

協調型画像描画システムの並行図式

6K-5

守屋 洋 丹羽 直人 村尾 洋 榎本 肇

芝浦工業大学

1. はじめに

システムは相互関連のある機能要素を適切なくつかの機能単位(モジュール)にまとめ、それらの間を周到に関連づけることにより、システムの目的を効果的に達成できるように設計される。以下では、相互作用のあるシステム(interactive system)における協調処理(cooperative process)を一般的に規定し、画像システム記述言語WELL-PPP^[1](Window-based Elaboration Language for Picture Painting)上でインプリメントされる画像描画システムの図式表示を、さきの検討^[2]に基づいてさらに進め、画像描画システムが一つの協調型処理システムであること、協調型処理システムの実現において問題となる複雑なシステムの動きの理解し易い表記に対し一つの方法を与えている。

2. 協調処理とインタラクティブシステム

インタラクティブシステム^[3]とは、各機能の動作が相互に他の機能の動作に影響を及ぼすシステムである。一般に、システムの要素は有機的に関連して動作しているが、単純に一方的でなく、相互的であることが特徴である。それに対し、システムのモジュール化をはかることにより、相互作用に秩序を与えることが設計上重要となる。良いモジュールとは、モジュール内構成要素間の相互関連性が強く、モジュール間の相互関連性が弱い場合である。相互接続されたネットワーク構造で要素間関係が表される場合は相互関連性が強く、単一方向の階層構造で表示される場合は相互関連性が弱いといえる。相互作用のあるシステムが与えられたとき、このような考え方でモジュール化が追求される。

一般に、システムにおける協調処理とは、複数の構成要素がそれぞれ機能を分担し、かつ各機能間の相互作用が互いに他の機能を混乱や機能低下に陥れることなく、要素共通の全体的目的達成に対し、順方向に影響しあうよう仕組まれている処理ととらえられる。相互作用は、各機能のオペレーション実行の時間的順序確立に関して生じたり、同一資源アクセス時に生じたりする。それに対応し、協調処理はそれがどのような意味でなされるかが異なる。いま、モジュール構成されたインタラクティブシステムにおいて、相互作用が特に処理の時間的順序関係に関して生ずる場合を考える。モジュール化に関する上記の観点から、モジュール内の要素は処理順序に関しネットワーク構造で与えられている。従って、もし処理順序決定時の相互作用が必ず各要素の処理進行方向と一致した向きで影響しあうよう構造化されているならば、協調型処理となる。この場合のネットワークは非巡回(acyclic)型となる。各要素の機能がそれぞれ各処理実体(プロセッサなど)と対応していれば、協調型並列処理(cooperative parallel process)となり、対応していなければ、協調型順次処理(cooperative sequential process)となり、いわゆる並行処理(concurrent process)となる。すなわち、並行処理は協調処理の一形態である。

上記のとらえ方からも知られるように、協調処理の実現には要素間の相互通信機能が不可欠である。ここでは、クライアント・サーバモデルにおける(Request-Respond)方式をその効果的な方法として提案する。この相互通信問題も含め、協調処理のインプリメントには、複雑なシステム全体の動きを理解し易く表示することが重要となる。

3. 画像描画における可視化対象

描画システムにおいて画像(Picture)は画像クラス(Picture class)のインスタンスとして定義される。これを名詞オブジェクトとして捉え、それを総称して"可視化対象(visualized object)"と呼ぶ。可視化対象を実現するために画像描画過程を体系化したものがオブジェクトネットワークである。

画像描画システムにおいてはクライアントによって意図されている画像を描画する作業は、その内容に従って、以下の2階層に分類される。

[1] 要素ネットワーク層: 要素画像と考えられる個別オブジェクト^[2]を作成する。一つの要素画像を作成するためには、その輪郭を生成する部分(Frame Section)と色情報を輪郭や領域に従って与える部分(Color Section)が必要になる^{[2][4]}。

[2] 構造ネットワーク層: 要素画像として作成された個別オブジェクトに加影などとして作成された要素画像も個別オブジェクトとしてとらえ、各オブジェクトの位置関係や動きなどについての処理を行う。従って、要素画像データをそれに対するオペレーションと結合し名詞オブジェクトであるノードと考え、状態の変移を示すものとして関数を導入しプラントとすることにより、構造ネットワークが要素ネットワークと同様に構築される^{[5][6]}。

4. オブジェクトネットワークからベトリネットへ

3節において与えられたオブジェクトネットワークによってシステムの動きを表現するためには、テキストなどの形による補助説明が必要である。また、各ネットワーク層に含まれる相互作用のある処理順序や表現するためには、出現状態に依存する処理順序や進行状態を含めて、図式表示する必要がある。そのため、ベトリネットを拡張し、要素および構造の両ネットワークにおける動きを表示する。要素ネットワークについてはさき与えた^[2]が、今回さらにベトリネット表示のモジュール化、階層化の考え方を導入して、構造ネットワークと併せ統一的に示す。

5. 拡張ベトリネットのモジュール化と階層化

上記のような相互関係を持つシステムを記述するため、さきに提案したゲート制御型ベトリネット^[2]を更に拡張し、図式表示についてもモジュール化と階層化を行う。

5.1 ゲート制御型ベトリネット

インタラクティブシステムの順序制御と相互関係処理を行わせるベトリネット表現をするために、さらにアークの拡張として抑止アークに加えて、プレースにトークンがあるという状態が引き金になって処理が実行されるが、トークンを取り除くことはない参照アーク(reference arc; 点線の矢印で表現)を利用する。

5.2 ベトリネットのモジュール表現

オブジェクトネットワークと同様にベトリネットについても基本要素のモジュール化を行う必要がある。それをネットワーク構造にしたがって組み合わせることによってオブジェクトネットワークにそった処理動作を表すことができる。

5.3 ベトリネットの階層性

ある統一された基本表現でシステムを示す場合、必ずしもシステムの動きを詳細に表現できるとは限らない。この場合、ベトリネット表現に階層性を導入して、基本モジュールの一部を細かく表現することが可能。例えば、ポイント定義における作業準備の準備の実行要求ポイントの定義実行要求ポイントの状態⁽⁴⁾⁽⁷⁾を表現した場合右のような図式になる(図1)。

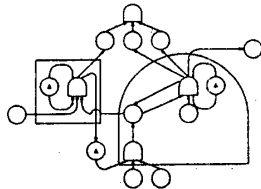


図1 ポイント定義の表現例

6. 要素ネットワークのベトリネット図式

6.1 図式表現の特徴

図式表現の特徴として、以下のものが考えられる。

1. 各生成部間に相互関係を持たせる Identify 機能が導入されている。
2. 輪郭部とカラー部のそれぞれについて基本モジュールを導入する。
3. 各オブジェクトがそれぞれ一つのトークンに割り当てられる。

6.2 輪郭部の基本モジュール

輪郭部で利用する基本モジュール(図2)において各プレースには以下のトークンが入る。

- a: 既作成データを示すマスタートークン
 - b: 注目関数実行を許可するポシビリティトークン
 - c: 注目関数実行の要求を示す実行トークン
 - d: 新規作成データを表すマスタートークン
 - e: 次関数実行を許可するポシビリティトークン
 - f: カラー部転送用マスタートークン
- ここで、dのプレースが次の段階でのaのプレースに、またeが次の段階のbに変化して利用される。

6.3 カラー部の基本モジュール

カラー部で利用する基本モジュール(図3)において各プレースには以下のトークンが入る。

- f: 輪郭部での生成データを示すマスタートークン
 - g: カラー部の既作成データを示すマスタートークン
 - h: 関数yの実行要求待ちを示すマスタートークン
 - i: 新規作成データを表すマスタートークン
 - j, k: 各関数実行の許可を示すポシビリティトークン
 - l, m: 各関数の実行要求を示す実行トークン
- iのプレースが次の段階におけるgのプレースに変化して利用される。

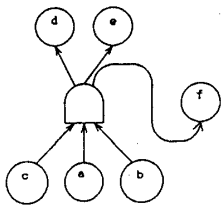


図2 輪郭部における図式表現の基本形

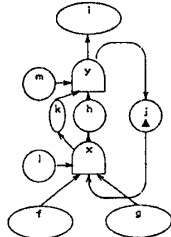


図3 カラー部における図式表現の基本形

6.4 要素ネットワークの全体表現

各部分ごとにこの図式表現をはじめに挙げた要素ネットワークに従って接続すると以下のようなになる。ただし、表現を簡略にするため、カラー部の基本モジュールに省略型を用いている。この省略型は図3

に示した図式表現の2つの関数をもつにまとめ、ポシビリティトークン用のプレースを削除したものである。

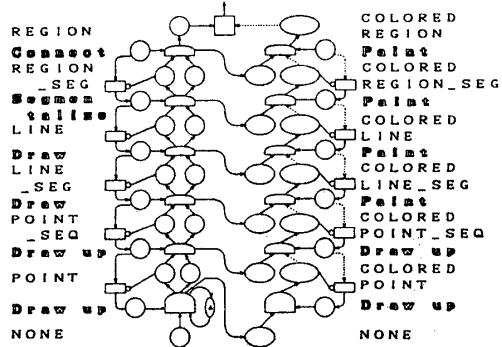


図4 要素ネットワークの全体表現

7. 構造ネットワークのベトリネット図式

要素ネットワークと同様に、構造ネットワーク層のオブジェクトネットワークに従ったシステム処理の進行を示す図式表現を与える。従って、構造ネットワーク用にもベトリネットのモジュール化を行い、基本モジュールを導入する。このモジュール表現はほとんど6.2であげたモジュール表現と同様の意味を持ち、相互関係を持つ処理が実行される。

8. 協調型画像描画システム

3節以下で示した描画システムは、2節で述べた協調処理の観点から以下のように考える。システム内における各ネットワーク層を各可視化対象実現のためのモジュールであると考え、そのモジュール内の各要素がネットワーク構造として与えられているうえ各処理目的に向かって順方向に接続されているのである。この描画システムは協調型で機能しているため、また、各ネットワークの要素機能が各処理実体と対応していないので、そのベトリネット図式が並行プロセスの表示として与えられ、システム間における協調処理の動的構造を協する事ができる。一方、ユーザとシステムの間にも協調処理が考えられ、それに従ったインタラクティブ仕様⁽³⁾を与える事によってその構造が得られる。

9. まとめ

インタラクティブシステム、協調処理について述べた後、描画システムの階層構造について述べた各階層におけるオブジェクトネットワークが拡張されたベトリネットを考へることによって並行図式記述された。また、これは協調処理を持つシステムとしてとらえらることを示した。本研究は、芝浦工業大学情報構造工学研究として、重点領域「高度ソフトウェア」(課題番号04219102)の補助を受け、KDD、富士通の支援のもとに進められてきたことを記して謝意を表します。

文献

[1] 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
 [2] 守屋、鴨志田、村尾、榎本: "オブジェクトネットワークを利用した並行システムのベトリネット表現", 情報処理学会第44回全国大会, 1992.3
 [3] 橋本、村尾、榎本: "システム間インタラクションの仕様記述とそれによる画像ソフトウェアへの応用", 情報処理学会第46回全国大会, 1993.3
 [4] 丹羽、守屋、関、村尾、榎本: "並行実行型画像システム記述言語(Concurrent WELL-PPP)の実現", 情報処理学会第46回全国大会, 1993.3
 [5] 宮本、猪野、守屋、村尾、榎本: "画像の質感・実在感における構造ネットワーク", 情報処理学会第46回全国大会, 1993.3
 [6] 猪野、宮本、守屋、村尾、榎本: "動画像表現における構造ネットワーク", 情報処理学会第46回全国大会, 1993.3
 [7] 平井、丹羽、守屋、村尾、榎本: "Concurrent WELL-PPPにおけるサービス管理", 情報処理学会第46回全国大会, 1993.3