

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 アーキテクチャに関する研究

M2005MM024 大谷 洋子

指導教員 青山 幹雄

1. はじめに

Web 上でビジネスを連携する技術として、Web サービスを用いたビジネスプロセス連携が利用されている。ビジネスプロセス連携において、トランザクション処理は不可欠な技術である。しかし、単一システム上の集中処理を行う既存のトランザクション処理技術[2]は、Web 上での長期疎結合型の分散処理に適さない。そこで Web サービスのトランザクション処理仕様が提案されている。これらの仕様では、ACID 特性の原子性と独立性を緩和し、長期疎結合の処理に対応する。しかしビジネスに適用するには柔軟性が低く、動的なサービス選択に対応する処理ができないため、新たな処理方法として対話型トランザクション処理を提案する。

2. 研究の目的

本研究では、ビジネスに対応した Web サービスのトランザクション処理アーキテクチャと処理方法を提案する。既存の Web サービストランザクション仕様では、トランザクションに参加する全てのサービスと並列に個別の処理を行い、処理の完了時に全体の結果に含めるサービスを選択する。この処理方法は長期処理におけるリソースロックという問題点を解決しているが、個別の処理を取り消すため、補償処理が必要になる。キャンセルの多い処理はビジネスにとってリスクが高い。そこで、必要な機能を満たすサービスの集合を1つのサービスとすることで部分的な処理を行う、成功の期待値が高い処理方法を提案する。

また、サービスの選択において期限や優先度を調整し、ビジネスの状況に応じて処理結果を柔軟に変更できる仕組みが望ましい。トランザクション内で期限や優先度の交換を行うため、対話モデルを用い、状態遷移によってメッセージを選択し、交換する仕組みを提案する。

これを対話型トランザクション処理と定義し、アーキテクチャと処理方法を提案する。

3. Web サービストランザクション処理仕様

Web サービストランザクション処理仕様として WS-TX (Web Services Transaction)[1], Business Transaction Protocol, Web Services Composite Application Framework などが提案

されている。本研究では、広く支持されている WS-TX を基盤となるトランザクションの仕様とする。WS-TX は、WS-C (WS-Coordination), WS-AT (WS-Atomic Transaction), WS-BA (WS-Business Activity)で構成される。分散したアプリケーションごとに Coordinator を持ち、調整プロトコルに基づきメッセージを交換しながら分散アプリケーションの振る舞いを調整する。調整プロトコルは、WS-AT が ACID 特性を保証し、WS-BA がビジネスプロセス連携への対応を定義する。個々のサービスとの処理を完了し、全体の処理結果に含めないサービスは補償処理を行い処理を取り消す。WS-TX は複数のサービスを処理単位にまとめ、一つのビジネスタスクとして処理する「スコープ」機能を持つ。しかし、WS-TX では、複数の Web サービスの状態遷移の具体的な制御方法やスコープの決定方法は定義していない。

4. Web サービストランザクションへの課題

4.1. Web サービストランザクションの問題点

(1) 原子性の部分的緩和

WS-TX のスコープ機能は原子性を緩和し、サービスとの部分的な処理を行うための機能である。しかし処理範囲の指定方法などは定義されておらず、サービスの組み合わせが機能を満たすことを保証してはいない。トランザクション全体の一貫性、機能の原子性を保証しながらサービスの選択を行える、原子性の部分的な緩和方法が必要である。

(2) 独立性の緩和に対するリスク

WS-TX の処理はトランザクションに参加した全てのサービスと並行処理を行うため、全体の処理を確定するまでに複数のサービスと処理を完了することになる。これは契約の重複と補償処理による課金のリスクをもたらす。処理の効率を上げ、契約の重複を減らす処理を行う必要がある。

(3) 処理中の状況の変化に対応する処理

ビジネスプロセス連携ではビジネスの変化に対応し、実行時の状況に対して最適なサービスを選択する処理が求められる。WS-TX ではサービスの選択方法を定めておらず、処理中の変化に対応できない。処理期限やサービスの利用状況など、サービス選択を行うのに必要な条件についてサービスと交渉し、トランザクション処理内で条件に合うサービスを選択する処理が求められる。

4.2. ビジネスプロセス連携に対応する処理方法の提案

4.2.1. スコープ制御による部分的な処理

機能の原子性を保証したサービス選択を行うには、サービスが満たす機能を把握し捨捨選択する、部分的に原子性を緩和できる処理が必要である。図 1 に示すように、WS-TX のスコープ機能を用いて、トランザクション処理を完了と判断するために最低限満たすべき機能集合と、各機能を満たすサービス集合を階層的に分割し管理する。機能集合を「親スコープ」と定義し、同じ機能を満たすサービス群を「子スコープ」と定義する。各子スコープ内で一つずつサービスと処理が確定すると、親スコープの原子性を満たす。Coordinator 内で各スコープに 1 つずつ管理機能を付加することでスコープの階層的な管理が可能になる。

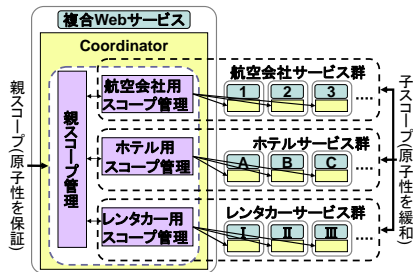


図 1 スコープ分割による階層的な管理

4.2.2. 対話制御を用いた柔軟性の向上

トランザクション処理の中で状況に応じたサービス選択を行うには、どのサービスが処理中の条件に最適か判断する必要がある。また期限内に必要なサービスを取得できる仕組みが求められる。WS-TX の調整プロトコルではメッセージや状態遷移の様子が固定されており、サービス選択機能はない。処理中のサービス選択や変化する条件に対応するために、ビジネスプロセス連携の手法を取り入れる。

ビジネスプロセス連携の手法として、オーケストレーションとコレオグラフィが挙げられる。オーケストレーションは複数のサービスの連携をフロー記述によって制御する。コレオグラフィは複数のサービスを個々の独立したメッセージ交換によって制御する。本研究では、個々のサービスの状況の変化に対応する処理を実現するため、コレオグラフィの手法の一つである対話制御を用いる。

図 2 に示す対話モデルでは、対話制御部が対話の流れを定義する対話ポリシー(CP)を用いて、状態に合ったメッセージを交換し相互運用性を実現する。この機能は WS-TX の調整プロトコルを用いたメッセージ交換手法に類似しているため、Coordinator のメッセージ交換機能を対話制御部、調整プロトコルを対話ポリシーに変更することで対話処理を実現する。トランザクション処理の流れを保持し、対話を行う対話ポリシーを作成する。Coordinator が対話制御部を用いて対話を行うトランザクションを管理する。

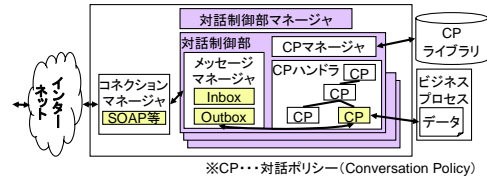


図 2 対話モデルの構成

5. 対話型トランザクション処理のモデル

5.1. 対話型トランザクションのアーキテクチャ

アーキテクチャは WS-TX の Coordinator を基礎として構成する。図 3 に示すように、各サービスが Coordinator を持ち、Coordinator 同士のメッセージ通信によりトランザクション処理を進める。トランザクション起動時の参加サービスとの準備フェーズは起動、登録サービスが行い、コンテキストによってコネクションマネージャのエンドポイント情報を伝達する。コネクションマネージャがメッセージマネージャにメッセージを振り分け、対話制御部同士がメッセージ交換を行うことでトランザクション処理を進める。

個々のサービスとの対話とトランザクション全体の一貫性を両立するために、Coordinator 内にサービスと対話を行う子の対話制御部と、全体を管理する親の対話制御部を用意する。処理の進行中に状況が変化した場合対話による調整を行う。このアーキテクチャを用いることで、参加者との間で期限や優先度の情報を基に対話を行い、サービス連携を実現する。処理の進行による状況の変化に応じて対話による調整を行い、ビジネスプロセスに対応する。

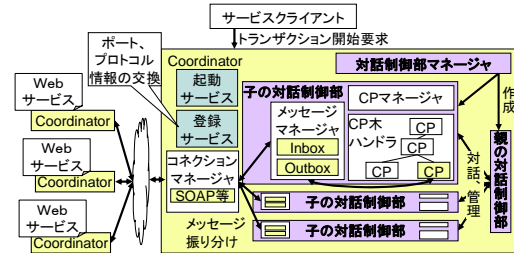


図 3 対話型トランザクションアーキテクチャ

5.2. 対話型トランザクション処理の流れ

トランザクションにサービスを登録し、処理を開始する流れを図 4 に示す。複合 Web サービスを利用するユーザはクライアントアプリケーションがあらかじめ提示する指標に沿って提供サービスの組み合わせや優先度を決定する。それを元にアプリケーションがトランザクションに含めるサービスを選択、順位付けし、Coordinator を通じて登録する。アプリケーションが処理を起動すると、サービスと子、親と子の間で対話による期限の交渉を行いながら、期限内にトランザクション処理を完了させ、結果を返す。図 4 のトランザクション処理は以下の順に処理を行う。

- (1) アプリケーションがサービスと優先度の指標を提示
- (2) ユーザがサービスを選択(優先度, 期限の指定)
- (3) 登録するサービスをスコープごとに優先度により順位付け
- (4) スコープごとに子の対話制御部を作成
- (5) サービスがコーディネータを用い, 対象のスコープのトランザクションに登録
- (6) 優先順に個別のトランザクションを開始
- (7) サービスと子, 親と子の対話による期限の交渉
- (8) 個々のサービスとのコミット
- (9) 全体のコミット/アボートによる処理の確定
- (10) アプリケーションに結果を返す
- (11) ユーザに結果を返す

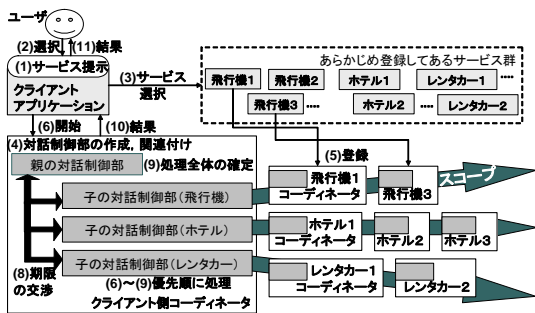


図 4 対話型トランザクション処理の流れ

5.3. 対話型トランザクションの制御方法

対話型トランザクション処理はメッセージ交換機能を用いて期限を交渉できる。補償処理の代わりに期限を用いた処理方法を提案する。個別のコミット結果を保持する期限を交渉し、期限内に処理が確定しない場合はサービス側がコミットを取り消す。自動的にトランザクション処理を制御することで、無期限に処理の確定を待つサービス側のリスクを回避する。トランザクション管理側はサービスを無期限に拘束できない代わりに、期限内であれば補償を行わずに取り消し処理ができる。処理全体の期限はユーザの示した期限を元に、サービスの個別のコミット保持期限に対し対話による交渉を行うことで変更できる。

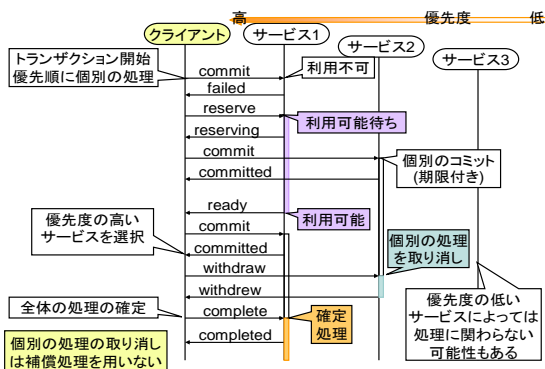


図 5 対話型トランザクション処理フロー

図 5は対話型トランザクションの処理フローの例である。優先度の高い順にサービスと1対1の処理を行い、そのサービスとの処理が利用できなかった場合は次の候補との処理に移る。その際、個別の処理に対する予約を行い、全体の処理が確定するまでに利用可能になった場合は優先度の高いサービスとの処理を優先する。この処理により、直列処理を行いながら優先度の高いサービスとのビジネスチャンスを保ち、ユーザの満足度の高い結果を得る。

6. 対話ポリシーの設計

6.1. 親の対話制御部と子の対話制御部の対話ポリシー

対話モデルではサービス同士の対等な対話を扱うため、双方が同じ対話ポリシーを呼び出す。しかし対話型トランザクションでは親と子で役割が分担されているので、図 6に示すように、それぞれの役割に合った対話ポリシーを設計する。対話ポリシーはWS-TXのBACCプロトコル(Business Agreement with Coordinator Completion protocol)のフローを基礎とし、状態ごとに対話を行うよう機能を追加する。

子の対話制御部は個々のサービスの状況を把握し、個別のトランザクション処理を実現する。トランザクションの各状態で、サービスと対話するポリシーと、親と条件を交渉するポリシーを呼び出す対話ポリシーを作成する。子の対話制御部は各サービスとの処理を木構造で管理する。複数のサービスとの処理を一つの本で管理することで、スコープ内のサービス選択の一貫性を保つ。

親の対話制御部は子の状況を把握しながら全体のトランザクションを管理する。子の進行を把握するトランザクション管理ポリシーと、子と条件の交渉を行うポリシーを作成する。処理の一貫性を保ってトランザクション全体を管理するため、全ての子と同じポリシーを用いて処理を進める。全ての子が個別の処理を完了したら、処理を確定する。

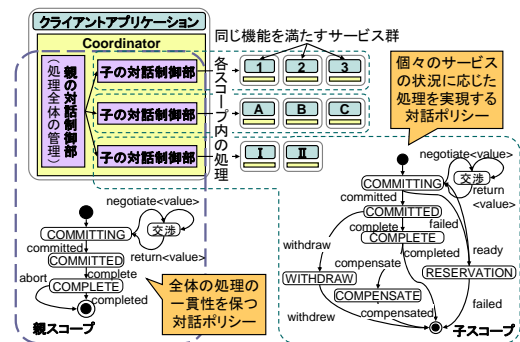


図 6 親と子の対話ポリシーの分離

7. シナリオを用いた評価

提案した対話型トランザクション処理がビジネスプロセス連携において有効かどうか、表 1の旅行代理店の予約シ

ナリオを用いて評価する。全体の処理期限は三日とする。

表 1 旅行代理店の予約シナリオ

スコープ	航空会社			ホテル			レンタカー	
サービス名	航空1	航空2	航空3	ホテルA	ホテルB	ホテルC	レンタカー I	レンタカー II
優先度	10	8	7	7	6	4	5	5
中間結果保持期限	7日	5日	5日	1日	3日	3日	7日	7日
サービスの初期状態	満席	○	○	○	○	○	○	○

上の条件に沿った場合、WS-TX の処理結果は、航空会社 2、ホテル A、レンタカー I となる。それに対し、対話型トランザクション処理では状況に応じて複数の処理結果が得られる。例として航空会社 1 に空席が生じた場合の処理の流れを図 7 に示す。優先度によってホテル A より航空会社 1 を優先して確保する。利用可能待ちを用い、状況に対応した結果、航空会社 1、ホテル B、レンタカー I を得る。

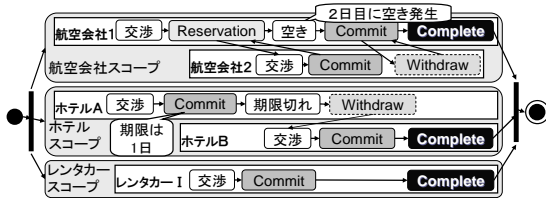


図 7 対話型トランザクションの予約フロー

表 2 WS-TX と対話型トランザクションの処理結果

スコープ	航空会社			ホテル			レンタカー		優先度合計	
サービス名	航空1	航空2	航空3	ホテルA	ホテルB	ホテルC	レンタカー I	レンタカー II		
優先度	10	8	7	7	6	4	5	5		
WS-TX	x	○	x	○	x	x	○	x	20	
対話型トランザクション	パターン1	○	x	x	○	x	x	○	x	22
	パターン2	x	○	x	○	x	x	○	x	20
	パターン3	○	x	x	x	○	x	○	x	21
	パターン4	x	○	x	x	○	x	○	x	19

表 2 に WS-TX と対話型トランザクションの処理結果を示す。優先度の合計は WS-TX の 20 に対し、対話型トランザクションを用いた場合 4 パターンの処理結果が得られ、優先度は 19~22 となった。3 パターンで WS-TX 以上の優先度となり、対話による条件の交渉により、処理の中でサービス選択を行いながらクライアントの満足度の高い結果を得ることができた。このように対話型トランザクション処理はビジネスプロセス連携の状況に対応する処理を実現できる。

8. 関連研究

文献[6]では、複合 Web サービスでのトランザクション処理の保証を重視し、問題点として登録期限やプロトコルの差異を挙げている。また、Web サービスのモデル駆動開発でトランザクション機能を組み込む手法として、レイヤを分離した設計方法を提案している。しかし具体的な問題点の解決策、期限や優先度の交換方法には触れていない。

文献[4]では複合 Web サービスにおけるトランザクション処理の一貫性を保つ体系的なデザイン手法を提案してい

る。トランザクションの終了状態の組み合わせに対し必要な処理機能を判定する。動的連携時にこの判定手法を用いてサービスを連携することで、トランザクション処理の一貫性を保証する。この手法で判定するのは処理機能の一貫性であり、実際の処理におけるプロトコルの違いや、連携したサービスに対する一貫性の保証は含まれていない。

9. 今後の課題

今回提案した対話ポリシーは、Web サービスの動的連携とサービス選択に対応したものである。しかし、現在の Web サービスにおいては、全てのサービスが同じプロトコルを用いる可能性は少ない。他のプロトコルにも対応するポリシーを用意する必要がある。また、連携に合った対話ポリシーを開発者が作れるように、トランザクションの一貫性を保つ対話ポリシーの作成方法を体系化する必要がある。

10. まとめ

ビジネスに対応するトランザクション処理を実現するため、Web サービストランザクション仕様の WS-TX と、ビジネスプロセスの連携方法である対話モデルを組み合わせ、対話型トランザクションアーキテクチャを提案した。これを用いることで、参加サービスがクライアントと条件に関する対話を行うことができ、変化するビジネスの状況に対応できる。また、優先度や期限を考慮することで、クライアントにとって望ましいサービスを処理中に選択でき、効率の良い処理が実現できる。従って、クライアントにとってもプロバイダにとっても満足度の高い処理を提供することが可能となる。

参考文献

- [1] F. Cabrera, et al., WS-Transaction, 2004, <http://www-106.ibm.com/developerworks/library/specification/ws-tx/>.
- [2] J. Gray and A. Reuter, Transaction Processing: Concepts and Techniques, Morgan Kaufmann, 1993.
- [3] J. E. Hanson, et al., Conversation Support, 2003, <http://www.research.ibm.com/convsupport/>.
- [4] F. Montagut, et al., Augmenting Web Services Composition with Transactional Requirements, ICWS 2006, Sep. 2006, pp. 91-99.
- [5] 大谷 洋子他, 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, SES2006, Oct. 2006, pp. 145-148.
- [6] B. A. Schmit, Model-Driven Development of Web Service Transactions, <http://www.infosys.tuwien.ac.at/Staff/sd/papers/btw05.pdf>