

## Web サービスにおける巡回型トランザクション処理方式の実現

松本 直樹<sup>†</sup> 鈴木 和久<sup>†</sup> 横田 裕介<sup>††</sup> 大久保 英嗣<sup>††</sup><sup>†</sup>立命館大学大学院理工学研究科 <sup>††</sup>立命館大学情報理工学部

## 1 はじめに

ネットワークや端末の技術の進展により、近い将来ユビキタス環境が実現されようとしている。ユビキタス環境では、多種の機器、機能が偏在し、様々なサービスを提供している。この環境下では、利用者の要求は多様なものとなる。現在、我々は、このような利用者の要求に応えるため、複数の Web サービスを複合したサービスを提供するための基盤システムの研究・開発を進めている。本システムでは、利用者の要求を、その内容に応じて集中処理方式と巡回処理方式のいずれかの処理方式によって実行する。本稿では、このうち、複数の Web サービスを巡回し処理を行う巡回型トランザクション処理方式の実現について述べる。

## 2 連携処理方式

ユビキタス環境では、ネットワーク上の資源を利用するという分散環境の側面と、利用者が携帯端末を持ち移動するというモバイル環境の側面がある [1]。現在、我々は、このようなユビキタス環境におけるサービス合成を実現する処理方式について研究している。

## 2.1 集中処理方式

利用者の求めるサービスは、利用者の状況により変化する。しかし、既存のワークフローにもとづく Web サービス連携技術は、サービス要求元の状況変化を考慮しておらず、利用者の状況に応じたサービスの複合は困難である。集中処理方式は、利用者の要求をワークフロー化して利用者の位置や要求内容、サービスの実行結果を集中処理し、それらの情報から呼び出すサービスを切り替える処理方式である。集中処理により、連携システムは、ワークフロー処理途中に、呼び出すサービスをその場で発見し切り替えることが可能になる。

## 2.2 巡回処理方式

現在の WSBPEL [2] に代表されるサービス連携技術は、WSBPEL サーバが集中的にサービス間のフロー制御を行う。巡回処理方式は、サービスプロバイダによる巡回型 SOAP メッセージの転送によって実現される。巡

回型 SOAP メッセージは、複数の巡回先サービスの実行引数やその処理結果、サービスプロバイダ間のデータフローが記述される。巡回処理方式は、それぞれのサービスプロバイダが、フロー制御を行い、サービス間のデータフローを集中的に処理するサーバを必要としない。さらに、SOAP メッセージは、下位プロトコルに HTTP や SMTP などを使用するため、コネクションレスな分散処理を実現できる。

## 3 巡回処理システム

## 3.1 システムの想定環境

巡回処理方式の適用場面として、データベースの巡回検索が考えられる。まず利用者は携帯端末を持ち、無線通信を使用して処理を要求したのち、処理結果を待たず移動する。システムは、利用者の要求を受け取ると、ネットワーク上の複数のサービスプロバイダを巡回してデータを収集する。利用者の端末が電波到達範囲外へ移動し、ネットワークへの接続性が切れた場合、利用者は、別のネットワークへ再接続し、処理結果を受け取ることもできる。

## 3.2 巡回型 SOAP メッセージの記述法

本稿で提案する巡回型 SOAP メッセージの記述例を図 1 に示す。巡回型 SOAP メッセージは、巡回型ヘッダ (RoundHeader) を持ち、SOAP メッセージを識別する一意の識別子 (RoundSoapID)、処理結果の最終送信先アドレス (RootAddress)、処理終了時間 (EndTime) が記述される。

巡回型 SOAP メッセージは、複数のサービスプロバイダ間を巡回するため、各サービスプロバイダに対応した複数のインタフェース要素を持つ。フォーマットには、サービスプロバイダの SOAP Body 内要素の処理方法が記述される。

SOAP Body 内の各要素は、multiRef タグに格納される要素を参照する内容を格納できる。本記処理方式では、サービスプロバイダの処理結果を multiRef タグに格納し、インタフェース要素にそれを参照する内容を記述する。この記述により、サービスプロバイダは、別のサービスプロバイダの処理結果を受け取る。

transmission 要素には、複数のサービスプロバイダのアドレス、処理順番、処理回数が記述される。サービスプロバイダは、メッセージ処理後、transmission 要素により次の送信先を決定する。

Implementation of Transaction Processing Scheme for Web Service

Naoki Matsumoto<sup>†</sup>, Kazuhisa Suzuki<sup>†</sup>, Yusuke Yokota<sup>††</sup>, and Eiji Okubo<sup>††</sup><sup>†</sup>Graduate School of Science and Engineering, Ritsumeikan University<sup>††</sup>College of Science and Engineering, Ritsumeikan University

ConditionalSentence 要素は、分岐処理に関する情報を扱う。分岐条件を定義し、サービスプロバイダは、条件を満たした場合、メッセージの次の送信先を変更する。

```

<SOAP-ENV:Envelope>
<SOAP-ENV:Body>
<r:RoundHeader xmlns:r="round.header.com">
<r:RoundSoapID>12345</r:RoundSoapID>
<r:RootAddress>www.solos.ritsumei.ac.jp/~naoki</r:RootAddress>
<r:EndTime>2006-12-24-10:10</r:EndTime>
</r:RoundHeader>
<spa:SPInterface-a xmlns:spa="sol.spa.com">
<spa:parameter xsi:type="xsd:string">防災クマさん</spa:parameter>
<spa:OrderSP href="#transmission"/>
<spa:ConditionalSentence>
<conditionEQ/condition>
<subject1>防災クマさん</subject1>
<subject2 href="#d0"/>
<to>www.solos.ritsumei.ac.jp/~naoki</to>
</spa:ConditionalSentence>
</spa:SPInterface-a>
<spb:SPInterface-b xmlns:spb="sol.spb.com">
<spb:parameter1>8989</spb:parameter1>
<spb:parameter2>2006-12-17</spb:parameter2>
<spa:OrderSP href="#transmission"/>
</spb:SPInterface-b>
<multiRef id="transmission">
www.001.com.1
www.002.com.1
www.003.com.0
www.004.com.0
www.solos.ritsumei.ac.jp/root
</multiRef>
<multiRef id="www.001.com" soap:root="0" soap:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" xsi:type="xsd:int" xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/">1111</multiRef>
<multiRef id="www.002.com" soap:root="0" soap:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" xsi:type="xsd:int" xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/">2222</multiRef>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

図 1 巡回型 SOAP メッセージの記述例

### 3.3 システム構成

本システムの全体構成を図 2 に示す。本システムは、オペレーティングシステム上のミドルウェアとして実装している。システムは、Java で記述しており、サービスの要求には巡回型 SOAP メッセージを用い、その通信プロトコルには HTTP または SMTP を用いる。ディレクトリサービスは、サービスプロバイダのインタフェース情報、アドレス、提供しているサービスの名前、サービス内容を管理し、利用者からの検索を可能にする。

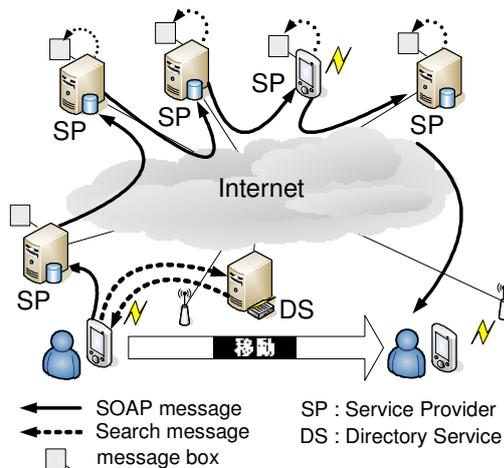


図 2 巡回処理方式

利用者は、まず、ディレクトリサービスを利用してサービスプロバイダを検索し発見する。利用者は、サービスプロバイダの実行順番、サービスの実行引数、処理終了

時間を決定する。クライアントアプリケーションは、それに基づき巡回型 SOAP メッセージを作成する。この際、クライアントアプリケーションは、巡回型 SOAP メッセージに一意の識別子を生成する。利用者の端末とサービスプロバイダは、この識別子を記憶し、処理結果の格納、受け取り、障害復旧に使用する。クライアントアプリケーションは、巡回型 SOAP メッセージをサービスプロバイダへ送信し、処理を依頼する。

サービスプロバイダは、メッセージを受信すると、メッセージ内の処理可能なフォーマットを処理し、格納された要素を取り出す。その際、multiRef タグを参照する記述があると、multiRef タグの要素を取り出す。そして、サービスプロバイダは、サービスの実行や次の送信先の決定を行う。サービスプロバイダは、処理結果を巡回型 SOAP メッセージに格納し、次のサービスプロバイダに送信するか、もしくは、巡回処理が終了した場合や処理終了時間が経過した場合は、利用者の端末へ送信する。

### 3.4 耐障害性

サービスプロバイダは、サービスの処理結果を自身の持つメッセージボックスに書き込み、メッセージの転送先の処理の成否を監視する。サービスプロバイダは、自身の処理の成否を SOAP メッセージの送信元に通知する。障害発生時、サービスプロバイダのメッセージボックスは、ログとしての働きを担う。メッセージボックスに保存されている処理結果は、障害直前に行われた処理結果であり、そのデータから SOAP メッセージの転送処理を復旧できる。

## 4 おわりに

本稿では、複数の Web サービスを複合したサービスを提供するための基盤システムにおける巡回処理方式について述べた。巡回処理方式は、サービスプロバイダによる巡回型 SOAP メッセージの転送によって実現される。この方式は、常時駆動しているサーバを必要としない、サービス要求元が処理結果を待つ必要がないなどの利点があり、データベースの巡回検索などに適用可能である。

今後は、サービスプロバイダの移動に適応する巡回型 SOAP メッセージの記述法を検討する予定である。

### 参考文献

[1] 西山 正浩, 毛利 公一, 大久保 英嗣: ユビキタス環境におけるトランザクション処理方式, 情報処理学会研究報告, 2004-OS-098, Vol. 2005, No. 16, pp. 105-110 (2005).

[2] WSBPEL: Web Services Business Process Execution Language, <http://docs.oasis-open.org/wsbpel/2.0/wsbpel-specification-draft.html>