

Web サービスのトランザクション管理

大澤 貴彦[†] 高橋 篤史[†] 志村 修一[†] 片岡 信弘[†]

東海大学工学部電子工学科

1. はじめに

ビジネスの世界に Web サービスを適用するために、複数のサービスをひとまとまりとする、トランザクション管理に観点をあいた。

Web サービスのトランザクション管理の例として、当研究では、旅行ポータルサイトの予約サービスをトランザクションで実現するシステムを構築し、その有効性を評価した。

2. Web サービスによるトランザクション管理

2.1 Web サービスの概要

Web サービスとは、インターネット上に分散した複数のアプリケーションシステムをシステム同士で動的に連携させる技術である。

Web サービスの仕組みは Fig.1 に示す。

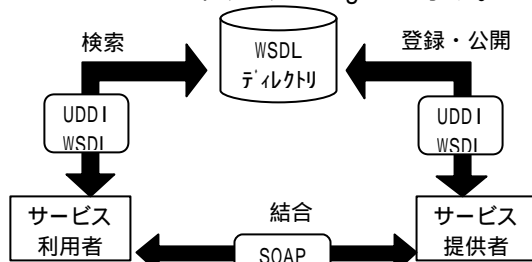


Fig.1 Web サービスアーキテクチャ

提供者がディレクトリにサービスを登録・公開する。利用者はディレクトリからサービスを検索する。発見したサービスと結合する。

2.2 Web サービスの課題

Web サービスが今の社会でさらに普及するには、まだ次のような課題が残っているといえる。

- ・ セキュリティレベルの向上
- ・ トランザクション機能の実現
- ・ 性能とスケーラビリティの向上

etc

当研究では、トランザクション機能の実現に観点をあいて、研究を進めてきた。

3. Web サービスによるトランザクション管理

3.1 概要

Web サービスのトランザクションに関する仕様である "WS-Transaction" を参考にした。この、WS-Transaction には 2 種類の規定がある。

3.2 アトミックトランザクション(AT)

AT とは、all-or-nothing の厳密なトランザクションである、2 フェーズコミットである。

旅行代理店ポータルサイトを例にすると、代理店はまず仮予約をし、すべての予約ができる場合は本予約 (コミット) し、1 つでも予約ができない場合は失敗 (アポルト) となり、実行前の状態に戻す (ロールバック)。

3.3 ビジネスアクティビティモデル(BA)

BA は、トランザクションの参加者の処理がすべて成功すると仮定し、すぐに予約 (コミット) する。その後、他の参加者の処理が失敗した場合は、それまでに成功したすべての参加者の処理をキャンセルし、元の状態に戻す。

3.4 それぞれのモデルの特性

ネットワークを介して複数の Web サービスにまたがるトランザクションの場合、提供側の環境によっては長い時間を要することがある (ロングトランザクション)。

AT は、リソースのロックなどが問題となるため、比較的短時間で処理を終わらす必要があるため、ロングトランザクションには適していないといえる。

それに対し、BA は、リソースを長時間ロックすることがないため、ロングトランザクションに適している。しかし、コミットする処理によっては、ロールバックをするためのキャンセルの処理に費用が発生する可能性がある。また、トランザクションが完了するまでの間には、中間結果が見えたり、誤った状態が伝わってしまったりするなどのデメリットがある。

よって、開発者は、開発するシステムが AT と BA のどちらの仕様が適しているのかを把握し、開発をする必要があるといえる。

Transaction management of Web service

[†]Takahiko Osawa, Atsushi Takahashi
Syuichi Shimura, Nobuhiro Kataoka
Tokai University

4. 構築したシステム

当研究で構築したシステムは、旅行ポータルサイトであり、このサイトは外見的には、航空会社とホテルの検索と予約をするシステムであるかのように見えるが、実際の中身では、Web サービスを利用して構築されている。そして、航空機の予約とホテルの予約がトランザクションとして処理されている。

4.1 従来のシステムとの比較

従来のポータルサイト

航空機とホテルの予約を別々に行っていたので、「航空機は予約できたけどホテルは満室で予約ができなかった」といったようなことが起こる。この場合、その計画は失敗であるのに航空機の予約だけされてしまう。ホテルが予約できないので、この計画が中止になった場合、ユーザは航空機のキャンセルをまた別の処理をとらなくてはならない。

トランザクションを用いたポータルサイト

航空機とホテルの予約をひとまとまりの処理とすれば、仮に、ホテルが満室で航空機には空席がある状況であっても、ホテルの予約ができないので、その計画は失敗であるということで航空機とホテルの両方の予約をしない。つまり、このシステムでは、航空機とホテルの予約が共にできる場合のみ、両方の予約をしてくれるシステムであるので、確実性のある計画を立てられる。

4.2 構築システムのトランザクション処理

現状で AT を実現させるには Web サービスのミドルウェアと TP モニタ製品を組み合わせる必要があり、トランザクション処理のスキルと大量のプログラムの作り込みが発生する問題から、AT がサポートする 2 フェーズコミットによる構築は出来なかった。

当研究で構築したシステムは、AT の考え方を参考にした。このシステムの、トランザクション処理は、まず、利用者が各予約サービスのトランザクションに参加する。次に、利用者は、各サービスの仮予約をするために、航空機やホテルの部屋の種類、予約日、予約者名の情報を提供者に送信する。この情報で予約が可能かの反応を待つ。

各サービスすべての予約が可能であれば、その計画は成功であるので『コミット』であることを各サービス提供者に送信する (Fig.2)。

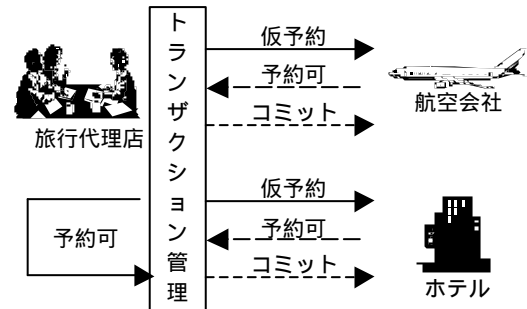


Fig.2 トランザクションの成功時の流れ

一つでも予約ができないものがあれば、この計画は失敗であるので『アボート』であることを各サービス提供者に送信し、提供者側はすべての処理を実行前の状態に戻すために『ロールバック』をする (Fig.3)。

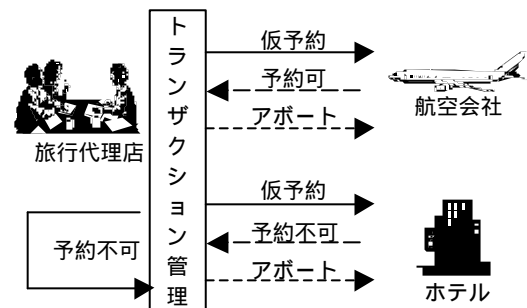


Fig.3 トランザクションの失敗時の流れ

5. おわりに

当研究では、従来の Web サービスにトランザクションが導入されることによって、今までとは違ったビジネスが提供され、そして、新しいビジネスチャンスが生まれる可能性があることを提案し、その有効性について検証した。

今後の予定としては、トランザクションとしての信頼性を向上するとともに、品質の向上をしていきたいと思えます。

参考文献

- [1]Felipe Cabrera 他 『WS-Transaction』
<http://www.microsoft.com/japan/msdn/webservices/spec/ws-transaction.asp>
- [2]本 俊也 『Web サービス構築』