

# Webサービスにおけるリクエストプログラムの再利用に関する研究

2000MT001 秋田太陽 2000MT051 丸吉祐也

指導教員 青山幹雄

## 1. はじめに

現在Webサービスは、インターネット上の様々なソフトウェア資源を、XML(eXtensible Markup Language)を介して提供・再利用する技術として注目されている。現在の技術では各サービスに対してリクエストプログラムを個々に開発しなければならない。各リクエストプログラムは、それに対するサービスと密に結びついているので、そのリクエストプログラムを用いて他のサービスを利用することはできない。そこでわれわれは、あるサービスを利用できるリクエストプログラムで他のサービスを利用する方法を提案する。

本研究の目的は、あるリクエストプログラムで複数のサービスを利用できる仕組みを提案して構築し、リクエストプログラムの再利用を実現することである。

## 2. 背景と問題点

### 2.1. 課題

現在、Web サービスを利用するときは、新たなサービスを発見する度に、リクエストプログラムを開発しなければならない。そして、あるリクエストプログラムはそれに対するサービスにしか使用することができない。その原因は、様々なサービスが存在しているが、各サービスは独立しているからである。そのため、同じようなサービスが複数存在しても、それらには関係が全くない。

サービスが独立する理由は、サービスのインターフェースの不一致が原因であり、以下の問題点がある。

- (1) 属性名と機能名: 入出力の属性名やサービスの機能名に違いがある。
- (2) シグネチャ: 入出力要素の型やその順番そしてその数に違いがある。

## 3. 解決方法

課題を解決するための仕組みとして、サービスのグループ化(以下グループ化と呼ぶ)とSOAPメッセージの変換を提案する。まずはサービスをグループに分類し、グループ内で共通する項目を抽出する。そして共通する項目をまとめたものを仲介項目し、それを利用してサービス同士の

要素を対応させる。次に、その対応を元にSOAPメッセージの変換を行う。SOAPメッセージの変換によってインターフェースの不一致を吸収し、リクエストプログラムはグループ内の複数のサービスを利用できるようになる。

### 3.1. グループ化

#### 3.1.1. 概要

様々なサービスを扱うデータを基準にして、適切なグループに分類する。各グループ内では、サービス間に対応付けるために必要な仲介項目を決定し、共有する。そして、仲介項目を介して各サービス間の要素に対応関係を持たせる。また、各サービスの情報をグループ内で管理する。

#### 3.1.2. サービスの分類

様々なサービスを同じようなサービスごとに分類していく。分類することによって、サービスをグループごとに整理でき、サービス間の関係の構築が容易となる。

例としてサービス A, B, C, D, E があるとする。A, B, C が図書サービスに関連しているならば、A, B, C は図書グループに含まれる。D, E が航空サービスに関連しているならば、D と E は航空グループに含まれる。

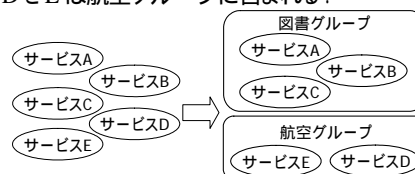


図1: サービスの分類

#### 3.1.3. 分類の基準

分類はサービスが扱うデータを基準にして行う。それはデータによってサービスの内容が表されるため、データから何のグループに属するかを判断できるからである。

#### 3.1.4. グループ内における仲介項目の決定

グループ内の各サービスが扱うデータの中で、意味的に共通するものを決定し、それを仲介項目とする。ここで仲介項目とはグループ内に共通する属性や機能を集めたものであり、サービス間に関係を持たせるために必要なものである。仲介項目の決定方法を、図2を用いて説明する。各図書サービスは、図書サービスの概念から生まれる。それ

らは異なるサービスであるが、共通の概念から生まれていることから、それらには共通する項目がある。そこでわれわれは、二種間以上に共通する可能性のある項目を図書グループの仲介項目とし、それを図書サービスの概念と対応するように抽出する。各グループは仲介項目 XML 文書を作成する。

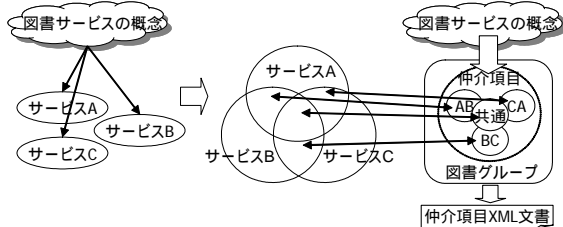


図2: 仲介項目の決定

### 3.1.5. サービス間の対応関係

3.1.4 で説明した仲介項目を介して、グループ内のサービス同士の属性や機能を対応させる。そのための仕組みを、図3を用いて説明する。尚、本研究では一対一の対応関係のみを扱う。

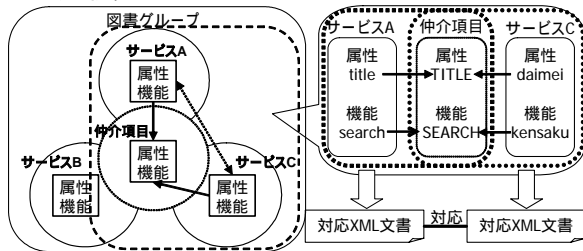


図3: サービス間の対応関係

サービス A とサービス C の各要素を対応付けるには、サービス A の各要素と仲介項目の各要素を対応させ、同様に、サービス C と仲介項目を対応させる。これによって仲介項目を介して、サービス間に関係を持たせることができる。各プロバイダは仲介項目 XML 文書を元に対応 XML 文書を作成する。

### 3.1.6. グループの管理

グループの情報やグループに属しているサービスの情報は管理する必要がある。具体的には、仲介項目 XML 文書、各サービスの名前空間、対応 XML 文書の URL 等を管理する。また、SOAP メッセージの変換を広く利用してもらうために、そのアクセスポイントの URL も管理し公開する。

## 3.2. SOAP メッセージの変換

### 3.2.1. 概要

SOAP メッセージの変換とは、サービス同士の要素の対応関係を元に、ある形式の SOAP メッセージを意味的に等価な別の形式に変換することである。変換の対象となるの

は要素名とメッセージの構造であり、その具体的処理は要素の値を変化させずに新たに SOAP メッセージを作成することである。この SOAP メッセージの変換をプロカーで行うことによって、一つのリクエストプログラムで複数サービスの利用を実現する。図4に SOAP メッセージの変換を用いた複数サービス利用のイメージを示す。

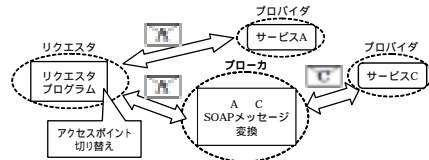


図4: SOAP メッセージの変換を用いたサービスの利用

### 3.2.2. デフォルト値と nil の設定

デフォルト値と nil の設定は対応関係がとれない要素に対して、値の情報を補うための設定である。この設定によって SOAP メッセージの変換を促進させる。

デフォルト値は要素に対してあらかじめ定められている値であり、その要素が他のサービスの要素と対応関係が取れない場合に適用される。また、nil の設定は空要素にできることを明示する設定である。どちらの設定もプロバイダが任意に設定するもので、通常は設定してもサービスの品質を落とさない要素に対して設定する。

### 3.2.3. 変換の方向性

変換には方向性が考えられ、例えばサービス A の SOAP メッセージをサービス B の SOAP メッセージへ変換できるならサービス B からサービス A への変換ができるとは限らない。しかし、リクエストメッセージとレスポンスメッセージの変換は互いに異なる方向への変換であるため、双方向に変換できなければサービスを利用することはできない。

変換の方向性については SOAP メッセージを変換できるかどうかによって決まり、要素の対応関係、デフォルト値と nil の設定から決定される。

### 3.2.4. エラーの対処

元の SOAP メッセージの内容、要素の対応関係、デフォルト値と nil の設定の組み合わせから SOAP メッセージを変換できない場合があり、そのエラーを SOAP フォルトとしてリクエストに通知する。エラーはリクエストとレスポンスのどちらか一方でも SOAP メッセージを変換できない場合の変換実行時に発生する。

## 4. プロトタイプ開発と実験

### 4.1. グループ化

グループ化では、仲介項目 XML スキーマと対応 XML スキーマの作成を行い、グループの例として、図書グループ

プの実装を行う。

#### 4.1.1. 仲介項目 XML スキーマ



図 5: 仲介項目 XML スキーマのクラス図

3.1.5 で説明した仲介項目 XML 文書を作成するための XML Schema(以下 Mediation.xsd)を作成する。要素は group(ルート要素), dataName(item を子要素とする), methodName(item を子要素とする), item(仲介項目の各要素名を属性として持つ)の4つである

#### 4.1.2. 対応 XML スキーマ



図 6: 対応 XML スキーマのクラス図

3.1.5 で説明した対応 XML 文書を作成するための XML Schema(以下 GroupRelation.xsd)を作成する。要素は, service(ルート要素), method(名前空間の情報を持つ), methodPart(仲介項目での機能の要素名と, サービスでの機能の要素名を属性として持つ), requestMessage(リクエストメッセージの内容を持つ), responseMessage(レスポンスメッセージの内容を持つ), parts(各メッセージの要素を持つ), part(仲介項目での属性の要素名とサービスでの属性の要素名を属性として持つ)の7つである。

#### 4.1.3. 図書グループの作成

ここでは, 図書グループを管理する仕組みを構築する。図書グループの名前を BookGroup とする。

- (1) 図書グループの仲介項目 XML 文書の作成  
グループ内の各サービスに公開するための仲介項目 XML 文書を作成する。
- (2) BookGroup のデータベースの作成  
各サービスの登録情報や SOAP メッセージ変換のアクセスポイントの情報を保持するデータベースを用意する。
- (3) BookGroup への登録 Web サービスの開発  
BookGroup への登録として, (2)のデータベースに情報を登録するための Web サービスを開発する。

## 4.2. SOAP メッセージの変換

SOAP メッセージの変換は JAXM(Java API for XML Messaging)の SOAPMessage クラスと SOAPConnection クラスを用いたサーブレットとして実現する。図 7 に SOAP メッセージ変換ブローカのアーキテクチャを示す。

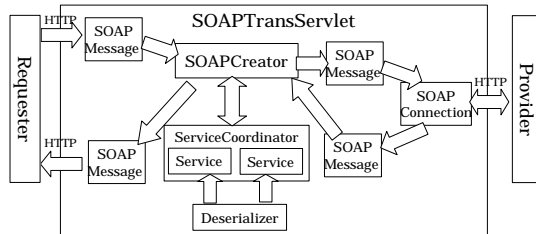


図 7: SOAP メッセージ変換ブローカのアーキテクチャ

SOAPTransServlet クラスがサーブレットであり SOAP メッセージの送受信や変換の制御を行う。Service は対応 XML スキーマで定義した要素をクラス化したもので, サービスのメッセージ構造と対応関係の情報を保持する。このクラスをサーブレットの起動時にインスタンス化するのが Deserializer クラスである。そして, ServiceCoordinator クラスはサービス間の関係を管理し, その情報を提供する。SOAPCreator クラスは図 8 の手順で変換を行い, その結果として SOAPMessage オブジェクトを生成する。



図 8: SOAP メッセージ変換のシーケンス図

## 4.3. 実験

実験の手順として, まず, SearchBookService と FindBookA という二つの図書サービスを開発し, それぞれ対応 XML 文書を作成する。次に, 4.2 で開発した SOAP メッセージ変換ブローカを二つの図書サービスに対応させ, サービスの名前空間や対応 XML 文書の URL 及び, 変換ブローカのアクセスポイントなどの情報を 4.1.3 で作成した図書グループに登録する。そして, それぞれのサービスのリクエストプログラムから SOAP メッセージ変換ブローカを介して二つのサービスを利用し, 評価を行う。

## 5. 評価

### 5.1. グループ化

グループの存在によって、複数あるサービスをまとめることができた。しかし、プロバイダが自身のサービスを属すべきではないグループに登録する可能性があるため、グループに登録の可否を判断させる必要がある。各グループが仲介項目の XML 文書を公開し、それを元に各プロバイダが対応 XML 文書を作成することでサービス間に対応関係付ける仕組みを構築できた。しかし、要素の内容の意味までを考慮した対応が取れていないこと、一対多、多対多の対応関係が表現できていないことが問題であり、その構築が必要である。また、サービス登録の仕組みは構築できたが、サービスの登録、削除などの操作に権限を与えなければならない。

### 5.2. SOAP メッセージの変換

実験の結果、変換された SOAP メッセージは各プロバイダ、リクエスタで正しく処理され、両方のサービスを利用できた。一部の対応関係のない要素にはデフォルト値が適用されたが、サービスの利用に問題はなかった。

次に、表 1 にソースコードのサイズ、図 9 にサービス平均応答時間を示し、再利用性を評価する。表 1 より、試作した SOAP メッセージ変換プロローカの開発コストはリクエスタよりも高いことがわかる。しかしプロローカは複数リクエスタで共有でき、また再利用可能な設計であるため、変換の利用が増えれば、個々にリクエスタプログラムを開発するよりもコストを抑えることができる。

表 1: ソースコードのサイズ

プログラム	行数	行数の比率
SearchBookService リクエスタ	568	1.00
FindBookA リクエスタ	811	1.42
SOAP メッセージ変換プロローカ	2,193	3.86

また、図 9 より、変換プロローカを介すとサービス応答時間は長くなる。しかし、通常考えられる 1 回、10 回の利用では応答時間に大きな差はなく、SOAP メッセージ変換が与える影響は少ないため、変換を利用する価値は十分にある。

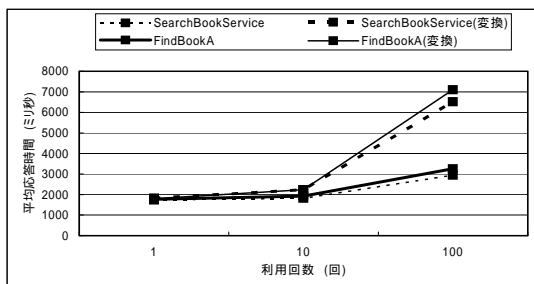


図 9: サービス平均応答時間

## 6. 考察と今後の課題

### 6.1. グループ化

今後の課題として、各グループのサービス登録時における審査と操作権限の付加、各サービスにおける要素の意味を考慮した対応関係の構築、及び一対多、多対多の対応関係の構築が挙げられる。

### 6.2. SOAP メッセージの変換

本研究で試作したシステムではサービスの組み合わせの数だけ変換プロローカが必要となる静的な変換であった。これに対して、変換要求を SOAP ヘッダに埋め込み、プロローカでは SOAP ヘッダを解釈しながら、必要な情報を取得し、動的に変換するように改良する必要がある。

## 7. まとめ

本研究では、グループ化の概念と SOAP メッセージの変換を提案した。これらによって、一つのサービスの型や引数の数などの制約に縛られているリクエスタプログラムで複数のサービスを利用できるようになり、リクエスタプログラムの再利用を実現した。その結果、Web サービスの利用範囲が広がり、Web サービスの利用が促進されることが期待される。

## 参考文献

- [1] Apache Web Services Project: Axis Documentation, <http://ws.apache.org/axis/java/index.html>(2003. 10)
- [2] B. Meyer (二木厚吉ほか訳): オブジェクト指向入門, アスキー出版局 (2002.10)
- [3] DIMENSIONAL DATA, INC.: XML, <http://www.techscore.com/tech/XML/index.html>
- [4] Eric van der Vlist (田村健人ほか訳): XML Schema, オライリー・ジャパン(2003.3)
- [5] H. M. Deitel ほか (吉舗紀子訳): Java プログラムのための Web サービス大全, コンピュータ・エージ社 (2003. 6)
- [6] 西澤秀和ほか: Web サービス分析・設計ガイド, オブジェクト指向によるモデリングの手引き, ソフトバンクパブリッシング (2002.12)
- [7] 日本ユニテック Digital Xpress 編集部: SOAP/UDDI/WSDL Web サービス技術, 技術評論社 (2002. 12)
- [8] Sun Microsystems: Java(TIM) APIs for XML Messaging Specification 1.1 <http://java.sun.com/xml/downloads/jaxm.html>(2003. 10)