

入出力データ型に透過な Web サービス動的実行システム

越田 高志[†]

松江工業高等専門学校[†]

1. はじめに

Web サービスは今後の CALS/EC の核となる分散処理技術として認識されているが、未だ広くビジネス分野で利用されるまでには至っていない。その原因として以下の問題点が考えられる。まず、1. 必要とする適切な Web サービスを検出し、それらを連携し、目的とする処理を実行する適切な手段が存在しないこと、2. ユーザ側で Web サービスの実行を制御するスタブを開発しなければならないこと、及び、3. 入力データ型が複合型である Web サービスを利用できないこと、である。1 に関しては、プリミティブ Web サービス(以下 PWS)という概念を提唱[1,2]し、PWS を利用してそれを解決する手法を提案している。2 に関して、我々は出力データ型に依存しない Web サービス動的実行システム[3]を開発し、スタブレスで Web サービスを実行する方式を示して、その問題を解決した。今回我々は、3 の問題点を解決する手法を考案し、それを実装したシステムを開発したので、その詳細について述べる。また、本システムを利用した、PWS による Web サービス標準化の可能性についても言及する。

2. Web サービスの問題点

ユーザが Web サービスを実行するためには、その Web サービスに対するクライアントプログラム(以下スタブ)を開発しなければならない。ユーザの大きな負担になっていた。我々は WSIF[4]をベースにして、Web サービスの出力データ型に依存しないスタブレスシステムを開発し、Web サービスの実行時に必要な処理を自動化することで、ユーザ負担を軽減し、Web サービスのユーザビリティを向上させた。Web サービスの出力データ型は、基本型と複合型(複数の任意の基本型データから構成される)に大別される。従来方式では、複合型出力データの場合、Web サービスからの出力データを受け取るための JavaBeans クラスをユーザ側が事前に開発しなければならない。そのため WSDL[5]データからの動的実行が不可能であった。我々は、その JavaBeans クラスを Web サービス実行時に動的に生成する手法を考案・実装することで、複合型に対しても WSDL データからスタブレスな動的実行を可能にした。この問題は入力データ型でも同様であり、複合型入力データの場合には、入力データを Web サービスに送付するための JavaBeans クラスを事前にユーザ側で開発しなければならない。ユーザにとって大きな負担となる。従って、現在公開されている全ての Web サービスは、その事前処理が不要である基本型入力データをもつ。しかし、入力データ型が、複数の基本型パラメータをまとめた複合型であれば、1つのオブジェクトとして操作できるので、コードがシンプルになり、開発と処理操作が容易になる利点がある。また、それにより Web

サービスの入力インターフェース(以下 IF)が統一され、PWS を利用した Web サービスの標準化へと進展できる[2]。そこで、我々は、複合型入力データをもつ Web サービスに対しても、ユーザ側で一切の事前プログラム開発が不要な完全なスタブレスシステムを開発し、それを基に PWS による Web サービス標準化を目指す。

3. 入出力データ型に依存しない Web サービス動的実行システム

2章で示した Web サービスの入出力データ型に関する制約のため、公開されている Web サービスの入力データ型は基本型のみである。そこで我々は、その制約を解決し、ユーザに負荷をかけることなく、複合型入力データをもつ Web サービスが動的に実行可能な手法を開発し、入出力データ型に依存しない動的実行システムとして実装した。本システムを用いることにより、次に説明する PWS が動的に実行可能となる。

3.1 プリミティブ Web サービス(PWS)

PWS とは、様々なビジネスモデルに対して、共通に利用可能な一意に統一された名称、機能、入出力 IF を持つ標準化された Web サービスとして定義される。例えば、商品調達のビジネスプロセスにおいては、在庫確認、価格見積り、受発注などが PWS として考えられる。現在 Web サービスに対して、その名称、機能、入出力 IF に関する統一規約はなく、全て提供側に委ねられている。この PWS を実装する場合、入出力 IF を提供側に依存しない一意な形式にするためには、そのデータ型を基本型として個々に定義するのではなく、まとめて複合型として定義すればよい(図1参照)。また、その複合型名を一意に定義しても、それを構成する個々の基本型データは提供側で任意に設定できるので、本システムを利用することで、入力データ型を意識することなく、Web サービスが動的に実行できる。

```
public class GoodsService6 { // (A)
    public SData6 getStocks(String btype, String id, String gcode,
        int quantity, int price) throws Exception {

public class GoodsService6 { // (B)
    public SData6 getStocks(GData6 details) throws Exception {
```

図1 入力データ型が(A)基本型と(B)複合型である場合の Web サービスメソッドの定義例

現在公開されている複合型出力データをもつ Web サービスの入力 IF の例を図2に、その出力表示例を図3に示す。この出力結果は Web ブラウザで表示されるので、目視で確認する以外に、プログラムで自動処理することができない。しかし、本システムを利用すれば、データ型に関係なく、その出力結果をユーザ側プログラムで自動処理可能となる。

[†] An Input-output Data Type Independent Dynamic Invocation System for Web Services

[†] Takashi KOSHIDA

Matsue National College of Technology

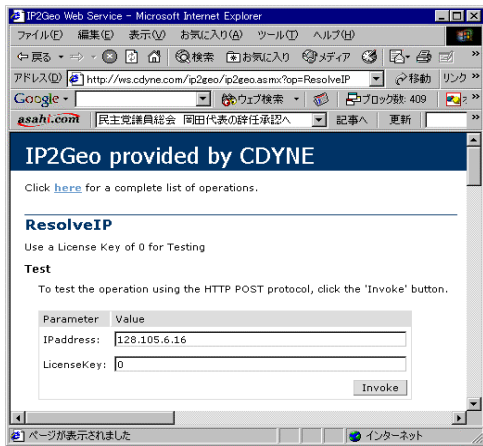


図2 複合型出力データをもつ WS の入力 IF 例

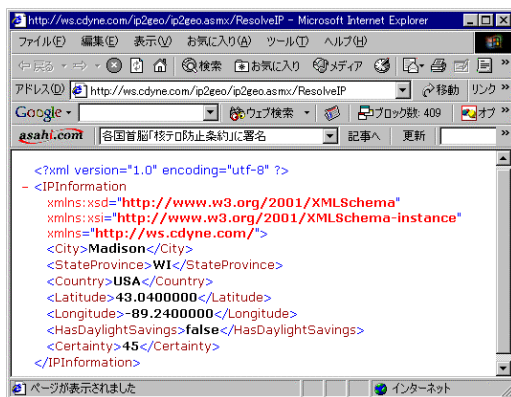


図3 複合型出力データをもつ WS の出力表示例

3.2 システムの機能

本システムは WSIF をベースに構築されており、文献 [3] の研究成果を利用している。システムは、Web サービスの WSDL URI とその Web サービスに対する入力パラメータ値を入力データとして実行される。まず、WSDL データを解析して、入出力データに対応した JavaBeans クラスを自動生成する。入力パラメータ値は、setter メソッドを用いてオブジェクトに格納する。Web サービス実行後に出力データオブジェクトを解析し、getter メソッドを用いて、個々の出力データを抽出する。図4に入出力データが複合型である Web サービスの WSDL データを示す。

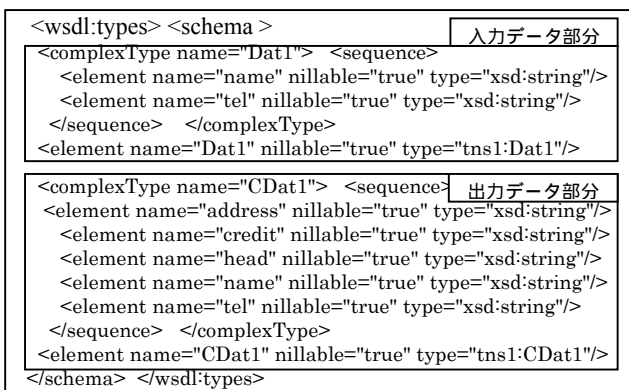


図4 複合型入出力データをもつ Web サービスの WSDL (一部)

3.3 実験結果

本システムの実験結果を図5に示す。これは図4に示した複合型入出力データ型をもつ Web サービスに対する実行結果である。入力パラメータ値は Web サービスの WSDL の URL と Web サービスに対する入力データである企業名、電話番号である。出力データは JavaBeans クラス名 CDat1 のオブジェクト (全て文字列型の5データから構成される) である。それを動的に解析して、個々の文字列型データ値を抽出し、表示している。

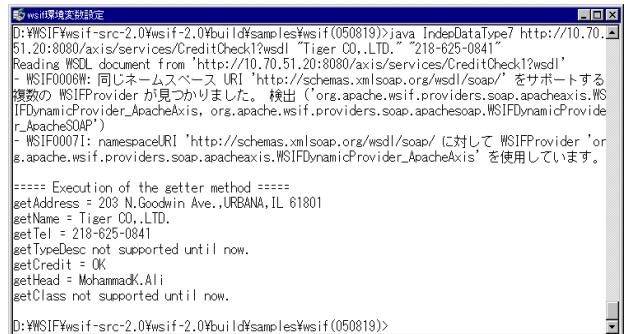


図5 本システムによる Web サービス実行例

Web サービスの入力データ型、及び出力データ型は各々基本型と複合型の2種類に大別できるので、その組合せは全部で4パターンである。本システムで同じ形式で動的に実行できる。

4. おわりに

本システムを利用することにより、複合型入力データをもつ Web サービスに対しても動的実行が可能になる。これにより、Web サービス提供側でも複合型入力データをもつ Web サービスが提供可能となり、提供側の Web サービス開発工数と処理効率の向上が期待できるとともに、Web サービスメソッドの入力 IF の統一が容易になる。従って、基本的なビジネスプロセスに対応した、一意に定義された Web サービス名、メソッド名、入力 IF をもつ Web サービス標準化が実現でき、また Web サービスの動的連携も容易になると考えられる。

現在、RosettaNet の PIP に沿って、PWS として標準化された Web サービスを開発中であり、実証実験を通して PWS の効果と有効性を示したい。

[参考文献]

- [1] 越田高志, 植村俊亮: プリミティブWeb サービスとエージェントによる商品調達システムの設計と実装, Proceedings of DBWeb2004, pp.9-16 (2004).
- [2] 越田高志, 植村俊亮: プリミティブWeb サービスの入出力データに関する一考察, 電子情報通信学会2005年総合大会論文集CD-ROM, 論文番号D-9-1, 2005年3月.
- [3] 越田高志, 波多野賢治, 天笠俊之, 宮崎純, 植村俊亮: 自動化・統合化されたWeb サービス動的実行システム, 情報処理学会論文誌, Vol.46, No.3, pp.671-682 (2005).
- [4] WSIF, <http://ws.apache.org/wsif/>
- [5] WSDL, <http://www.w3.org/TR/WSDL/>