

平成 22 年 5 月 21 日現在

研究種目：若手研究 (B)

研究期間：2007～2009

課題番号：19700066

研究課題名 (和文) 拡張ペトリネットを用いた並列システムの実行環境の研究

研究課題名 (英文) The execution environment for the Parallel System by using extended Petri Net

研究代表者

山口 真之介 (Yamaguchi Shin' nosuke)

九州工業大学 大学院情報工学研究院・助教

研究者番号：00380733

研究成果の概要 (和文)：拡張ペトリネットによる並列システムの実行環境として、XML をベースとした、Java によるエンジンプログラムを開発。この実行環境を拡張する為に、三次元グラフィックの描画、アニメーション制御、分散アルゴリズムの設計に適用。エンジンプログラム、及び XML データ構造の拡張を行った。

研究成果の概要 (英文)：We develop the engine program by Java based on XML for the execution environment of the parallel system by the extended Petri net. We extend this engine program and the data structure of XML. We applied the execution environment to drawing, the animation control of 3D graphic, and the designs of the distributed algorithm.

交付決定額

(金額単位：円)

	直接経費	間接経費	合計
2007 年度	800,000	0	800,000
2008 年度	500,000	150,000	650,000
2009 年度	600,000	180,000	780,000
年度			
年度			
総計	1,900,000	330,000	2,230,000

研究分野：総合領域

科研費の分科・細目：情報学・計算機システム・ネットワーク

キーワード：分散システム・ペトリネット

## 1. 研究開始当初の背景

並列処理システムは仕事を並列に処理するため、資源を有効に利用する点において逐次処理システムよりも優れている。

現在、並列システムは、多数の要求を処理するネットワークサーバやデータベース、また、多量な演算を実行しなければならない暗号化処理や、シミュレーション、検証等に導入され有効に活用されており、現在の情報技術において必要不可欠なものとなっている。

しかし、実際に有効な並列システムを新たに開発、構築する為にはシステム開発者は、次の点を考慮しなければならない。

- (1) 同じ処理を並列に行う場合、分散された資源は正しく収集しなければならない
- (2) 共有する必要がある資源は占有された後、必ず解放されなければならない
- (3) 並列処理の部分と逐次処理の部分は正しく連携しなければならない (処理の全てが並列化できるとは限らない)

上記の点は逐次システムには存在しない間

題である。並列システムが大規模になればなるほど、これらが影響し合い、開発者が見逃す可能性が高くなる。

この様なミスが設計過程で見落とされ、実装過程、或いはその後において発覚した場合、まずその原因を辿る事が困難であり、改善に大きく時間がかかる。さらに、判明した原因の大きさによっては設計段階にまで遡って修正しなければならない状況も起こりうる。これはシステム開発者にとって非常に大きな負担であり、障害が見つかった際の損害も大きく、可能な限り避ける必要がある。

この人為的ミスを設計時点で発見するために、開発者はシステムを開発する際にペトリネット等、数学的モデルを利用した設計を行い、これ基に正当性を検証することが多い。

ペトリネットは、離散的システムをモデル化するために開発されたもので、システムを構成する要素の状態をプレース、トランジション、アーク、マークという記号を使って図的に表現する。ネットモデルの動作は、まずトランジションの発火と呼ばれる動作が行われ、この発火によってプレース間のマークの移動が行われる。トランジションの動作はそれぞれが独立しており、この一連の動作がシステムの状態の変化を表している。これをモデルで表現することで、設計者はどのような順序でシステムが動作し、次の処理には何が必要であるかを容易に理解できる。また、システムを形式的に表しているの、数学的に表記、検証ができる。

## 2. 研究の目的

本研究はペトリネット技術を利用した並列ソフトウェアの実行環境を提供するものである。一般に開発と実装が困難である様々な並列処理システムを、数学的モデルによって設計し、そのモデルをデータとして実装、動作検証に利用する。ペトリネットをベースとした研究は、システム検証やペトリネットの拡張、特定のモデルの数学的検証など、幅広く行われている。近年までに提案されている数多くの拡張ペトリネットは、従来のペトリネットに比べ記述能力が高く、複雑なシステムでも図式が可能であり、様々なシステム検証に用いられている。

しかし、システムの検証から、問題点や安定性が確認されたとしても、ペトリネットモデルから実システムの構築や動作の検証については、結局そのモデル図を基に、プログラム言語等を用いて人手で作成するのが主であった。研究者が独自に設計、検証したネットモデルをプログラム言語に変換、回路への実装する手法についての研究も行われていたが、近年まで拡大するまでには至らなかった。これは拡張された様々なペトリネット

を記述するための定義が無かった為、それぞれの研究者が作成したモデルをデータとして共有することができなかった事が考えられる。

しかし、XML による PNML (Petri Net Markup Language) の提案から、ペトリネットを計算機上で構造化されたデータとし、これを他の形式記述言語やプログラミング言語へ自動変換する研究が、近年多くの機関で行なわれてきている。

本研究では、このペトリネットモデルのデータを元に動作する駆動プログラムを作成して、設計者の要求する動作を再現する。これによりモデルが設計されれば、動作はそのまま確認できるので、プログラム作成などにかかるコストが大きく削減される。駆動プログラムの開発により、複雑なシステムの設計から実装までにかかる手間を削減するシステム開発手法の提供を目的とする。

システムのアウトラインを図 2.1 に示す。

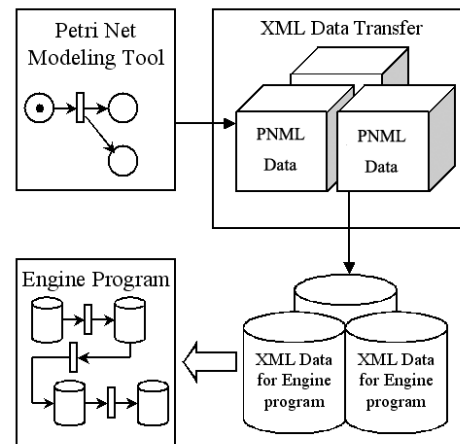


図 2.1 システムのアウトライン

開発者は自身が開発するシステムのモデル仕様に基づいて設計する。この設計にはフリーの Petri Net モデリングツールを利用する。この時点で設計者はモデルが正常かどうか、マークの動きによる検証は可能である。

作成されたモデルは PNML に変換され、電子データとなる。これをエンジンプログラムに渡して、実際に開発者の望む処理を実行させる。ここで、開発者は自身が設計した仕様に適しているか、また問題点が存在しないか検証する。問題が生じた場合は、再びモデルの設計に戻り、モデルの改善を行なう。

## 3. 研究の方法

### (1) ネットモデルの拡張に対応した駆動プログラムの拡張

駆動プログラムが扱うペトリネットモデル

は、信州大学の不破，和崎らが提案した「色つき論理ペトリネット」をベースとしている。今後はファジーペトリネットや，時間ペトリネットなど，検証するシステムの特徴に応じた，様々な格調ペトリネットが存在しており，本研究の駆動プログラムをこれに適応させることで，設計可能となるシステムをより拡大させる。

## (2) 並列システムの設計と検証

3DES 暗号，AES 暗号アルゴリズムについて，モデル化と動作検証を行なっている。さらに楯田暗号アルゴリズムのモデル化を行い，システムで動作検証を実施する。

## (3) 2次元，3次元グラフィック制御への適応

グラフィック制御への適応を実施する。まずネットモデルの形を骨組みとして，マークに画像情報を与え，2次元，3次元の画像表示の実現を目指す。

次にマークに動作情報を与え，ネットモデルが発火する事で画像を制御する，アニメーションの制御について検証を行なう。

本学の高度マルチメディア教育システムで導入された，3次元入出力装置を用いて，ネットモデルの3次元表示と，3次元グラフィックのアニメーション制御について，研究を行なう。これらを基に，作成される並列システムの視覚的な表現と画像情報の削減を目指す。

## (4) 自立分散ネットモデルへの拡張

ペトリネットの拡張の一つとして，マルチエージェントによる自立分散モデルの作成を検討している。これはネットモデル内の状態を示すマークが，さらに別のネットモデルを内包しているペトリネットである。マーク内のネットモデルが動作することにより，マーク自身の情報が外側のペトリネットが発火することなく変化する。マーク内のネットモデルと，外側のネットモデルが情報を交換する事で，マーク自身が状態を検知，それに応じた内部ネットの発火処理を行い，マーク自身が外部ネットモデル内で自立した動作を可能とする。これは，並列に動作するそれぞれのシステム（マーク）が，状況に応じた動作を選択できることとなる。

この拡張は，複数のシステムがそれぞれ通信を行い，状況に応じた動作をする経路探索の様な，思考が伴うシステムの設計に適応できる。

## 4. 研究成果

本研究ではエンジンプログラムの拡張を

行い，次の2つについて，実証実験を行った。

### (1) 3次元グラフィックモデルの描写とアニメーション制御

拡張ペトリネットで設計したモデルが，3次元のグラフィックを表現し，これをエンジンプログラムによって描画，動作させる。

3次元グラフィックを動作させる為に，拡張したネットモデルの構造を定義した。図4.1にネットモデルのXML構造を示す。

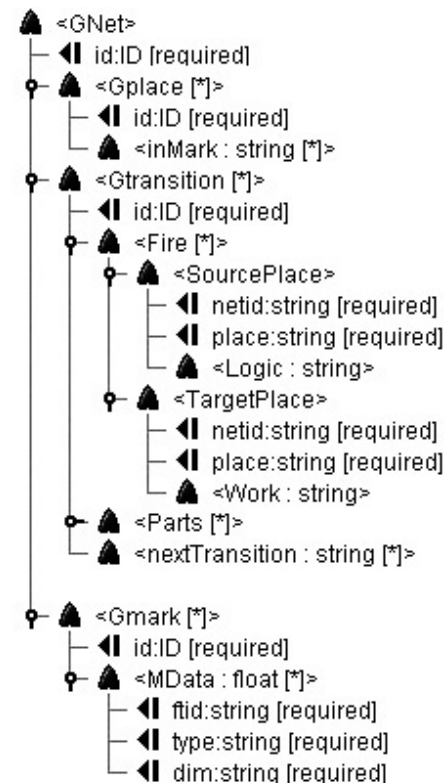


図 4.1 ネットモデルのXML構造

このXMLデータはPetri ネットエンジンが評価に利用する。Gplace 要素，Gtransition 要素，Gmark 要素はそれぞれが，プレース，トランジション，マークを示している。

またグラフィック描画用のXMLのデータ構造を定義，Parts 要素にはグラフィックの描画用データが記録されている。Parts 要素の構造を図4.2に示す。トランジションの発火時には，ここのTarget 要素のデータを更新した後，描画用データを更新する。Gtransition 要素の持つParts 要素は，最初に描画用データを作成する際に利用される。Gtransition 要素はParts 要素を複数保持することが可能だが，このうち動きを制御できるのは1つ(可動部)だけである。残りのParts 要素は，可動部に接続されたパーツとなる。それらのパーツは，可動部の動きに合わせて

動作するが、パーツが独自に動作することはない。その様なパーツを増やす場合は、トランジションを一つ増やす必要がある。

Gmark 要素はマークの持つデータを示している。今回のエンジンの仕様では、トランジションに対応した部品が、どれだけ動作するか、についての情報が記述されている。

一つのパーツに複数の動作を行わせる場合は、同じ属性 Ftid を持った MData を複数持つ必要がある。

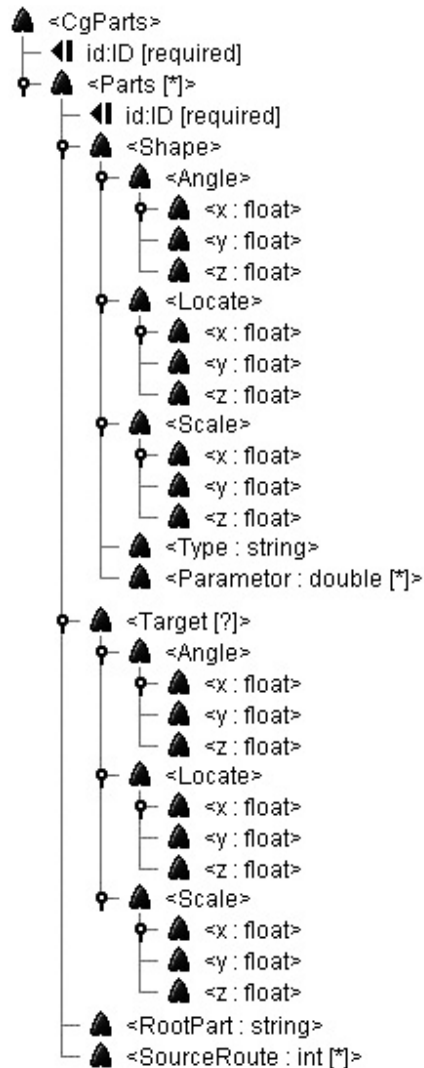


図 4.2 グラフィック描画の為に XML 構造

エンジンプログラムは、この XML データを取り込み、発火評価を行う事でグラフィックを描画する。ペトリネットの発火に対して、描画されたパーツの動作を事象とする。発火毎に更新されるデータを元に、繰り返し描画することで、アニメーションを表現する。

グラフィックの描画に対して、ペトリネットの評価は高速である。ペトリネットエンジンは発火評価を終了した後、描画用データに更新したデータ（位置、大きさ、角度）を記述する。描画用エンジンは、定期的に描画用

データを読み込み、その都度描画するデータを更新して、アニメーションを表現する。

エンジンプログラムは Java で作成した。開発には eclipse を用いている。Petri ネットエンジンと描画用エンジンは、独立して動作する必要がある。よってそれぞれのエンジンプログラムは、マルチスレッドによって動作するように作成している。以下にエンジンプログラムの動作順序を示す。

- ① Petri ネットエンジンがネットモデルを読み出す
- ② ネットモデルから Target 要素と、描画用の XML データを作成する
- ③ 描画用エンジンが描画用データを読み、初期状態のグラフィックを描画する
- ④ Petri ネットエンジンがネットモデルを評価し、条件を満たしたトランジションを発火させる
- ⑤ 発火評価が終了した後、Petri ネットエンジンが描画用データを更新する
- ⑥ 描画用エンジンが描画用データと現在の状態を比較し、描画する位置、角度等を更新する
- ⑦ 描画用エンジンが、更新したデータを元にパーツを描画する
- ⑧ ④にもどり、これを繰り返す

実験例を次に示す、グラフィックに途中で分岐する二股の多関節アームのモデルを作成、エンジンプログラムの動作を確認する。

エンジンプログラムの実行結果は図 4.3 に示す。この実験では、スレッド間が正常に動作している事を確認する為、エンジンプログラム側でマークをランダムに生成し、繰り返し発火評価を行っている。

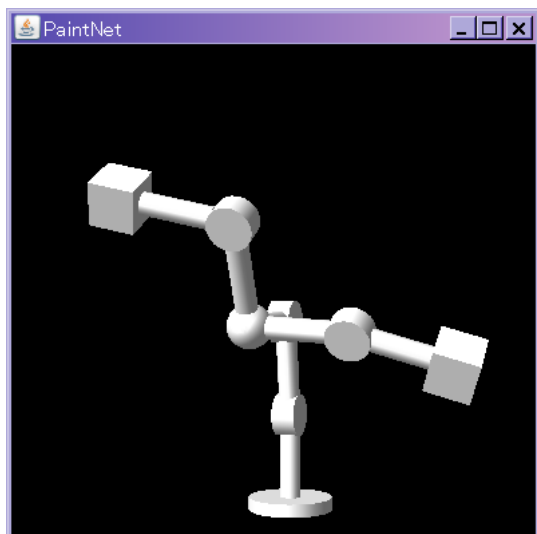


図 4.3 エンジンプログラムによる描画

エンジンプログラムがマークを繰り返し生成する為、アームは繰り返し動き続ける。土台の回転に従って、全体が回転し、下部の関節の動作に従って上部の動作する。全体としてアニメーションに矛盾はなく、正常に動作している事が確認できる。

このモデルの実験では、分岐したパーツの動作は独立に動作すること、かつ土台の動作には影響を受けている事を確認できる。

以上、3次元のグラフィックの制御にPetri ネットモデルを用いてアニメーションを描画する方法を提案。グラフィックを描画するエンジンプログラムを作成して、多関節アームのアニメーションを描画した。これについて発表論文1で報告を行った。

## (2) 分散アルゴリズムのモデル設計とシミュレーション

2つ目の情報収集の為の探索アルゴリズムの設計とシミュレーションについて述べる。このアルゴリズムは、周辺の情報を自分の記憶領域が限界になるまで保存し続ける。この時、一度探索した場所は無視して他の場所へ移動する。他のエージェントとの距離が近づいた場合には、互いの探索情報を交換して、探索済みの場所を記録していく。

以下にエージェントの探索アルゴリズムの処理を述べる。

- ① エンジンプログラムから呼び出される
- ② ランダムに探索空間に配置される
- ③ 自身のいる場所で情報を保存する
- ④ 周辺の各エージェントの位置を把握する
- ⑤ 近いエージェントが存在した場合、今まで探査した情報を交換する
- ⑥ 自身の進む方向をランダムで決める
- ⑦ 進行可能な場所であれば、一歩進む
- ⑧ その場所が探査済みでなければ、情報を保存する
- ⑨ ④～⑧を記憶場所一杯になるか、移動可能回数が限界になるまで繰り返す

このアルゴリズムをネットモデルで設計したものが図4.4になる。

実験では特定の画像データの各ドットの情報をエージェントが収集する。それぞれのエージェントが情報を交換しながら、できるだけ情報の重なりを避けて探索を行っているか確認する。探索状況の情報として、全ての位置を確認できるだけのBoolean型の二次配列をエージェントに持たせている。エージェントは近付いた他のエージェントに対して、この探索状況の情報だけを交換する。

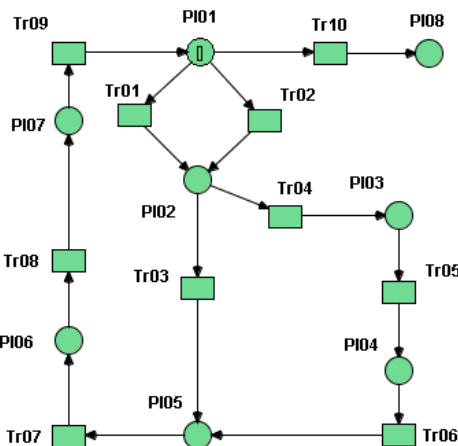


図4.4 エージェントのネットモデル

実験の条件と環境を以下に示す。

- 探索する範囲：250000, (500×500 ドットの画像)
- エージェントの数：40 個
- エージェントの記憶容量: 位置情報と色情報, 10000 ドット分
- エージェントの移動可能回数：60000 回
- 探索状況の情報：Boolean 二次配列 (500\*500)

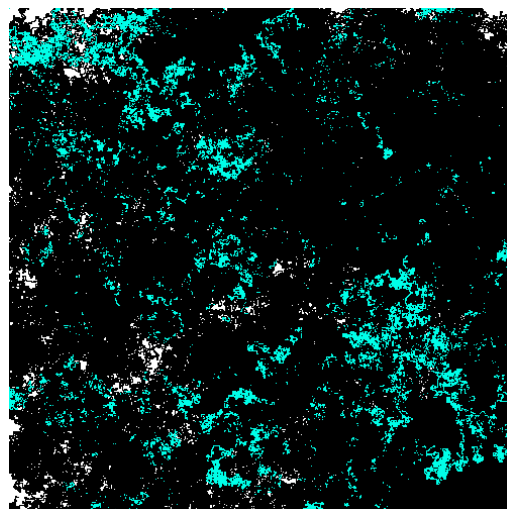


図4.5 エージェントによる情報探索

実験結果を図4.5に示す。この図はエージェントがもつ情報が重複していた箇所を示している。黒い部分が情報を取得した箇所であり、色の薄い箇所は、他のエージェントも情報を取得した場所であることを示している。これは探索した情報は、互いのエージェントが近づいて初めて行われるため、ある場所で情報を取得したエージェントが離れ、その後別のエージェントが来た場合は、そこが探索済みである事を把握しないまま、再び情報を取得することを示している。

なお、この実験で、エージェントの取得した情報が重なった数は 159345 であり、半分以上が重複したデータである事を示している。

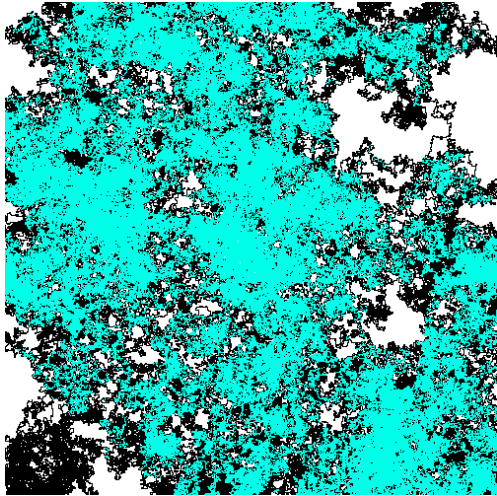


図 4.6 情報を交換しない場合のエージェントの動作

図 4.6 は比較対象として、エージェントが相互に情報交換を行わない場合の実行結果を示している。この場合、全てのエージェントは自身の探索した情報以外は知る事がないので、ほぼすべての記憶容量を使い切って終了している。情報交換の処理がないため、処理時間も短くなるが、図に示す通り大半の個所が重なっており、結果としては交換する場合よりも、探索した箇所が少なくなっている事が分かる。

図 4.7 は、それぞれのスレッドの動作結果をグラフに表したものである。グラフには、エージェントが移動した回数と、記録した情報の数を示している。このグラフからは全 40 のスレッドのうち、記憶容量を使い切って終了したものが 13 個、移動回数が限界となって終了したものが、27 個存在していることがわかる。ものによっては、限界まで移動しても、殆ど記録していないものもある。これは移動をランダムで決定しているため、探索済みの個所を延々回り続けていた可能性がある。

全体で情報を記録した数は 296884 になっている。これは全エージェントの記憶容量の合計 (400000) のうちの約 75% であり、まだエージェントの記憶できる容量には余裕があることを示している。

以上、情報収集の為の分散アルゴリズムをネットモデルで設計、これを開発しているエンジンプログラムによって実験を行った。相互の探索情報の交換により、単純な探索よりも全体を把握しており、より広く探索を行うことは確認できた。これらの結果について発表論文 2 で報告を行った。

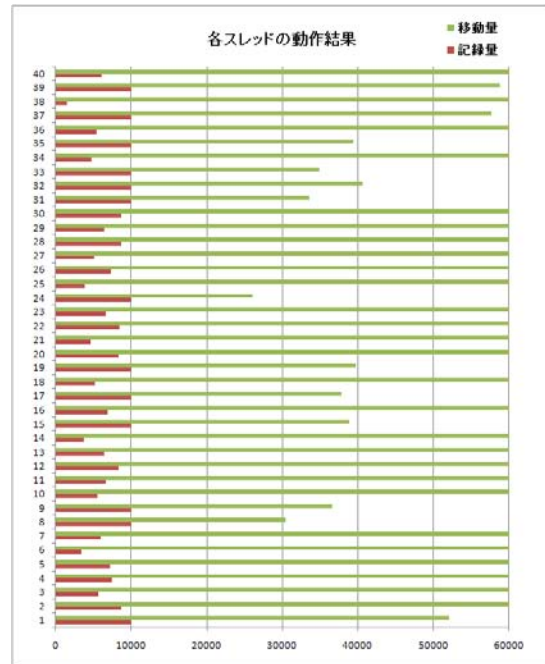


図 4.7 エージェントの移動回数と記録量

## 5. 主な発表論文等

(研究代表者、研究分担者及び連携研究者には下線)

[学会発表] (計 2 件)

- ① 山口真之介, 和崎克己, 師玉康成: 拡張ペトリネットを用いた情報収集の為の分散アルゴリズムの設計, 電子情報通信学会技術報告 CST2009-109, pp83-88, 2009 年 11 月 27 日, コンカレント工学研究会, 名古屋大学.
- ② 山口真之介: ペトリネットモデルを用いた, グラフィックのアニメーション制御についての一提案, 電子情報通信学会技術報告 CST2008-108, pp. 41-46, 2008 年 8 月 5 日, コンカレント工学研究会, 静岡大学.

## 6. 研究組織

### (1) 研究代表者

山口 真之介 (Yamaguchi Shin' nosuke)  
九州工業大学・大学院情報工学研究院・助教  
研究者番号: 00380733