ペトリネットでの実行(トランジション発火)手順は、以下の2段階で表される。

入力プレースのトークン数チェック
資源的に発火可能かどうかのチェック。

トランジションタイマ - チェック
入力遅延時間を経過したかどうかのチェック。

このように、時間ペトリネットでは、資源的条件のみではなく、トランジションの遅延時間に依存してトランジション発火の生起が決まる。

よって、時間ペトリネットでは構造的に競合の形を持っていてトークンが不足した場合にも、競合の起こらないことがある。

それは、例えば、入力遅延時間(スタンバイ時間)が異なる複数トランジションが競合構造にある場合等である。実効競合はある時刻でのトークンの取り合いであるので、たとえ構造的に入力を共有していて、かつトークンが足りないような場合でも、遅延時間が異なるトランジション同士ではある時刻で動的に無競合となる場合がある。なぜならば、スタンバイ時間の小さいものの方が常に先に発火可能になるためである。

以上、時間ペトリネットでの実効競合が成立するのは以下の3つの条件が揃った時である(図9)。

構造的(資源的)に競合している。

資源が不足している。

時間的に競合が起こる。

時間ペトリネットでの競合問題には、ある時刻に同時発火(並行処理)可能なものと、動的に競合となるもの、動的に無競合なもの3種類の表現が必要である。

まず、構造的に競合にあるか、そうであれば、トークンは十分にあるか、あれば動的に無競合であり、同時発火可能である。なければ時間的な発火可能条件を確かめる。もし、時間的に同時発火時刻を迎えたものであれば、動的競合となる。もし時間条件が成り立たなければ、動的に無競合となる(図9)。

構造的競合において、資源が十分にある場合には並行同時発火となる。これもまた先に述べたグローバルタイムの付加によって、容易に確認することが可能となる。

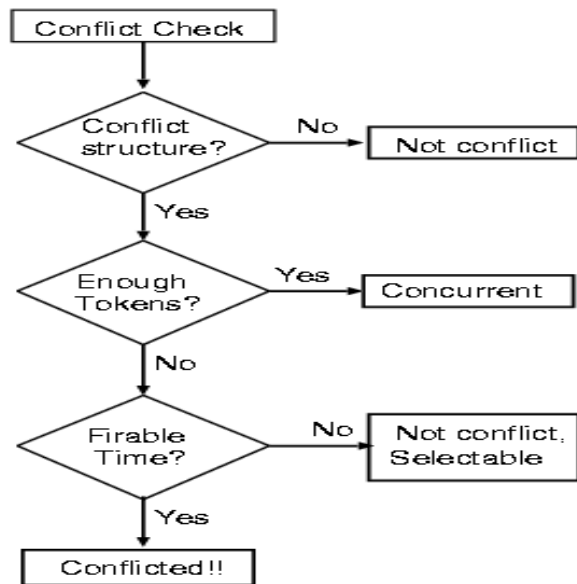


図9 競合の判定3条件

また、資源的に競合していても時間的に発火条件を満たしていないものは発火不能と認識される。

このように、競合が起こるのは図9の3条件を満たす時である。こうして、先ず動的競合の有無をチェックする。次に、動的に競合しているトランジションがあれば、その中から、実際に発火させるトランジションを選択する。この選択法は、システムの特徴によって異なるので、それぞれに合った設定ができる環境が望まれる。

次の節では、生産システムの競合として処理装置の選択方法を例に挙げ、この選択法のペトリネット表現を検討していく。

4.2.3 動的競合における発火トランジションの選択

前述の通り、生産システムの処理イベントは、前処理、自動処理、後処理の3つの段階に分けられ、これらの各処理時間は、各装置の種類(性能)やその状態によって異なる。ある製品を処理しようとする時、どの装置で処理を行うのかは、例えばその時

に可動な装置群の中で性能の高いものを選ぶなど、多くの場合意図的である。その意図的決断法の例としては、

- A) 単位量当たりの合計処理時間(サイクルタイム)が出来るだけ短い処理ラインから割り付けていく方法、
- B) 出来るだけ前処理/後処理の短いものから割り付けていく方法、
- C) 前処理と後処理の合計が短いものを選択する方法、
- D) 自動処理時間が出来るだけ短い処理ラインから割り付けていく方法
- E) 一様乱数的(適当)

など時間的選択が多くみられる。

そこで、この装置選択問題をペトリネットの競合表現として付加することを考える。基本的には、装置をトランジションに見立てて競合構造を記述し、各トランジションに処理時間に対応する遅延時間を持たせてやればよい。ただし、それら比較選択すべきトランジション群では、意図的に競合を起こす必要があるので、簡単には構造的競合を持つトランジションのスタンバイ時間を等しく設定する。スタンバイ時間を異なる値にした場合には、上で述べたように実際に競合が起こらない場合があるので、注意が必要である。これが、競合表現の記述面での課題である。そのため、図10のような生産システムをペトリネットで記述をするならば、図11のように記述する。図11で大切なことは、競合構造となる全てのトランジションのスタンバイ時間を等しいものとすることである。これは、例えば前処理と自動処理を1つのトランジションで表し、前処理をスタンバイ時間として設定すると、上記のB)の割り付け方法になってしまうため、希望する様々な割り付け法の適切な表現ではなくなるためである。

例えば、図11のようなn個のトランジションがあり、それらのスタンバイ時間が $d_1 < d_2 < \dots < d_n$ だとする。この時、全てのトランジションが入力トークン待ちで、Input Buffer内のトークン数がm個 ($m < n$) の場合、nトランジシ

ンは資源的競合状態になるはずである。しかし、スタンバイ時間の小さいほうが時間的発火条件を満たす時刻が早いため、Machine 1 から順に発火可能時間を迎える。時間が経つごとに、入力プレースのトークンが少なくなり、資源がなくなるまで順に発火する。そして、あるトランジションを発火させると、次のトランジションが発火できなくなるところで初めて競合が起こる（ただし、他からの投入がないとする）。ここでの問題は、トークンが入力プレースに滞在する時間がトランジションの発火条件（スタンバイ時間）になっているため、これを考慮しないと生産システム解析のための競合表現が誤ったものになってしまう可能性があることである。そのため、時間ペトリネットにおける競合で生産システムで行われている意図的な割付を表現するには、従来のような各トランジションの時間的要素で決断する方法では表現できない。

そのため、従来のペトリネットから、ネットの変換もしくは実行仕様の変換をする必要がある。

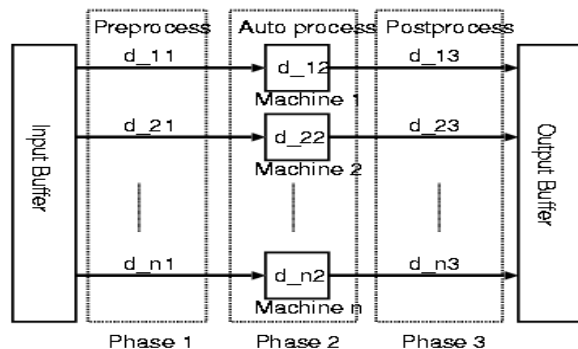


図 10 複数並列ライン構成

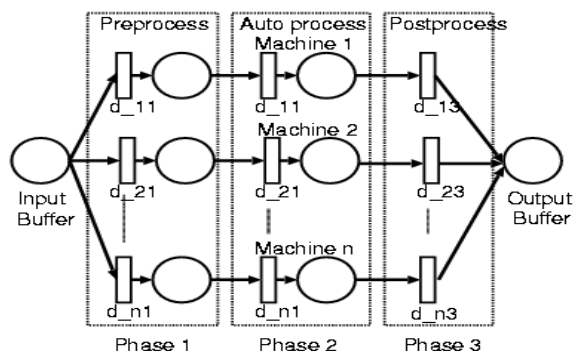


図 11 複数並列ライン処理のペトリネットモデル

4.2.2 ネット実行選択法について

生産システムをペトリネットで解析するためには、前述のような生産システム記述により効果的な競合時の解法を付け加え、ペトリネットの生産システム解析への適合性を高めることが望まれる。そのためには、図11のようなコンフュージョン（並行と競合の混在したネット形式）構造中の各種パラメータの中から主要な条件を選び、その条件をもとに各ラインの優先順位を付け、競合の解法とする必要がある。この競合の解法には、「競合選択条件の決定」とそれを元にした「トランジションの優先順位付け」、そして「その優先順位付けによる発火優先順の決定・実行」を意図に合わせて設定できるような柔軟性が必要である。この具体例を、第5章に記す。

5. ペトリネットツールの試作

具体的に以下のようにプログラムの改良を行った。ここでは、元になるソースとして、ソースコードが公開されているSANET²⁾を参考にした。

5.1 並行性確認の実現

グローバルタイマーの導入
前述の通り、同時発火としての並行性は、それを確認するための一つの共通な基準を定めることで確認可能になる。

5.2 競合の解法

5.2.1 生産システムにおける処理割り付け問題と従来の競合定義

生産システムでの装置やオペレータなど処理に必要な資源の割り付け問題には、一般的に設備能力による選択が行われることが多い。具体的には、製品の処理を行う装置を決定する場合に、より優先度の高い製品をより高性能の装置で処理するなどである。この生産システムにおける割り付けとペトリネットの競合解法を。

同一処理を行う単工程で、異なった性能を持つY台の装置あるいは処理ラインがある場合を考える。これに対して、ある時刻に入力バッファ中にX個のWIPが存在し、これを各ラインに割り付けようと思う。ある

時刻に、Y個の装置（ライン）の中でN個の装置（ライン）が使用可能だとする．それに対して、入力バッファにあるX個のWIPの中で優先順位の高い順にWIPを処理ラインに割り付ける．これを、今考える割り付け問題、すなわちペトリネットで考える競合問題とする．

通常の競合の解法では、 $X < N$ であるならば、競合が起こることが予測できる．しかし、その競合処理では、1 WIPずつ割り付けていき、最後の1 WIPになった時に始めて残りの使用可能な複数 $N - X + 1$ ラインで競合が起こるといった表現になっており、それ以前の発火トランジション選択順序は任意とされている．生産システムの割り付け問題を考える際には、この競合解法では表現が不足である．そこで、この解法の改良案を次節に示す．

5.2.2 競合問題の新解法

具体的に以下に示す2段階を用いた競合解法を提案する．

トークンの優先順位付け：

トークンは、生産システムの仕掛品に対応する．仕掛品の優先順位の決定条件には、納期、品種、数量などが挙げられる．

割り付けの際には、まずこれらを総合して仕掛品のレベル付けがなされ、優先順位が決定される．そこで、まずプレス内のトークンの優先順位をつける必要がある．しかし、通常はFIFOを基本として、そこに特急品などの割り込みがかかる程度の優先順位決定になっていることが多い．そこで今回は、FIFOと仮定してトークンに区別はつけないことにした．

トランジションの選択：

トランジションは、生産システムの装置に対応する．前述の通り、製品の装置への割り付けは各装置ラインの性能（サイクルタイムなど）によって行われる．

- a) 各装置（ライン）の性能指数を計算する．
- b) 性能指数の高いものを で決定した優先順位の高いWIPに割り付ける．

この割り付けにはまた、使用可能な装置ラインのみを割り付け対象にするものと、全ての装置ラインを対象にして割り付けるもの（使用ラインの予約を含む）の、異なる2種類が考えられる．予約付きについては、本紙では触れない．

今回は、各装置への割り付け条件を単純にその処理時間によるものとして考えた．よって、前述a)の各装置（ライン）の性能指数は各装置の処理時間で与えられる．これをペトリネットの競合条件に対応付けると各トランジションの発火遅延時間と考えられる．そこで、本報告では動的な競合が起こった時の発火優先順位付けを各トランジションの発火遅延時間が短いほど優先順位が高くなるよう設定した．

6. ペトリネットツールの確認

上記の並行性と競合の新機能を持つツール機能の確認として、簡単なシミュレーションを行った．

6.1 シミュレーションネットと設定

図13のネットを用いて並行性と競合表現が改善されたことを確認する．このネットは、競合構造を持っているが、入力プレースに十分なトークンを置けばトランジションは並行動作をするので、並行性（同時発火性）の確認にも用いることができる．

6.2 シミュレーション結果

6.2.1 並列性の確認

図13において、各トランジションタイマー（スタンバイ時間 / 遅延時間）を Tr_1, Tr_2 では0/1、 Tr_3, Tr_4 は0/2に、また Tr_5 は0/4と設定してシミュレーションを行った結果を図14に示す．まず、同一遅延時間を持つ Tr_1 と Tr_2 または Tr_3 と Tr_4 が同じ値で推移していることから同時発火が確認できた．また、遅延時間の比に対応した発火回数の比が得られることも確認できた．例えば、 Tr_1, Tr_3, Tr_5 において遅延時間の比は $Tr_1:Tr_3:Tr_5=1:2:4$ であるが、それらの発火回数の比は $Tr_1:Tr_3:Tr_5=4:2:1$ となっている．

以上により、グローバルタイムの導入で並列性が確認できるようになったことを確認した．

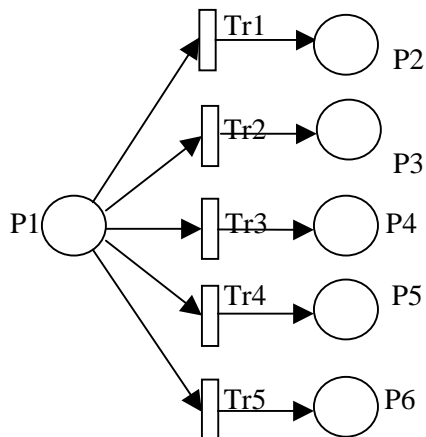


図13 並行・競合シミュレーションモデル

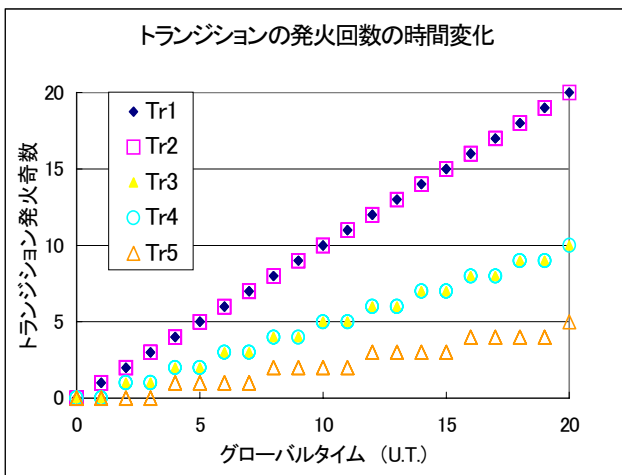


図14 並列性のシミュレーション結果

6.2.2 競合

まず、P1のトークンの数を十分にしたときに動的に無競合となり、遅延時間を等しく設定をした場合には上記の並行性と同様に、同時刻に発火することが確認できた。また、遅延時間による優先順位付けの設定をし、トランジションの遅延時間を $Tr1=0/1$, $Tr2=0/2$, $Tr3=0/3$, $Tr4=0/4$, $Tr5=0/5$ と設定した時に、各トランジションの発火優先順位が $Tr1=1$, $Tr2=2$, $Tr3=3$, $Tr4=4$, $Tr5=5$ と設定ができたことを確認した。また、競合の解法を優先順位順にし、入力トークン数をトランジション数以下にして、競合を起こしたときには、その順位通りに発火させることもできるようになった(表4)。

表4 競合時の発火トランジション(・発火)

トークン数 (P1内)	Tr1	Tr2	Tr3	Tr4	Tr5
1					
2					
3					
4					
5					

7. おわりに

本研究では、生産システムのペトリネット解析を目的として、先ず生産システムの構成を簡単にまとめた。それをもとに、生産システム単工程をペトリネットモデル化し、既存のペトリネットツールによるシミュレーションを行い、ツール改良点を検討した。特に、生産システム解析のためにペトリネットツールに必要とされる機能の中で、最も重要と思われる2点、並行性と競合について考察した。その考察から、不足な機能を実際にプログラムに付加し、同時発火による同時並行性の確認と、競合時の並行動作ならびに競合時の発火順序について、期待通りのシミュレーション結果を得ることができた。

今回は、ごく単純なシステムを想定したが、実際には非常に複雑なシステムを解析しなければならない。今後、プログラムの改造を続け、より適応性のあるシミュレーションを目指すと共に、結果の解析法についても研究を行っていく。

参考文献

- 1) 藤本英雄：コンピュータ統合生産システム，76/116，コロナ社(1993)
- 2) T.Murata：Proc. IEEE, vol.77,541/580 (1987)
- 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) Raj Jain：The Art of Computer Systems Performance Analysis, 507/544, John Wiley & Sons, Inc.(1991)
- 5) 長谷川健介：ペトリネットの生産システムへの応用，計測制御学会論文集，775/783 (1989)