

エージェントの概念に基づいた長時間トランザクション管理*

4R-7

浅野 正靖† 牛尼 剛聡† 渡邊 豊英†

名古屋大学大学院 工学研究科 情報工学専攻†

1 はじめに

従来、データベースは単一の処理要求を効率的に実行することが主な利用法であった。しかし、データベースに対する処理要求が複雑になるにつれ、また適用業務が拡大するにつれ、複数の関連し合う処理要求を統合的に扱う必要性が高まっている。このような処理要求を長時間トランザクション(long-lived transaction)と言うが、長時間トランザクションを効果的に実行可能とする枠組みを本稿では提案する。我々のアプローチは関連する処理要求を長時間トランザクション層、トランザクション層、アクション層の三階層モデルで実行制御し、段階的に細分化されて実行される処理要求をエージェント指向概念の基に実行する。このような枠組みにより、階層モデルを用いた長時間トランザクション処理に対する柔軟制御の問題を効果的、効率的な実行制御法を実現可能としている。

2 トランザクション・モデル

長時間トランザクションは複数の関連した処理手続きから構成された処理要求であり、それらの処理手続きが相互に関連して実行されるために一般に長い処理時間を要する。このような性質を有する長時間トランザクションを、長時間トランザクション層、トランザクション層、アクション層という三階層で実行制御する。図1に、この三階層トランザクション・モデルを概念的に示す。

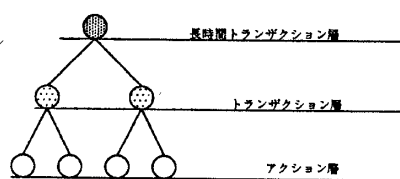


図1: 三階層トランザクション・モデル

長時間トランザクション層はユーザからの処理要求を実行・管理する。そのために、ユーザの処理要求を構成する個々の処理手続きをノードとするワークフローを生成し、処理手続き間の実行状態を制御する。トランザク

ション層はワークフローで表された一つ一つの処理手続きを実行・管理する。トランザクション層で実行・管理される処理手続きは、さらに原始的処理操作によって実行され、この実行・管理がアクション層で制御される。個々の処理手続きを構成する原始的操作をアクションと呼ぶ。アクションをノードとしてアクション間の実行制御関係を表したタスクフローをトランザクション層が生成し、アクション間の同期、並行性などを制御する。アクションは実際にデータベース・アクセス、計算処理などの実行主体である。

例えば、旅行代理店の旅行スケジュール作成業務を考える。あるユーザの旅行スケジュール作成という処理要求が長時間トランザクションであり、旅行計画の立案、交通手段や宿泊施設の予約、代金の支払、チケットの発行、計画変更などの個々の処理手続きが、相互に関連するが、それぞれ独立なトランザクションである。そして、アクションは一つ一つの行程における飛行機、列車、タクシーなどの予約、宿泊施設の予約に対応する。ここで、トランザクションは一般に個々の処理手続きが状態変移的な逐次処理として扱え、アクションは複数の処理操作が同時に実行可能な並列処理として扱える。従って、長時間トランザクション層で管理されるトランザクションの実行管理がワークフローで、またトランザクション層で管理されるアクションの実行管理がタスクフローとして表現可能である。

このような階層構成で、長時間トランザクション処理を実現する際の問題は、階層制御のために下位層の処理が完了しないと、上位層の処理が動作できないことであった。すなわち、下位層の処理が実行不可能になったとき、何らかの手段が必要で、実行制御の管理が階層モデルでは十分に達成できなかった。本三階層トランザクション・モデルでは、これを回避するためにエージェント指向パラダイムを導入する。それぞれの層における処理プロセスを自律的なエージェントで扱うことにより、必要に応じてエージェントが直接他のエージェントと交信し、実行制御の柔軟な実現を図る。すなわち、我々の三階層トランザクション・モデルはこれらエージェントの役割、能力を規定するための概念であり、分散協調を指向するマルチエージェント・モデルに集中制御的考えを導入し、トランザクション間の柔軟な実行制御と、効率的な実行性能を達成することを目標としている。

*Long-lived Transaction Management Based on Agent-oriented Concepts

†Masayasu ASANO, Taketoshi USHIAMA, and Toyohide WATANABE

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

3 エージェント機構

本三階層トランザクション・モデルでは、それぞれの層に固有の役割を有したエージェントを配置して下位層のエージェントを管理すると同時に、柔軟なトランザクション処理の実行制御を達成する。そのために、共有メモリを用意する。この共有メモリは総てのエージェントが参照可能な作業域として位置付けられ、エージェントが階層間の役割に拘束されることなく、実行状況を必要に応じて直接参照し、個々のエージェントが状況に応じた行動を実現可能とする。メッセージパッシングでは、状況に応じた行動をエージェントがとれない場合があるためである。例えば、アクション層のあるエージェントが実行遅延、または実行不可能状態にあるとき、トランザクション層のエージェントはメッセージを送付しても、そのエージェントは待機にあり、メッセージ自身が保留されてしまう。

我々は、2で導入した処理の種類に基づいて、マネージャ、エグゼキュータ、アクタという三種類のエージェントを導入する。三種類のエージェントはそれぞれ異なる特徴を持つ。以下にそれぞれのエージェントの概要を示す。

マネージャ

マネージャはユーザの要求に基づいて長時間トランザクションを実行する。マネージャは長時間トランザクションを実行するために必要なトランザクションをワークフローによって実行制御する。ワークフローは図2のように示される。ノードは仕事を表し、一つの仕事は

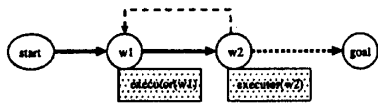


図 2: ワークフロー

一つのトランザクションに対応する。すなわち、一つの仕事に対して、エグゼキュータが一意に対応づけられている。エグゼキュータに委託した処理が成功したときには、太い実線の有向辺で示されている仕事の遷移が起こる。例えば、旅行手続きにおいて、旅行計画という仕事が成功した後、次の仕事である予約の仕事に遷移する。また仕事が失敗した場合は、細い破線の有向辺で示される仕事に遷移する。例えば、予約が失敗した場合には、再度旅行計画が実行される。

エグゼキュータ

一つのエグゼキュータは一つのトランザクションを実行する。エグゼキュータはマネージャからの委託を受けることにより実行を開始する。エグゼキュータは委託に基づいてタスクフローを生成する。エグゼキュータはトランザクションを実行するのに必要なアクションをタス

クフローによって実行制御する。実行制御可能な処理の代表例を以下に示す。

- 複数のアクションの逐次的な実行
- 複数のアクションの並行的な実行
- 複数のアクションの代替的な実行

タスクフローは図3のようにノードと有向辺によって示される。ノードはタスクを表し、有向辺はタスクの遷移を示す。一つのタスクを実現するために一つ以上のアクションが必要である。例えば、図3では、タスク

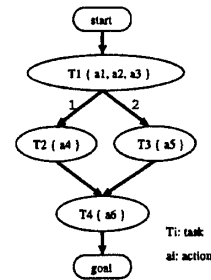


図 3: タスクフロー

T_1 は三つのアクション a_1 , a_2 , a_3 を有する。これらは並行的に実行される。タスクの遷移はすべてのアクションが成功することによって生じる。図3の例では、 T_1 の実行が成功した後、 T_2 または T_3 に遷移する。複数の遷移先があるのは、マネージャから委託された仕事には複数の実現方法が考えられるからである。図3では、 $T_1 \rightarrow T_2 \rightarrow T_4$ または $T_1 \rightarrow T_3 \rightarrow T_4$ の実現方法がある。複数の遷移先の選択は有向辺に指定された優先度に基づいて行なわれる。図3の例では、 T_1 の実行が成功した後に優先度に基づき T_2 に遷移し、アクション a_4 を実行する。ここで a_4 の実行が失敗すれば、 T_2 も実行が失敗することになる。このとき次の優先度を持つ T_3 を実行する。 T_3 は T_2 の代替であると考えることができる。このようにして、start のノードから goal のノードまで遷移したとき、マネージャから委託された仕事が成功したことになり、結果を委託元のエグゼキュータに返答する。

アクタ

一つのアクタは一つのアクションを実行する。アクタはエグゼキュータから委託されたアクションを処理し、それに対する結果を返答する。アクションはプリミティブな操作であり、データベースに対する検索、更新などの単一的な処理である。

4 おわりに

本稿では、長時間トランザクションの効果的、効率的な実行を可能とするために、ユーザの複数に関連し合う処理要求を三階層上の処理として表し、それぞれの処理にエージェントを割り当てる実行制御法を提案した。