

ポリシー制御に基づく Web サービスの対話型トランザクションアーキテクチャの提案と評価

大谷 洋子[†]青山 幹雄[‡]南山大学 大学院 数理情報研究科[†]南山大学 数理情報学部 情報通信学科[‡]

1. はじめに : Web サービストランザクション処理とその問題点

Web サービス上の複数のトランザクション処理を実現するために WS-T(WS-Transaction)が策定されている[2]. ACID 特性を基に, 複数のサービスを一つのビジネスタスクにまとめるスコープ機能, 処理を元の状態へ戻す補償機能を実現する.

しかしビジネスプロセスを連携するトランザクション処理では, 処理全体の完了より, 部分的処理の迅速な完了の方が望ましい場合がある.

また, 長期リソースロックを避け, 個々のサービスで処理を確定するため, 全体の処理結果と一致しない場合があり, 独立性を保証していない.

この問題を解決する対話型 Web サービストランザクションアーキテクチャを提案し, 効果を示す.

2. トランザクションモデルへのアプローチ

2.1. 原子性の緩和と独立性の保証

(1) 原子性の緩和

部分的な処理の完了を可能にするために, 必ず成功すべきサービスの範囲を親スコープ, 部分的に成功すればよいサービス群を子スコープに分離する. 原子性を保証する範囲を親スコープに限定することで処理の柔軟性が向上する.

(2) 独立性の保証

処理の進行に合わせ, 個々の処理結果と全体の結果を区別して公開する. 処理状態と外部から見た状態が一致するので, 独立性が保証できる.

2.2. 柔軟性の向上

上記の解決策を生かし, 処理を確定する期限や, ビジネスプロセス間の優先度を設定するシステムを提案する. 優先度の高いビジネスプロセスから順に予約処理を行い, 期限に応じて対応を変えることで柔軟なトランザクション処理を可能にする.

3. 対話型アーキテクチャの提案

対話とは, メッセージ交換によるビジネスプロセス間の協調を行う方法である. これにより優先度や期限の交渉が行える. 状態に合ったポリシーを呼び出すことで状況に柔軟に対応できる対話モデル[3]を利用する. 対話モデルでは, メッセージ

を送受信するための「メッセージング」と, メッセージの形式定義や構文解析を行う「対話支援」の2つで協調を実現する.

3.1. 対話型トランザクションアーキテクチャ

トランザクション全体の構成は WS-T, 対話機能は後述の WS-Policy によるポリシー制御で実現する(図 1). Coordinator 内に対話機能を付加することで, サービス間で対話を行い, 期限や優先度の交渉やトランザクションが失敗した時の補償処理を決定する. サービス毎の対話と, トランザクション全体の制御を独立に行うことで, 全体の原子性を保証しながら柔軟な処理を可能にする.

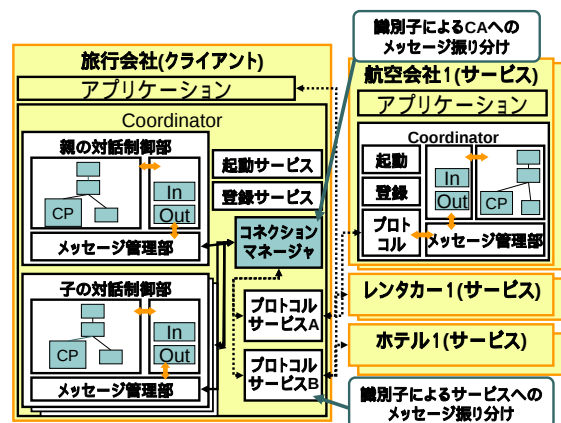


図 1 対話型トランザクションアーキテクチャ

3.2. 制御部の分離

親スコープを親の対話制御部が管理し, 子スコープを子の対話制御部が管理する. 親は子と対話を行い全体の処理を制御する. 子スコープ内のタスクは全体の状況とは独立に処理を完了し, 長時間リソースをロックするリスクを回避できる.

3.3. トランザクションの適応型制御

対話型トランザクションでは, 対話で調整中の中間結果と最終結果を区別して公開する. そのためトランザクションの進行状況と他のシステムから見た進行状況が一致し, 独立性が保たれる. 中間結果に対しキャンセル待ちの交渉を有効にし, 最終結果が異なる場合の処理をより柔軟にする.

3.4. WS-Policy による対話の制御

3.4.1. WS-Policy

サービスの機能, 利用方法, プロバイダとリクエスト間の通信の構文とセマンティクスを XML で定義する. 範囲指定や優先度の設定ができるため, 本提案ではこれを「対話支援」に用いる.

Policy-based Conversation Architecture for Web Services Transaction Processing and its Evaluation

[†]Yoko Otani Graduate School of Mathematical Sciences and Information Engineering, Nanzan University

[‡]Mikio Aoyama Graduate School of Mathematical Sciences and Information Engineering, Nanzan University

3.4.2. 対話ポリシーの設計

全体の処理の流れと対話を独立に管理するため、対話ポリシーは親の対話制御部と子の対話制御部で別々に設計する。対話は双方が同じポリシーを呼び出し、定義したメッセージ形式に沿って情報を送受信することで進められる。トランザクション処理のフローは WS-T のプロトコルを基礎とし、状態毎の対話を可能にする。

3.4.3. 親の対話制御部の対話ポリシー

親の対話制御部の対話ポリシーは、子と対話するためのポリシーと、トランザクション全体を管理するためのポリシーの二種類を作成する。複数の子の状態を把握し、子スコープが全て中間結果を確定した時点で最終結果の確定を通知する。原子性を保証する範囲を満たした処理を確定できる。

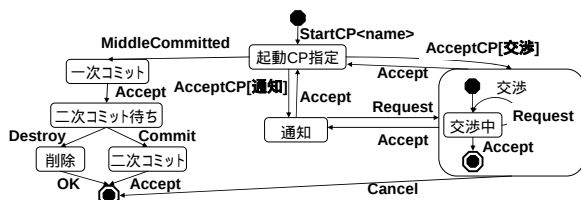


図 2 親の対話制御部の状態遷移

3.4.4. 子の対話制御部の対話ポリシー

子の対話制御部の対話ポリシーは、サービスと対話するためのポリシーと親と対話するためのポリシーの二種類を作成する。全体の決定までは親の対話制御部や外部のサービスと対話による期限や優先度の交渉を呼び出せるようにする。

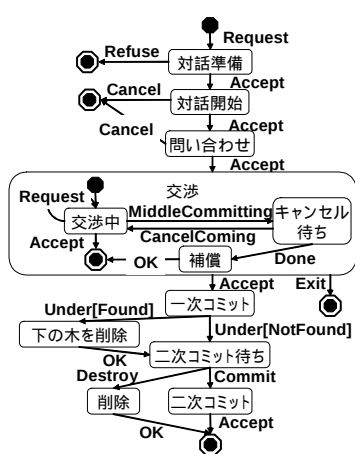


図 3 子の対話制御部の状態遷移

4. 対話型トランザクションの評価

WS-T と対話型トランザクションを比較し、提案するアーキテクチャの有用性を評価する。評価には、図 1 に示す旅行代理店予約システムを用い、表 1 のシナリオに沿って行う。

表 1 旅行代理店予約シナリオ

スコープ間の優先度は、航空会社 > ホテル > レンタカーである。スコープ内の優先度は、航空会社は 1>2, ホテルは A>B, レンタカーは > とする。予約処理を確定させる期限は 3 日とする。予約処理を実行した時点で、航空会社 1 は空きがなかったとする。ホテル A は一次コミット状態を 1 日しか保持できない。

このシナリオに沿った場合 WS-T の処理結果は、航空会社 2, ホテル A, レンタカー となる。それに対し対話型トランザクションでは表 2 に示すように条件に応じて 4 通りの処理結果が得られる。

表 2 対話型トランザクションの処理結果

	キャンセル待ち	期限延長	処理結果
(1)	OK	OK	航空会社1, ホテルA, レンタカー
(2)	OK	NG	航空会社1, ホテルB, レンタカー
(3)	NG	OK	航空会社2, ホテルA, レンタカー
(4)	NG	NG	航空会社2, ホテルB, レンタカー

例として表 2 の(2)の処理の流れを図 4 に示す。キャンセル待ちにより WS-T では取れなかった航空会社 1 の予約に成功している。これはホテルの期限より航空会社を優先した結果、3 日以内に空きが生じたため、状況に対応した成果と言える。

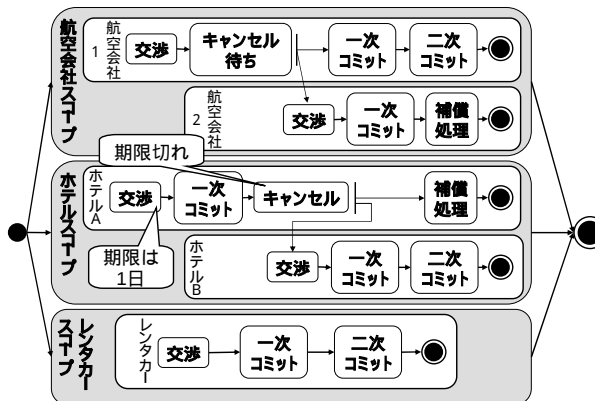


図 4 対話型トランザクションの状態遷移

対話型トランザクションでは、処理の優先順位に基づき順次実行し、予約できない場合は優先順位の低い予約処理に移る。状況に応じて原子性の緩和の範囲を変更できる柔軟な処理が実現できた。中間結果と最終結果を明確に区別でき、独立性を保証した。キャンセル待ちなどの処理状態を増やし、対話によって交渉が行えるので、参加者の要求も組み込みながら、状況に応じた様々な処理結果を得ることができる。

5. まとめと今後の課題

Web サービスに対応したトランザクション仕様である WS-T と WS-Policy を組み合わせた対話型トランザクションを提案した。今後はより柔軟に対話を行える対話ポリシーを検討する。

参考文献

- [1] 大谷 洋子他, Web サービスの対話型トランザクションアーキテクチャの提案と評価, 情報処理学会ソフトウェア工学研究会, 2005-SE-147, Mar. 2005, pp.65-72.
- [2] F. Cabrera, et al., WS-Transaction, 2004, <http://www-128.ibm.com/developerworks/library/specification/ws-tx/>.
- [3] J. E. Hanson, et al., Conversation Support, 2003, <http://www.research.ibm.com/convsupport/>.