6F−7 オブジェクトネットワークを利用した並行システムのペトリネット表現

守屋 洋　鴨志田 稔　村尾 洋　榎本 肇

芝浦工業大学

1. はじめに

ユーザから提起される多様な要求に対し、システムは機能を適当なレベルで基本的なモジュールに分け、それらを必要な順序で縦続的に、あるいは並行的に組み合わせ、積み上げることにより処理を進める。各要求に対するこのような処理のしかたとユーザ要求の相続く発生とから、システムは処理の階層構成と各層における種々の並行処理を必要とする。

一般に処理は名詞オブジェクトと動詞オブジェクトからなるオブジェクトネットワーク上の移動として考えられる。複数のプロセスが並行処理される場合はオブジェクトネットワーク上の移動に関する制御が必要となり、その明確な表示を工夫しなければならない。

本論文では、1）並行処理の順序制御、2）任意時における階層的並行処理の進行状態表示、の記述を明確にするため、新たに拡張したペトリネットの利用を提案し、具体例として、分野記述言語[1]の一種である画像システム記述言語（ＷＥＬＬ−ＰＰＰ）[2]をとりあげ、その記述構成のペトリネット表現を示す。

2. オブジェクトネットワークと並行プロセス

2. 1. オブジェクトネットワークの定義

名詞オブジェクトのクラスをノードで示し、そのノードどうしはブランチで接続されている。ブランチには方向があり、それにしたがいオブジェクトがノード上を移動する。このようなブランチは関数であり動詞オブジェクトであるといえる。移動には実行中の状態を示すノードとブランチが必要である。

2. 2. 並行プロセス

独立な制御を受ける処理の単位をプロセスとし、複数個のプロセスの進行が考えられるシステムを並行プロセスシステムとする。とくに複数個のプロセスが1つのまとまった仕事を遂行する場合、ある操作の実行や同一資源へのアクセスに際しプロセス間に相互作用が生じ、順序等を規定する必要が生ずる。並行プロセスをオブジェクトネットワーク上で実現させるには、ノード間の移動を制御することが必要となる

3. システムのペトリネット表現

一般のペトリネットはプレースとトランジションと呼ばれる2種類のノードを持つ特殊な有向グラフである[4]。通常、黒丸で表現されるトークンをプレース上に置くことによりその状態をとっていることを表すが、画像システムのようなインタラクティブシステムの順序制御と並行処理を行わせるペトリネットを表現するためにさらに2種類のトークンを追加、合計3種類のトークンを用いる。

1）マスタートークン：各プロセスが現在どのような状態にあるかを示す（●印で表現）。

2）ポシビリティトークン：各プロセスの処理の開始許可を示す（▲印で表現）。

3）実行トークン：クライアントからの処理要求の発生を示す（■印で表現）。

このことより、実行を進行させるためには、これら3種のトークンが共にそろってからトランジションを発火(fire)させる（このトランジションを □ で表現）。この場合のトランジションを本来の状態の移動を表す場合と区別して”論理動作”を表すトランジションと呼び、この場合は条件を満たすとすぐに発火する。このようなペトリネットをゲート制御型ペトリネットと呼ぶ。また、拡張ペトリネットとしてプレースにトークンがない場合に限り発火できることを示す抑止アーク(inhibition arc:——○で表現[4])を利用している。

4. 画像システム記述言語のオブジェクトネットワークと平行プロセス

4. 1. WELL-PPPとオブジェクトネットワーク

画像システム記述言語（WELL-PPP）は、カラー画像処理・描画プロセスの効率よい記述とユーザフレンドリなシステムの構築を目的に設計されている。画像データをそれに対するオペレーションと結合しオブジェクトとしノードと考え、状態の変移を示すものとして関数を導入しブランチとすることにより、描画用オブジェクトネットワークが構成されている。ユーザが関数をリクエストすると、実行に必要なデータの要求を行い、すべてがそろったところで関数の実行を行う。もし、要求したデータがない場合は、データ駆動でそのデータを生成する関数をリクエストする。このオブジェクトネットワーク上にトークンを走らせることが描画もしくは処理のための手段実行に対応することになる。

4. 2. 複合、個別オブジェクトと並行プロセス

ユーザのリクエストした画像はピクチャ要素と呼ばれる基本的なオブジェクトの複合であるととらえ、それを複合オブジェクト、またピクチャ要素を個別オブジェクトと呼ぶ。個別オブジェクトはあるクラスオブジェクトのうち特定化された固有名詞オブジェクトであるともいえる。ピクチャ要素は上記の描画用ネットワークの処理を終わり、輪郭と色・明るさの情報がともにそろっているものと考えられその生成は輪郭と色・明るさの生成がそれぞれ項目と属性[2]として独立に制御されうることから並行プロセスになる。ここでは例として”雪だるま”を生成する場合をあげる。複合オブジェクトが雪だるま全体、個別オブジェクトがそれぞれ胴、顔、手でさらにそれぞれが輪郭、色・明るさから生成されていると考えられる。

4. 3. システムの階層表現と並行化

以上の結果を含め、画像データであるオブジェクトの描画または処理のシステムを次の4つの階層に分けて考える；

［1］プロセス構築層：複合オブジェクトの生成リクエストを受け、個別オブジェクトを生成するプロセスを決定する

［2］構造ネットワーク層：個別オブジェクト間の構造のオブジェクトネットワークを示し、処理順序を制御する

［3］要素ネットワーク層：個別オブジェクトである画像データクラスを作成する

［4］関数実行層：実際に関数を実行する

Petri Nets Representation of Concurrent System Described by Object Network
Yo MORIYA , Minoru KAMOSHIDA , Yo MURAO , Hajime ENOMOTO
Shibaura Institute of Technorogy

なお、これらの層の関係はリクエストーレスポンド方式に従っている。
次に並行プロセスの生ずる状態分けを階層別に示す。
A：プロセス構築層に複合オブジェクトを生成する特定のリクエストが複数存在する場合。
B：構造ネットワーク層において、複数個の個別オブジェクト間に特定の処理順序の規定がない場合(雪だるまの顔と手)。
C：要素ネットワーク層において、項目と属性の処理を並行的に行う場合(胴の輪郭と色・明るさ情報)。

Aの場合について並行システムへの対応方法を示す（図1）。1）プロセス構築層にリクエストされた複数の複合オブジェクトに対しそれぞれ処理順序を決定し、各複合オブジェクトに関する構造ネットワークを作成する。2）各複合オブジェクトの構造ネットワーク層から要素ネットワーク層へ個別オブジェクト作成のリクエストを出す。3）要素ネットワーク層は、各構造ネットワーク層ごとに割り当てられる。4）構造ネットワーク層からのリクエストがすべてレスポンドされると全ての処理が終了したことをプロセス構築層へレスポンドし、完成された複合オブジェクトがシステムから退去する。

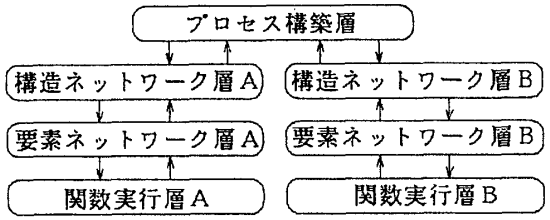

図1　システムの階層表現と並行処理の概念図

5．各層における具体的なペトリネット表現

5．1．プロセス構築層

外部ユーザからのリクエストを受け、そのオブジェクトの処理順序を決定し、次段の層の構造を作成する（図2）。

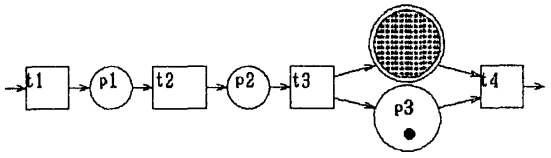

図2　プロセス構築層のペトリネット表現

図2におけるt，pは以下の意味を示す
t1：作成する複合ｵﾌﾞｼﾞｪｸﾄに対する処理要求発生
t2：複合ｵﾌﾞｼﾞｪｸﾄの処理プロセスを決定する
t3：複合ｵﾌﾞｼﾞｪｸﾄの処理開始
t4：複合ｵﾌﾞｼﾞｪｸﾄの処理終了
p1：複合ｵﾌﾞｼﾞｪｸﾄの処理プロセスの決定待ち
p2：複合ｵﾌﾞｼﾞｪｸﾄの処理の開始待ち
p3：下位層にリクエスト中の複合ｵﾌﾞｼﾞｪｸﾄを保持
2重丸：下位の層でオブジェクトの処理実行中
（以下の層についても同じ意味を表す）

5．2．構造ネットワーク層

プロセス構築層によって決定された処理順序に従って要素ネットワーク層にリクエストを出す。ネットワークは各ユーザごとの複合オブジェクトにおける個別オブジェクトの処理順序の条件に従って作成される。従って、雪だるまの例でペトリネットを示す。個別オブジェクト間に作成順序の前後関係がない場合、並行処理が起こることから（状態分けB）、雪だるまの例では胴の生成が顔、手に先行し、顔、手は並行であるとして、ペトリネット表現している（図3）。図は胴を作成中であることを示している。

5．3．要素ネットワーク層

個別オブジェクト作成のためのネットワークを持ち、各関数の実行の段階で

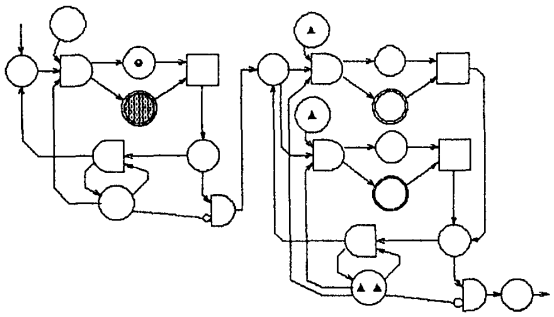

図3　構造ネットワーク層の例のペトリネット表現

i）前段階まで終了していることを示すマスタートークン、ii）注目をしている段階の処理を許可するポシビリティトークン、iii）ユーザからの関数実行要求が存在することを示す実行トークン、の3つがそろってから処理開始となることを示す（図4）。画像システムではインデックス番号などの識別子をつけて作成したオブジェクトの区別を行っている[3]が、それに対応する番号をトークンにつけることにより、ペトリネット上でもそれを区別できる。

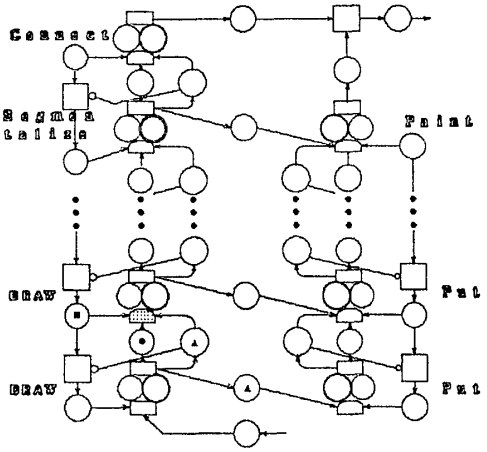

図4　要素ネットワーク層のペトリネット表現

5．4．関数実行層

実際に関数の実行を担当する部分を表す。具体的な処理として、1）Generic 関数をActual関数に変換、2）関数の実行に必要なデータを認識、3）Actual関数の実行、がある。

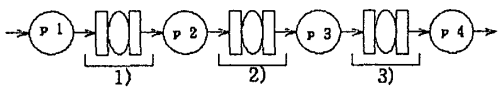

図5　関数実行層のペトリネット表現

但し、図5においてp1～p3のプレースは次の処理の開始待ちであり、p4は要素ネットワーク層へのレスポンド待ちを示す。

6．まとめ

本論文ではオブジェクトネットワークで記述された並行システムが3種のトークン、2種のトランジションを持つ拡張されたペトリネットを考えることにより、その階層構成として制御記述されることを示している。

文　献

[1] 榎本、鴨志田：" 分野記述型言語の構造"，情報処理学会第44回全国大会，1992.3
[2] 鴨志田、丹羽、榎本：" オブジェクトネットワークによる画像システム記述言語"，情報処理学会第44回全国大会，1992.3
[3] 丹羽、鴨志田、榎本：" オブジェクトネットワーク上の実行処理でのインターフェースとプロトコル"，情報処理学会第44回全国大会，1992.3
[4] J.L.PETERSON：ペトリネット入門／共立出版(1984)