

4K-3

# 長時間トランザクションにおける タスクフローを用いたトランザクションの実行制御\*

鈴木 幸一郎, 渡邊 豊英†

{ksuzuki, watanabe}@watanabe.nuie.nagoya-u.ac.jp

名古屋大学大学院 工学研究科 情報工学専攻‡

## 1 はじめに

今日、計算機システムの利用用途は単一的な処理から複合的な処理へと拡大し、それぞれの処理要求間の連携的、統一的な実行制御機構が重要となっている。データベースでは、このような長期的に渡って計算機システムに存在し、相互に関連する複数の処理要求を効率的、効果的に実現する課題として長時間トランザクション処理がある。

本稿では長時間トランザクションを効率的、効果的に処理するための枠組みを提案し、かつ相互に関連するが、個々に独立なトランザクション（処理要求）を迅速、かつ的確に実行する方法について検討する。すなわち、長時間トランザクションの処理を三階層トランザクション・モデルの下に構成し、最下位層における最小の実行単位であるタスクをタスクフローと呼ぶ実行制御関係を表したフローグラフにて制御する。

## 2 三階層トランザクション・モデル

長時間トランザクション処理のためのモデルとして Nested Transaction Model[1] や、SAGAS[2] 等のトランザクション・モデルが提案されている。Nested Transaction Model ではトランザクションの並行的な実行を可能にしているが、コミット、アポートのタイミングが下位層に依存しているため処理に遅延が生じる可能性が高い。また、SAGAS では補償トランザクションの概念を導入し、他のサブトランザクションに関係なくコミット可能な環境を提供しているが、予めそれらのスケジューリングが確定していなければならないという問題があ

る。

本稿では、長時間トランザクションを効率的、効果的に管理し、処理するために三階層トランザクション・モデルを導入する。すなわち、上位層に関連する複合的な処理要求を実行制御する長時間トランザクション層、中間層に個々の独立な処理要求を実行制御するトランザクション層、下位層に原子的な処理単位で実行するアクション層から成る枠組みである。

しかし、階層構造の実行制御では下位の処理が上位の処理に依存し、一般に効率的な実行を実現することが困難である。これを解決するために、それぞれの層に自律動作可能なエージェントを割り当て、層の役割に応じた機能を果たすと共に、層内の実行制御に束縛されることなく、層間に渡って相互に情報を交換できる枠組みを構成する。すなわち、三階層トランザクション・モデルではそれぞれの層における処理要求、処理単位をエージェントによって管理、処理する。長時間トランザクション層には関連した複数のトランザクションの系列を manager と呼ぶエージェントで管理し、トランザクション層での実行を制御する。トランザクション層には個々の独立なトランザクションを、そのトランザクションに割り当てられた executor と呼ぶエージェントが管理、制御し、アクション層での実行を制御する。そして、アクション層には最小の処理単位を実際に実行する actor と呼ぶエージェントが割り当てられ、並行的に個々の処理単位を実行する（図1）。

manager は関連する複合的な処理要求を実行制御するために個々のトランザクションの関係を表すワークフローを管理している。従って、executor はワークフローで表された個々のトランザクションを実行する。一方、executor は独立なトランザクションを actor によって実行される処理単位（タスク）の集合として管理し、このタスク間の実行関係を表すタスクフローを保持し、実行制御する。

\*Transaction Control Using Task Flow on Long Lived Transaction

†Koichiro SUZUKI and Toyohide WATANABE

‡Department of Information Engineering, Graduate School of Engineering, Nagoya University

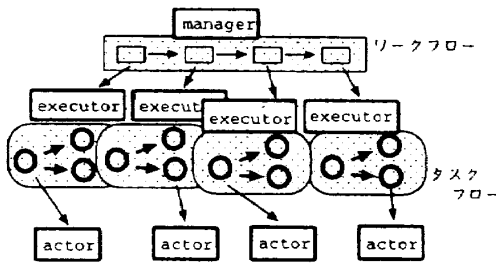


図 1: 三階層トランザクション・モデル

### 3 executor によるトランザクションの実行制御

#### 3.1 タスクフロー

タスクフローは executor が実行制御するトランザクション内に存在するタスクの関係を有向グラフの形式で表現したものであり、ノードとエッジはそれぞれタスクとタスク間の関係を表している。executor はトランザクションの処理を開始するとき、manager からそのトランザクションについての情報を受けとり、その情報をもとにタスクフローを構成し、構成したタスクフローをもとにトランザクションを実行制御する(図 2)。

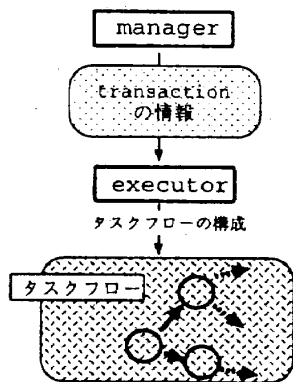


図 2: トランザクションの開始のイメージ

#### 3.2 タスク間の関係

manager が管理するワークフローが逐次的な処理の遷移を表すのに対し、トランザクションの処理内容に対応するタスクフローは並行な実行を表し、処理関係を制御する。従って、タスクフローでは個々の並行可能なタスク間の関係を表すことが必要となる。タスク間に存在する関係を以下に示す。

- 選択実行 複数の並行実行可能なタスクから一つを実行可能とする。
- 優先選択 優先度がついた選択。つまり executor はタスクの実行やどのタスクの結果を生かすかということを優先度によって決定する。
- 制御依存 タスク A がタスク B を制御する。すなわちタスク A の処理が成功するとタスク B の処理が開始される。
- 強制実行 あるタスクの実行結果が不十分な結果に終わった場合、同じタスクを強制的に実行。
- 強制終了 選択、優先選択、強制実行等で不要になったタスクを強制終了、またはその結果の取消。
- 待機(条件) ある条件が満たされると実行される。
- 排他 タスクが競合することを表す。
- 代替 このタスクが別のあるタスクの代替であることを示す。

これらは並行的に実行可能なタスクをタスク相互間の性質に基づいて、効果的な処理を遂行するためにタスクフローの制御エッジとして導入される。

### 4 おわりに

本稿では、三階層トランザクション・モデルの下に相互に関連する複数の処理を効果的に実行制御するための枠組みであるタスクフローについて報告した。

### 参考文献

- [1] J.E. Moss: "Nested Transactions, an Approach to Reliable Distributed Computing", The MIT Press (1985).
- [2] H. Garcia-Molina and K. Salem: "Sagas", *Proceedings of the ACM Conference on Management of Data*, pp. 249-259 (1987).