

分散独立アクティブ DB による ワークフロー管理のためのフロー解析手法

小林昌徳[†] 横田治夫[‡]

[†] 東京工業大学 大学院情報理工学研究科 計算工学専攻

[‡] 東京工業大学 学術国際情報センター

masanori@de.cs.titech.ac.jp, yokota@cs.titech.ac.jp

あらまし 我々は分散独立アクティブ DB 間のメッセージ通信によって分散ワークフローを実現するメカニズム (DIADEM) を提案してきた。DIADEM による分散ワークフローでは各サブモジュールが独立性を維持しながらそれぞれのワークフロー間で連携をとるため、他のサブモジュール内の挙動はブラックボックスである。このブラックボックスな部分を階層構造にして隠蔽するため、今回我々は、DIADEM におけるワークフロープロセスを階層的カラーペトリネットを用いてモデル化、フロー解析を行うことを提案する。本稿では、階層的カラーペトリネットによるプロセス定義の検証についてその手法を述べる。

キーワード 分散ワークフロー、階層的カラーペトリネット、アクティブデータベース、プロセス定義

The flow analysis technique for workflow management using Distributed Independent Active Databases

Masanori Kobayashi[†] and Haruo Yokota[‡]

[†] Department of Computer Science, Tokyo Institute of Technology

[‡] Global Scientific Information and Computing Center, Tokyo Institute of Technology

masanori@de.cs.titech.ac.jp, yokota@cs.titech.ac.jp

Abstract We have proposed a mechanism named DIADEM for a distributed workflow system constructing nested transactions spread over distributed independent active databases. In the DIADEM, the inside behavior of a submodule is a black box from the other submodules, because each organization corresponding to a submodule is independent only by communicating each other with messages. To conceal such a part into a layer structure, we propose a method using a hierarchical colored Petri Nets to model and analyze workflows. This paper discusses the verification of the workflow-process definition on the DIADEM with the Petri Nets.

key words distributed workflow, hierarchical colored Petri Nets, Active Database,
workflow-process definition

1 はじめに

日本経済の低迷にも底打ち感が認識され始めたが、各企業にとって生き残りをかけたビジネス再編の必要性は依然変わらない。近年、ビジネスプロセスにおけるコスト削減、時間短縮、情報共有の効率化、そしてBPR(Business Process Reengineering)の実現手段としてワークフロー管理システムの導入が有効と期待されている。

ワークフロー管理団体のWfMCによるとワークフローはビジネスプロセスの管理と自動化と定義されている。既存のワークフロー製品は定型業務を対象にしてプロセス定義としているものが多い。そしてインスタンスをトランザクションの保護の下で実行するトランザクショナルワークフロー [1] が、データベースの一貫性制約を達成する上で信頼性が高いと認識されている。また、単一の組織内での閉じたワークフローだけでなく、他組織間にもたがる分散ワークフローのニーズがある。

例えば、テレビ局における定時のニュース編集を想定してみる。まず社会部や政治部、経済部といった取材各部が重大性の高いニュースを編集部へ提出する。放送すべきニュースを編集部が決定しニュース項目の時間配分を割当てる。仮にワールドカップ2002の日本の決勝トーナメント進出のニュースを社会部が担当するとしよう。一方、経済部はワールドカップ2002の日本経済への影響を伝えたいと考える。編集部ではニュース内容とその時間配分を決定し、編集計画を提出するよう社会部と経済部へ同時に要請する。ニュース内容や時間配分はその場で変化していくが、(1)取材各部からニュース項目の提出、(2)編集部が放送するニュース内容とその時間配分を決定、(3)編集部から取材各部への編集計画提出要請、(4)取材各部が編集計画を編集部へ提出、(5)提出された編集計画に基づいてニュースを編集といったプロセスは変わらない(図1参照)。

各部が協調してニュース編集プロセスを実現していくが、各部内部でのプロセスは他の各部へ公開する必要はないだろう。即ち、各組織が独立性を維持しながら、協調動作するワークフローを実現しなければならない。

我々は、分散かつ独立配置したアクティブデータベースシステム間のメッセージ通信によってそ

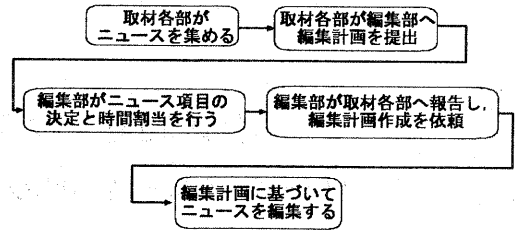


図1: ニュース編集のプロセス

れを実現する方式を提案してきた。これを DIADEM(Distributed Independent Active Database with subtransaction Exporting Mechanism) と呼ぶ [8, 9, 10]。DIADEMは、アクティブルールにより発生するサブトランザクションを、その階層構造を保ちながら独立した異なるアクティブデータベース上で実行することを特徴とする。しかし各サブモジュール内のルールはブラックボックスであり、全体としての協調動作が仕様通りであるかを知る手段はなかった。

ところで、ワークフローはアクティビティの系列としてみるができるため分散事象システムである。またペトリネットは分散事象システムのモデル化および解析に優れたツールであることから、ワークフローへ適用研究は既に行われている [2, 3, 4]。

我々は DIADEMの他サブモジュールにおける見えない処理を階層的カラーペトリネットの階層化の概念を用いて処理を委譲することで、プロセス定義の検証を行い、実行可能性 [5] や安全性、入力関係の整合性の判定を行うことを提案する。

本稿では、階層的カラーペトリネットを用いて DIADEMにおける分散ワークフローのプロセス定義を検証する手法を提案する。まず第2章で、我々が提案している DIADEMについてその特徴を紹介する。次に第3章ではワークフロープロセスのモデリングに用いる階層的カラーペトリネットについて解説する。第4章で、DIADEMにおけるワークフローへの階層的カラーペトリネットの適用方法を述べる。第5章で、ワークフロープロセスの検証についてその有用性と検証方法を示す。第6章では、テレビ局のニュース編集のワークフローを検証例とし、DIADEMにおける階層的カラーペト

リネットを適用することの有効性を確認する。最終章で本稿のまとめと今後の課題を述べる。

2 DIADEM

アクティブデータベースによる分散ワークフローメカニズム DIADEM は、ワークフロープロセスインスタンスを1つのトランザクションとし、入れ子トランザクション構造により管理され、処理されていく。この章では、まず DIADEM の特徴的なトランザクションモデルと実行主体となるアクティブデータベースを簡単に紹介し、DIADEM のメカニズムと現状の問題点を指摘する。

2.1 トランザクションモデル

ワークフロープロセスのインスタンスは生存時間が長いケースが多い。これを伝統的な ACID 特性を持つトランザクションモデルとして管理すると、システムのスループットが低下することが知られている。そのため DIADEM では、1つの親トランザクションを複数の子トランザクションに分割、更にその子トランザクションの分割を許す入れ子トランザクションモデル [9] を用いてインスタンスの実行管理を行う。また、親トランザクションのアポートの影響を受けて既にコミットした子トランザクションを補償する際は補償トランザクション [7] を用いる。

2.2 アクティブデータベース

従来のデータベース (Passive Database) は、ユーザが問い合わせをして初めて処理を行うのに対して、アクティブデータベースでは ECA(Event-Condition-Action) ルールに従い、イベントをトリガし評価、それによってアクションを起動する [10]。このように、ルールベースとデータベースを統合することでデータベースシステムが自律的に動作することを可能としている。

このアクティブデータベースを分散配置し、それらが互いにメッセージ通信し合いながらワークフローを実現しようというのが DIADEM である。

2.3 DIADEM

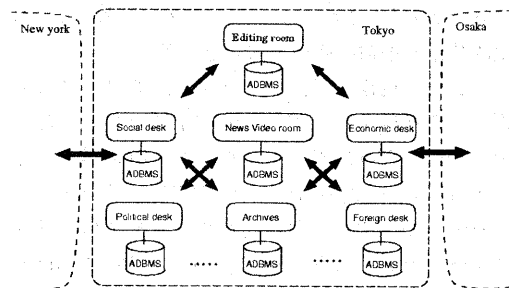


図 2: 例: テレビ局のニュースネットワーク

DIADEM は互いに独立したアクティブデータベースを1つのサブモジュールとし、サブモジュール間でネットワーク通信を行いながら、全体が協調してワークフローを実行するシステムである。

テレビ局におけるニュースネットワークに DIADEM を採用した例を図 2 に示す。東京本社内において、ニュース編集を行う編集部、取材部に相当する社会部や政治部、経済部、外信部、24 時間最新のニュースが到着するニュースビデオルーム、過去のニュースビデオをデータベース化して検索可能にしている資料映像室がある。DIADEM では、これらの各部局が独立なアクティブデータベースを持ち、互いにメッセージ通信し合う事で全体として協調した動作を行っている。また、東京本社の外にはニューヨーク支社や大阪支社があり、その中でも DIADEM のサブモジュールが分散独立して協調動作を行っている。

ワークフローが処理されるに従い、アクティブデータベースの ECA ルールによって生成されるサブトランザクションは、互いに入れ子となりながら異なるサブモジュール上で実行され、全体として一つのトランザクションを形成する。

各サブモジュールのアクティブデータベースは、自サブモジュールにおいて処理されるサブトランザクションに対してのみ責任を持つ。このため、各サブモジュールには既存の DBMS を利用することが可能となる。DIADEM のサブモジュールは、その既存の DBMS 上にアクティブルールの実行と階層構造の自サブモジュールに関する部分を保持し、DIADEM 全体としてワークフローを構成する。

DIADEMでは、ECAルールにおけるE-C、C-Aの発火をカップリングモードにより明示的にユーザが指定する。これにより、子トランザクションの生成時期と補償の範囲が決定される。

しかし、現段階ではワークフロープロセスのモデリング手段を持たず、その結果プロセス定義の検証やモニタリングといったフロー解析ができない状況にある。WIMCのワークフローリファレンスモデルの示すプロセス定義ツールやモニタリングツールを実現するためにも、DIADEMにおけるプロセスモデリングのフレームワークを示す必要性がある。その際、他のサブモジュール上でのプロセスは階層化し、自サブモジュールにおいて仕様のみを公開することで、サブモジュール間の独立性を維持しなければならない。

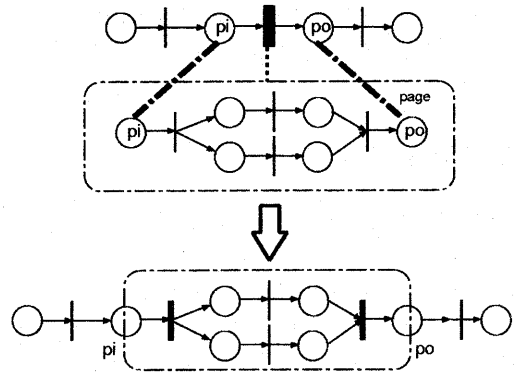


図3: 階層的カラーペトリネット

3 階層的カラーペトリネット

ペトリネットに階層的な記述を導入したものが、階層的カラーペトリネット (HCPN: Hierarchical Colored Petri Nets)[11]である。

一つのトランジションに別のペトリネット (ページ) を対応させ、ページは更に異なるページによって構成されることもある。図3のようにHCPNは階層構造を持たないペトリネットと等価変換できる。

HCPNは以下の要素から構成される。

1. 非階層的カラーペトリネット (ページ) の集合 S
2. 代入節点 SN : トランジションの部分集合
3. ページ割当 $SA: SN \rightarrow S$
4. ポート節点 PN : プレースの部分集合
5. ポート型 $PN \rightarrow \{in, out, i/o, general\}$
6. ポート割当 $PA: SN$ の各節点 $t \mapsto (t$ に隣接する節点, $SA(t)$ のポート節点) の対の集合

4 HCPNのDIADEMへの適用

4.1 HCPNを用いる利点

ワークフロープロセスのモデリングにHCPNを適用する理由を以下に示す。

グラフィカル表現 ペトリネットは視覚的に表現でき、その動的挙動も視認できる。

形式的意味論 形式的に定義されているペトリネットによるワークフロープロセスは、明快で精確な意味論を持つ。

表現力 HCPNはシーケンス、AND分岐、OR分岐、繰返し、入れ子構造など、ワークフローにおける経路構造全てをモデリングすることができる。

解析力 ペトリネットは、可達性、有界性、活性等の特性分析や、レスポンスタイムや生起確率等の性能分析における多くの解析手法が提案されている。それらを用いてワークフロープロセスを解析することで、プロセスの改善や他のワークフローとの相対的評価が可能になる。

ベンダー独立 ペトリネットは、ワークフロープロセスのモデリングや分析においてツールに依存しないフレームワークを提供する。

階層構造 階層構造を用いることで、トップダウンなプロセスのモデリングが可能になる。

またDIADEMにおいては、自分以外のサブモジュールの挙動は把握することができない。従って、他のサブモジュールに委譲されるプロセスにHCPNの階層部分を割当てることにより、自サブモジュールではあたかも1つの非階層的カラーペトリネットを解析しているかのように扱うことができる。

4.2 ECAルールへのマッピング

ペトリネットとワークフロープロセスとの対応は表1の通りとする。

表1: ワークフロープロセスとペトリネット

ワークフロープロセス	ペトリネット
事例	トークン
事前条件	入力プレース
事後条件	出力プレース
アクション	トランジション

ECAルールからペトリネットへのマッピングの例を図4, 5に示す。

図4はテレビ局の資料映像室におけるECAルールの例である。ニュース編集において、余ったニュース放送時間Y分を埋めるため、ある部局XがT分前後の映像を資料映像室へ要求する。これがEventになる。次に、資料映像室のサブモジュールは、最もT分に近いT分以内の映像を検索する。これをConditionとする。もし、Conditionを満たす映像Vを見つけることができたならば、部局Xに対して映像Vを送信する。これがActionに相当する。

```

E: When receive message
    from X to get video T
C: If there is a video V
    most near T
A: Then send video V to X
    
```

図4: ECAルールの例

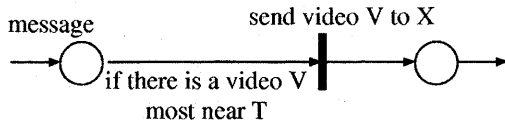


図5: ECAルールからHCPNへのマッピング

図4をマッピングしたペトリネットが図5である。プレース *message* に「X to get video T」というトークンが入ることでEvent発生と見なし、アーク関数「if there is a video V most near T」という

Conditionを評価する。真ならば、「send video V to X」というActionを発火させる。

図4, 5から分かるように、入力プレースにトークンが入ることをEventと見なし、あるトランジションに対して入力プレース全てにトークンが入ることによりConditionを評価、真ならばトランジションが発火し、これをActionの実行とする。このようにしてECAルールをペトリネットへマッピングしていく。

5 DIADEMの検証

5.1 HCPNを用いた検証の有用性

プロセス定義の検証とは、プロセス定義が要求仕様を満たし、またデッドロックなどの挙動上の欠陥の有無を検出するプロセスである。HCPNの構造解析に基づく特性の検証と、挙動をシミュレーションしその実行結果を分析する方法もある。

生成したHCPNを解析することにより、以下のような有用性を見出せる。

実行可能性 可達木を生成し状態遷移の全探索を行う。期待する終端状態に達しない場合、デッドロック発生の可能性をチェックできる。

安全性の判定 デッドロックが生じず期待する終端状態に達することが分かっても、危険状態を経由するパスが存在する可能性もある。可達木を調べることによりそのプロセスの安全性を調べることができる。

入出力の整合性の判定 初期状態と終端状態におけるトークンの持つ情報を比較することにより、入出力関係が一致するか否かを判定できる。

5.2 検証方法

DIADEMにおけるプロセス定義の検証は、サブモジュール内のHCPNアナライザが互いに通信し合いながら協調的に行われる。ここで、他のサブモジュールにおいて外部との通信を行うECAルールの仕様は公開されているとする。

仕様に対して初期状態を与えて検証を開始することで、可達木をHCPNアナライザが生成、階層

部分は他のサブモジュールに処理が派生し可達木の生成が行われる。その結果、全体として一つの可達木が生成され、ユーザはそれを用いてデッドロックや安全性、入出力の整合性の判定を行うことができる。

6 例：テレビ局におけるニュース編集

この章では、ワークフロープロセスの例としてテレビ局のニュース編集を挙げ、デッドロックを検出する検証過程を追ってみる。

6.1 ワークフロープロセス定義

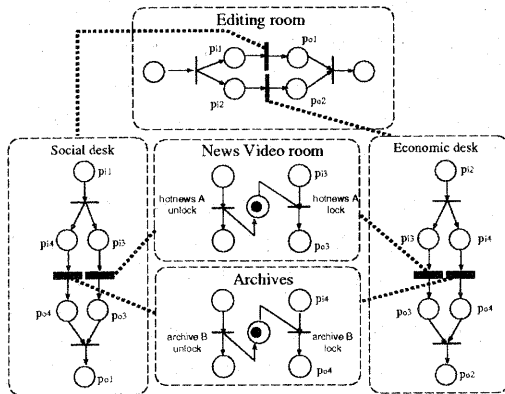


図6: 社会部、経済部へニュース編集計画提出を求めるプロセスの一部

報道局における数分程度のニュース番組の編集プロセスを例にとる。編集部が原稿に合わせた映像の編集計画を社会部と経済部へ依頼する。社会部では最新ニュース映像 (hot news A) と、資料映像 (archive B) を使用することを決めた。一方、経済部でも、社会部と全く同様の映像 A と B を使用することを決めたとする。

上記のような事例がたどるプロセス定義を図6に示す。数分程度という短いニュース番組において、異なるニュース項目で同様の映像を用いることはないという仮定があるとすると、社会部と経済部は互いに1つの映像の使用権を争う可能性がある。

6.2 プロセス定義の検証

プロセス定義の検証を行う様子は、各サブモジュールが持つ HCPN シミュレータによって協調的に行われる。

1. 編集部において検証が開始されると、編集部における HCPN をマーキングが遷移、 p_{i1} あるいは p_{i2} へトークンが入ると、検証はそれぞれ社会部、経済部へ派生する。
2. 同様に社会部と経済部において、 p_{i3} , p_{i4} へトークンが入ると、それぞれニュースビデオルームと資料映像室へ再び派生していく。
3. 図6では、異なるニュース項目で同様の映像を用いることはないという仮定があり、社会部と経済部は互いに映像の使用権を争うことになる。
4. 結果的に編集部においては社会部のページが発火できない状態 (p_{i1} と p_{o2} で遷移不能)、あるいは経済部のページが発火できない状態 (p_{o1} と p_{i2} で遷移不能)、両方とも発火できない状態 (p_{i1} と p_{i2} で遷移不能) のいずれかで停止してしまう。

図7は発生したデッドロックの中でも、社会部と経済部が互いに資源の奪い合いを起こしたことによるデッドロックが起こった直後の様子である。社会部は、ニュースビデオルームにおけるホットニュース A の使用権を奪うことができたが、資料映像室における資料映像 B の使用権を奪えないため、停止してしまっている。対照的に経済部では、資料映像室における資料映像 B の使用権を奪うことができたが、ニュースビデオルームにおけるホットニュース A の使用権を奪えないために停止している。その結果、編集部では社会部に派生した処理も経済部へ派生した処理も終端状態に到達することができず、それぞれに対応するトランジションが発火できないため、 p_{i1} と p_{i2} にトークンが入ったまま停止する。

このようなデッドロックの発生は編集部側で把握する必要があるが、それと各サブモジュール内はブラックボックスであるという分散ワークフローの仮定は矛盾する。

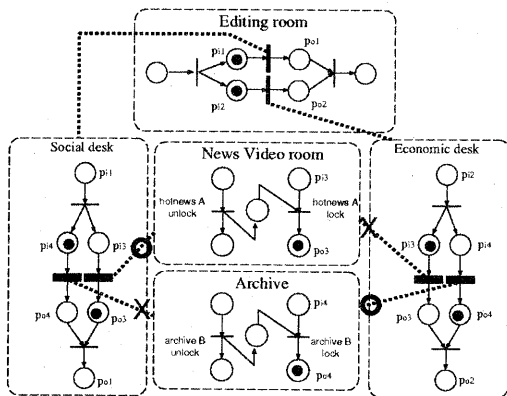


図7: デッドロックに陥ったマーキング例

デッドロックが発生したという情報伝達は以下のようにして行う。A,B,Cという3つのページがあり、A(上位)→B→C(下位)という階層構造を持つとする。図7では、Aは編集部のページ、Bは社会部と経済部のページ、Cはニュースビデオルームと資料映像室のページにそれぞれ相当する。

1. ページBにおいて階層構造を持つトランジション(ページC)が発火できない場合、ページCに問題があることを上の階層(ページA)へ伝える。ページCを用いているという内部情報をページBはページAへ伝えることになるが、ページCの様子は公開されているため問題がない。
2. ページBにおいて階層構造を持たないトランジションが発火できない場合、ページBに問題があることを上の階層(ページA)へ伝える。

このように図7のようなデッドロックは、社会部や経済部のHCPNを検証しただけでは発見できない。分散したサブモジュール間で協調したワークフローを行おうとした過程で発生するものもある。そのような分散ワークフローの検証においてHCPNでモデリング、検証する手法は有効である。

7 おわりに

本論文では、アクティブデータベースを独立に分散配置することで分散ワークフローを実現する

DIADEMにおいて、ワークフロープロセスのモデリングにHCPNを適用し、プロセス定義の検証を行う手法を提案した。HCPNの階層構造を利用して他のサブモジュール内部の処理は隠蔽しながら検証を行い、サブモジュール間の連携において発生するデッドロックを検出する過程を例を挙げて示した。

今後の課題として、モニタリングやワークフロー分析を考慮できるHCPNへ拡張し、その手法を提案、有効性を示すために実装を行う必要がある。またECAルールにおいて、データベースへの書込みや読み出しは擬似的にHCPNへモデル化しなければならないが、そのマッピングルールを検討する必要がある。

参考文献

- [1] M. Rusinkiewicz and A. Sheth. "Specification and Execution of Transactional Workflows". In W. Kim, editor, *Modern Database Systems: The Object Model, Interoperability and Beyond*. ACM Press, 1994.
- [2] W.M.P. van der Aalst, K.M. van Hee, G.J. Houben. "Modelling and analysing workflow using a Petri-net based approach". 1998.
- [3] W.M.P. van der Aalst. "The Application of Petri Nets to Workflow Management". *The Journal of Circuits, Systems and Computers*, 8(1):21-66, 1998.
- [4] 山口 真悟, 葛崎 偉, 田中 稔. "ペトリネットによるワークフローのモデル化と解析". システム制御情報学会, システム/制御/情報 第45巻 第8号, 2001.
- [5] Mohan Kamath and Krithi Ramamritham. "Correctness Issues in Workflow Management". *Distributed Systems Engineering (DSE) Journal : Special Issue on Workflow Management Systems*, Volume 3, Number 4, December 1996.
- [6] 井川 智崇, 宮崎 純 and 横田 治夫. "独立アクティブデータベース間の入れ子トランザクシ

ョンの実現”. 第10回データ工学ワークショップ
ブ, Mar 1999.

- [7] Hector Garcia-Molina, Kenneth Salem.
"SAGAS", ACM SIGMOD, 1987.
- [8] 井川 智崇. 分散独立アクティブデータベース
による障害ハンドリング. Master's Thesis, 東
京工業大学, 2001.
- [9] 井川 智崇, 宮崎 純 and 横田 治夫. "Java のス
レッドを用いた分散入れ子トランザクションの
実現". 情報処理学会研究会報告, データベース
システム DBS-119-60, pages 357-362, 1999.
- [10] 井川 智崇 and 横田 治夫. "分散独立アクティブ
データベース上でのワークフロー用トランザ
クションモデル". 電子情報通信学会, 信学技法
DE2000-86, pages 119-126, 2000.
- [11] 青山 幹雄, 内平直志 and 平石邦彦. "ペトリネッ
トの理論と実践". システム制御情報学会, pages
47-49, 1995.