

## 既存 Web アプリケーションを活用した Web サービス構築法

鈴木 亮<sup>†</sup> 小泉 寿男<sup>†</sup>

<sup>†</sup>東京電機大学 理工学研究科 情報システム工学専攻

企業の情報システムでは『迅速に変化へ対応する』ことが、従来以上に求められている。それを実現するためのアーキテクチャとして注目を浴びているのがサービス指向アーキテクチャ (SOA: Service Oriented Architecture) である。SOA では、Web サービスを利用することで個々のアプリケーションの開発言語や動作環境などが依存せず、共通のメッセージ交換インタフェースに対応していれば、システムの連携ができる。Web サービスは主に B2B など利用されるケースが考えられるが、企業内の情報システムにも活用可能と考えられる。しかし、既存の情報システムを SOA に基づくシステムに移行するには、各システムをサービスとして作り替える必要がある。本稿では、既存の Web アプリケーションを活用して、システムを SOA に移行するための Web サービス構築法を提案する。

### An Implementation Method of Web Services using Existing Web Applications

Ryo Suzuki<sup>†</sup> Hisao Koizumi<sup>†</sup>

<sup>†</sup>Department of Computers and Systems Engineering

Graduate School of Science and Engineering

Tokyo Denki University

In the information system, rapid response to changing is requested than ever before. Service-Oriented Architecture (SOA) is receiving special attention as architecture to realize rapid response to changing. On SOA, systems can cooperate and exchange interface by corresponding the common message independent of languages used for each application or system requirements. The technology of Web services is the most likely connection technology of service-oriented architectures. The spread of Web services speeds the attention of SOA. Web services is thought that it is often used at B2B and can be assumed as enterprise information system. But to change from existing information system to system on SOA systems must be transferred to Web services element. This paper proposes a Web services construction method to change system to SOA with existing Web application and we also describe the implementation and evaluation of the proposed development method.

#### 1. はじめに

現在、企業が情報システムを活用していないことは皆無に等しい。情報システムは企業内の複雑な業務を担い、業務の効率化を図っている。しかし業務内容の急な状況変化により、情報システムには追加や変更が行われ、その結果、システムは段階的に構築されることになり、多種多様なシステムが混在する状況となっている。その中でも既存システムの部分修正や、システムの統合などさらに柔軟に変化に対応することが求められてきている。

このような背景の下に、SOA や Software Factory などの、部品化されたサービスを定義・開発し、それらを組み合わせてアプリケーションを構成する技術体系が注目を浴びてきた。SOA は 1996 年米国の

ガートナーの論文で提唱され、Web サービスの登場により新たに注目を浴びている。SOA の異種環境でも利用可能なインタフェースの提供が Web サービスの標準化によって可能となった。

Web サービスは主として B2B などに利用されるが、社内システム間にも Web サービスを適用することで、複数のアプリケーションを整理統合、自社系列会社とネットワークを通じた機能の共有、新しいビジネスチャンス、既存する機能のオープン化など改善が考えられる<sup>[3]</sup>。既存のソフトウェア開発の資産を新規開発システムに再利用することで、重複開発を減らし、効率的な開発の実現が可能と考えられる。

SOA は概念的なものであり、またサービスに明確な定義はなく、粒度もシステムによってまちまちである。また既存のソフトウェアを再利用してサービ

スにする方法は確立されていない。

これまでに Web アプリケーションを Web サービスに変換する方法として以下のような研究が挙げられる。

### ① Proxy サーバによる Web サービス変換<sup>[9]</sup>

リクエストとプロバイダの間に Proxy サーバをおくことでサービスを提供するアーキテクチャ

### ② ページ遷移による Web サービス変換<sup>[9]</sup>

ページ遷移単位でモジュール化された Web サービスを連携してサービスを提供するアーキテクチャ

現状の既存システムは、オブジェクト指向によるソフトウェア開発が主流である<sup>[10]</sup>。本研究ではオブジェクト指向ソフトウェアである既存の Web アプリケーションを活用し、Web サービスの構築法を提案する。本研究で構築する Web サービスは、SOA で利用されるサービスの目的としている。そこで研究ではサービス抽出工程とリポジトリ登録工程から既存の Web アプリケーションからの Web サービス構築、Web サービスを利用したシステムの移行を行う。サービス抽出工程では本研究で開発するサポートツールを用いて Web サービスモジュールとなるコードを生成する。これはメソッド単位で抽出し、複数のクラス群 (コンポーネント)、または1つのクラスで構成される。さらにリポジトリ登録工程ではサービスの検索がニーズに即した検索を行うことができることを目的としている。

## 2. SOA 関連技術

SOA の実現・活用する技術として Web サービス、BPM、BPEL4WS についての機能概要を示す。

### 2. 1. Web サービス

Web サービスは、注目されている次世代のシステム基盤技術の一つであり、現在の業務システムに有効利用できる技術である。Web サービスの特徴として以下のようなものが挙げられている<sup>[11]</sup>。

- ・インターネット上で XML 技術によって連携される、URI で識別されたソフトウェア・アプリケーション
- ・インターネット上の RPC (Remote Procedure Call)
- ・XML, SOAP などの標準技術を利用したシステム間連携

従来の Web アプリケーションでは、ブラウザから入力データを受け取り、Web 上で動的に HTML 文書を生成していた。しかし結果が HTML 形式のため、さらなるデータの加工・保存は困難であり、くわえてシステム間のデータ受け渡しにはいまだに人が介

入している作業が存在する。CORBA や DCOM では、仕様の制約の大きさ、独自仕様の問題から、同仕様の製品でなければ連携できないという課題がある。

このような従来の Web アプリケーション、分散オブジェクト技術のシステム連携・データ交換の問題を解決につながるのが Web サービスである。Web サービスは、ほかのビジネスロジックと密な関係を持たず、処理単位でビジネスロジックのインタフェースを提供する能力があり (自己完結)、インタフェースが特定の実装などに依存することなく独立して提供されているアプリケーションとして位置付けられている。また標準化された XML メッセージを使用してネットワーク経由でアクセス可能にし、相互運用性を保つようになっている。この Web サービスを利用することで、ソフトウェアの再利用性が高まり、異なるプラットフォーム、言語で開発されても利用できる共通コンポーネントとなる。

## 2. 2. BPM (Business Process Management)

ビジネスプロセスモデリングとは現状の人材やシステム、情報などの資産を把握することにより業務プロセスをビジネスプロセスとしてモデル化し可視化・効率化を図る手法である<sup>[9]</sup>。システムを実現する組織の構造とそのシステムに必要な人材などの資源を関連付けることによるモデリングが可能であり、より業務プロセスを把握することができ、業務プロセスの改善の向上につながるという利点がある。図 1 に BPM のメリットを示す<sup>[9]</sup>。またビジネスプロセスを可視化することにより、ビジネスポイントを明確にすることが出来るという利点がある。

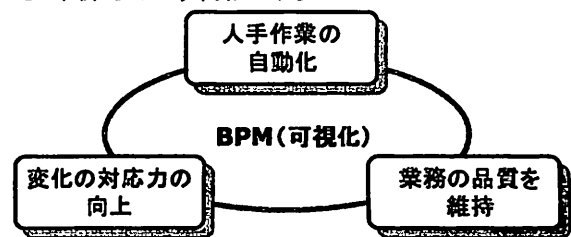


図 1 BPM の利点

現在、ビジネスプロセスの標準的な表記法は決められておらず、多くの表記方法を用いたビジネスプロセスモデリングツールが世間には多く存在する。このため、BPM ツールは製品間での相互接続が困難で限られた中での利用となった。しかし Web サービスを利用することで、孤立した BPM ツール間での接続を実現し、業務プロセスの連携を構築できるようになった。

業務プロセスの記述方法の一つに BPMN

(Business Process Modeling Notation) がある。BPMN はモデル使用の策定団体である BPMI が策定したもので、UML との統合が図られるという見方が強くなり、BPM ツールの標準となることが期待されている。業務処理の種別に応じて表記が決められており、基本的に次項で述べるサービスの呼び出し順序をフロー形式で記述する言語である BPEL4WS に 1 対 1 で変換できることを目的としている。

### 2. 3. BPEL4WS

現在 Web サービスの連携について、連携のための専用言語を開発し、その言語を解釈、実行するエンジンを構築することによって複合 Web サービスを実現する BPEL が開発された<sup>[7][11]</sup>。

Web サービスではこれまで、SOAP/WSDL によりサービス・インタフェースの記述が標準化されたが、個々のプログラムを Web サービスとして呼び出すには別途ラッピングが必要であった。そのため、複数の Web サービスを組み合わせ、すぐ使える Web サービスとして利用することが出来なかった。そこで BPEL では図 2 のように一連のプロセス (処理の流れ) をアクティビティ (処理単位) を繋げたフロー (遷移グラフ) として構成することによって複合 Web サービスを実現している。

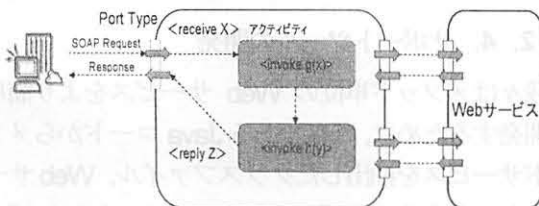


図 2 BPEL4WS の構成

BPEL4WS は、複数のサービスを連携させて新しいサービスとして公開するためのビジネスプロセス記述言語である。BPEL ではアクティビティと呼ばれる制御構文によって記述され、sequence, while, switch といったアクティビティと flow, link などのフローグラフ記述のためのアクティビティを組み合わせることによってビジネスプロセスを記述する。BPEL プロセス文書はそれ自身が Web サービスとして公開され、receive アクティビティによってプロセスが呼び出しのリクエストを受け取る。その後各アクティビティによってビジネスプロセスが実行され、reply アクティビティによって結果を返す。

図 3 に上記の基盤技術から構成される SOA のシステム構成を示す<sup>[2]</sup>。全体としては、業務プロセスを BPM ツールを用いて定義し、BPEL4WS に変換して存在する Web サービスを利用し動作させる。Web

サービスはインタフェースである WSDL を利用して、SOAP で呼び出される。

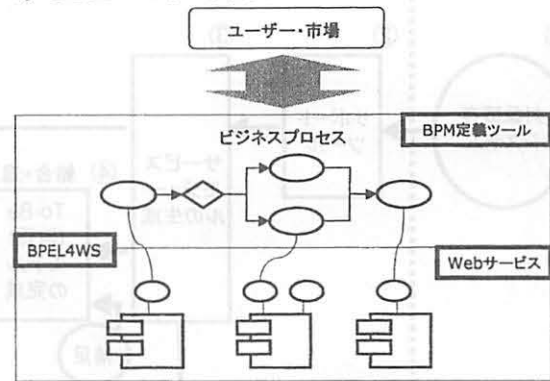


図 3 SOA のシステム構成と関連技術

### 3. Web サービス構築法

#### 3. 1. 本研究の概要

我々は既存の情報システムの SOA アプローチを明確にし、Web サービスを利用したシステムへ移行するための再設計構築基盤の確立を目指している。図 4 に本研究の SOA アプローチを示す。図 4 のように本研究のアプローチとしてサービス構築工程がある。リポジトリ登録工程については、概要を大まかに述べる。

#### 3. 2. サービス抽出工程

##### 3. 2. 1. サービスの粒度

サービスの粒度は、以下のように定義されることが多い。

- ・単一、または複数の機能を提供する
  - ・オブジェクトやコンポーネントより粒度が大きい
- これはオブジェクトやコンポーネントは、システムやアプリケーションの構成要素として利用されることが主体であるのに比べ、サービスは単体で実際の業務の一部や特定のビジネスプロセス処理を実現するとされているからである。しかし、明確な定義づけはされていない。

そこで本研究では、Web アプリケーションのロジック部分にあるメソッドをサービスとして抽出する。この際、サービスとして抽出されるモジュールは複数のクラスで構成されていることや、一つのクラスであることも考えられる。SOA におけるサービスとは、『独立して記述されるインタフェースの存在で、すべての言語や異種システム環境でも実現可能であり、他との依存関係を持たない自立したソフトウェアモジュール』と言われている。これはモジュールに対するインタフェースが定義されれば、そのモジュールの構成や処理はリクエストに対して隠蔽

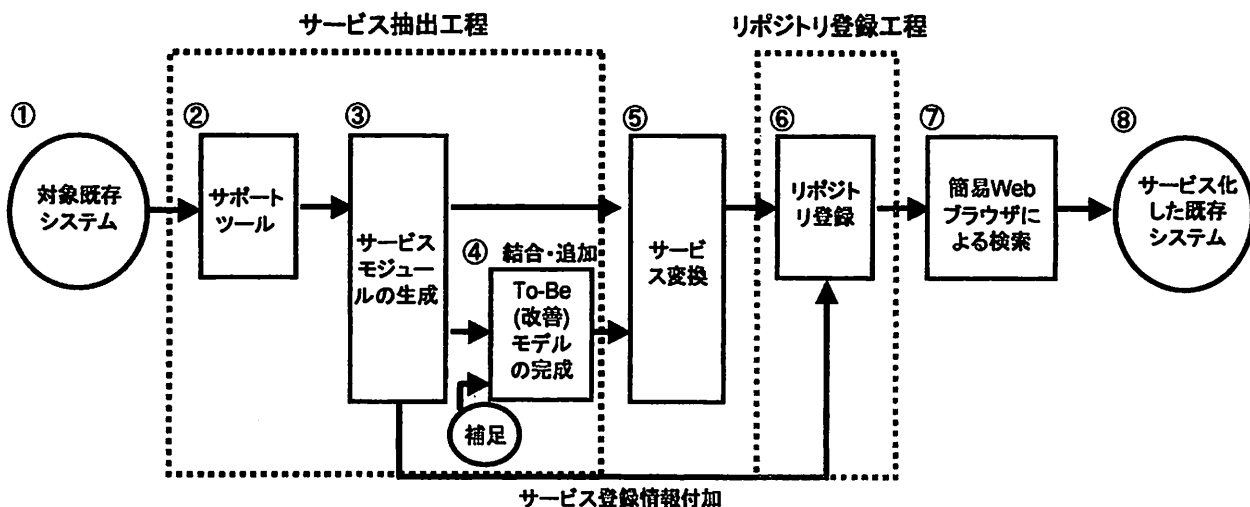


図4 Web サービスを利用したシステム移行のための再設計構築基盤

されていることになる。つまり1つのクラスであっても処理が可能であれば、SOAを構成するサービスになると我々は考えている。

### 3. 2. 2. Web サービスのステート管理

Web サービスへのアクセスは HTTP のようなステートレスで信頼性の低いプロトコルで行われることが多い。つまり Web サービスは、極力ステートレスの方が良いと考えられる。SOAP はステートの管理を標準仕様としていないことを加味すると、サービスの呼び出しは1回ごとに完結していたほうが良い。

### 3. 2. 3. メソッド単位の Web サービス

抽出されるモジュールのメソッドとして以下の4つが考えられる。

1. 戻り値を持つメソッド
2. setter メソッド
3. execute メソッド(戻り値を持たないメソッド)
4. getter メソッド

1. 戻り値を持つメソッドは、最も基本的なメソッドである。2. 3. 4. のメソッドは、アクセッサメソッドと呼ばれており、オブジェクト指向のポリモρφイズムに則るとメンバ変数の参照・操作・取得の操作が必要になる。これらアクセッサメソッドがあることでクラス内部のデータ表現を変えた場合でも呼び出し側のコードを変更する必要がないなどのメリットがある。

以上のメソッドについて Web サービス変換を考える。1. 戻り値を持つメソッドは、一度の呼び出しで処理・結果取得を行えるため Web サービスのステート管理より、サービスとして適した形であると言える。しかし 2. 3. 4. のメソッドは、共通する機

能を3つのメソッドで実現するため、『Web サービスはステートレス』の視点から Web サービス変換は難しい。そこで図5のようにアクセッサメソッドを用いてサービスモジュールとなるプロキシコードを生成する。

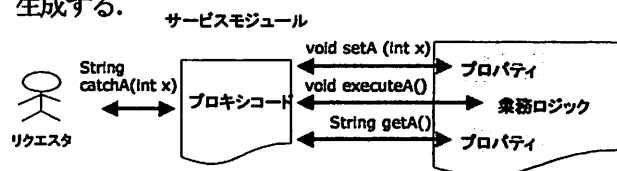


図5 サービスモジュールとなるプロキシコードの生成

### 3. 2. 4. サポートツールの開発

我々はメソッド単位の Web サービスをより簡単に開発するために、既存である Java コードからメソッドサービスを抽出したクラスファイル、Web サービスを実現するエンジンである Apache Axis ヘドプロイするための設定ファイル (.wsdd) を生成するサポートツールを開発した。

設定ファイルとは Apache Axis 上に Java コードを Web サービスとして配備するとき用いるもので、サービスの情報を記述する要素 service と、その子要素 parameter にサービスを提供するクラスの FQCN (完全修飾クラス) や公開するメソッドを記述する。

### 3. 3. リポジトリ登録工程

開発期間の短縮・開発コストの削減が見込められるとしている Web サービスはビジネスアーキテクトが作成した BPM から呼び出される。そのため、ビジネスアーキテクトのための劉殿大きいサービスを示す必要がある。

開発者は連携して構築されたサービスが、どのような Web サービスから構成しているかを把握し、連携する Web サービスの変更に対応しなければなら

ない。しかし BPEL プロセスの記述は困難であるため、変更を支援するツールが必須であると。

以上のことから抽出した Web サービスをまず階層構造で格納する。階層構造にすることで利用者を階層別に分けることができるため、利用者の目的に沿ってどの階層を対象にするかができる。また再利用者が BPEL によって連携されたサービスがどのような Web サービスから構成されているかなど視覚的に Web サービスを把握することができる。

連携する Web サービスを変更できるリファクタリングブラウザを構築した。このブラウザには Web サービスの階層関係を XML でデータ化し、それを読み込むことで既存 Web アプリケーションから抽出された Web サービス群を示し、ブラウザから連携する Web サービスを変更することができ、開発者の再利用を支援する。

#### 4. Web サービス研究適用環境

我々は3章で提案した Web サービスを利用した研究を適用できる『Web サービス研究適用環境』の構築を行っている。これは本研究室の方針におけるものづくりを基調としたもので、Web サービスを利用した研究をより実世界に近い形で評価するといったことを目的としている。また Web サービスの絶対数が少ないという問題点を解決するために、本研究室では構築した Web サービスの外部公開を推進する。この環境は実際にインターネットを通じて利用できる環境となっており、本研究室のサイトから公開されている Web サービスを利用することも可能である。

図6に Web サービス研究適用環境となるシステムの構成を示す。また表 1 に構築基盤ツールを示す。

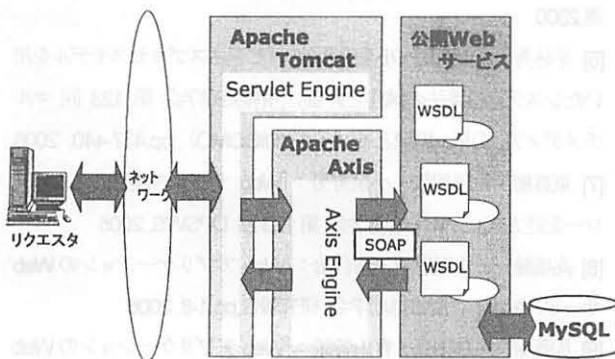


図6 Web サービス研究適用環境のシステム構成

表 1 構築基盤ツール

役割	ツール名
OS	Windows Server 2003
SOAP プロセッサ	Apache Axis1.2
プログラム実行環境	J2SDK1.4.2
Servlet エンジン	Apache Tomcat4.1
DataBase	MySQL4.0

### 5. プロトタイプの実用と評価

#### 5. 1. 対象システム

3 章で述べた方式によるプロトタイプを構築し、その評価を行う。

##### (1) チケット予約システムの概要

本方式によるプロトタイプとなるシステムの構成を図7に示す。

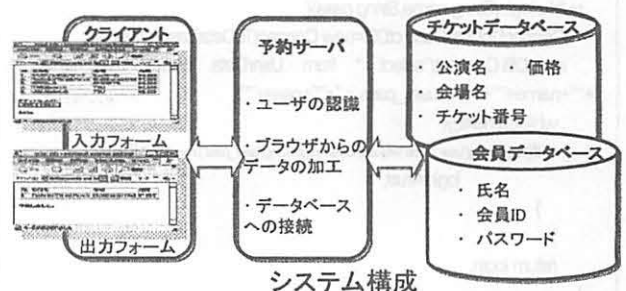


図7 チケット予約システム

このシステムはユーザがブラウザを用いてチケット予約サーバにアクセスし、オンラインでコンサートチケットの予約を行える会員制のシステムである。ユーザは事前に会員登録を行う必要があり、パスワードは会員ごとに発行されている。システムは1つの予約サーバと2つのデータベース（チケットデータ用、会員データ用）を持つ。また、このサービスを行うのは「スズキシステム」という会社の IT 事業部とする。

##### (2) サポートツールの適用

本研究におけるサポートツールで既存のオブジェクト指向プログラムを解析することで、Web サービスになるモジュールコードを生成する。

#### 5. 2. 構築

以下に、本研究のサポートツールによって生成された Web サービスになるモジュールのメソッド名を示す。

- login
- getAllTicket
- catchReserveTicket

login サービスは氏名とパスワードを入力し、認証できれば True を返すサービスである。getAllTicket サービスは現在提供できるチケット情報を取得する

サービスである。この二つは戻り値を持つメソッドである。catchReserveTicket サービスは setTicketInf メソッド, executeTicket Confirmation メソッド, getTicketInf メソッドで構成されていたため、プロキシコードを生成した。

図 8 に生成されたサービスコード (login), 図 9 にサービスプロキシコード (catchReserveTicket) を示す。また図 10 にサービスコード (login) の設定ファイルを示す。

```
import java.sql.*;
class ticket_reservation_system{
String
url="jdbc:mysql://localhost/TicketReservationData?useUnicode=true&characterEncoding=Window2-31J";
String driver="org.gjt.mm.mysql.Driver";
login=false;
...
public login(String name,String pass){
ConnectToDataBase ctdb=new ConnectToDataBase(driver,url);
rs=ctdb.Connect("select * from UserData where user_name = '"+name+"' and user_pass='"+pass+"'");
while(rs.next()){
if(!rs.getString("user_name").equals("")&&!rs.getString("user_pass").equals("")){
login=true;
}
}
return login;
}
}
```

図 8 生成されたサービスモジュール (login)

```
class catchticket_reserve{
public String[] catchTicketInf(String[] num){
ticket_reserve tr=new ticket_reserve();
tr.setTicketInf(num);
tr.executeTicketConfirmation();
return tr.getTicketInf();
}
}
```

図 9 サービスプロキシコード (catchReserveTicket)

```
<deployment xmlns="http://xml.apache.org/axis/wsdd/"
xmlns:java="http://xml.apache.org/axis/wsdd/providers/java">
<service name="Hello" provider="java:RPC">
<parameter name="className" value="ticket_reservation_system">
<parameter name="allowedMethods" value="login"/>
</service>
</deployment>
```

図 10 サービスコード (login) の設定ファイル

以上より、既存の Web アプリケーションから Web サービスを構築することができたといえる。

### 5. 3. 本研究の適用

4. 2. プロトタイプの実用は仮想的なシステムのため、データ内容においても仮定のものである。そこでデータを置き換えて、4 章で述べた公開の環境で評価を行う。実際に Web サービス研究適用環境で

公開する。Web アプリケーションのプロトタイプはチケット予約システムとする。

本研究をプロトタイプに適用したことで生成されたサービスを、実際のデータに置き換えて Web サービス研究適用環境で提供することを検討していく。

## 6. まとめ

本稿では、Web アプリケーションを活用した Web サービスの構築法の提案と、それを実装したサポートツールの開発を行った。またプロトタイプに対して適用評価を行い、Web サービス抽出の有効性を確認中である。SOA で求められるプロセスとなる Web サービスの抽出を自動的に行えることで、既存資産を再活用し、要求に合わせて新たなサービスを提供することができるようになることが狙いである。

今後は、評価を継続すると共に抽出パターンをいっそう充実させることにより、抽出の確実性や、妥当なのかということを検討し、Web サービス研究適用環境にて提供できる Web サービス要素を含む Web アプリケーションシステムを構築し、本提案方式を適用する。さらにまた、リポジトリ登録工程にて Web サービスの検索機能の追加を行っていく。

## 参考文献

- [1] 本 俊也:「詳細 Web サービス構築」,ソフトバンクパブリッシング,2003
- [2] 日本 BEA システムズ株式会社:「SOA サービス指向アーキテクチャ」,SHOEISHA,2006
- [3] XML コンソーシアム:「リアル Web サービス」,秀和システム,2005
- [4] 青木 利晴:「Web サービスコンピューティング」,電子情報通信学会,2005
- [5] 戸田保一・飯島淳一編:「ビジネスプロセスモデリング」,日科技連,2000
- [6] 平林秀一・上西司・小泉寿男他:「ビジネスプロセスモデルを用いたシステム構築法の試作と評価」,情報処理学会 第 123 回 マルチメディア,分散,協調とモバイル (DICOMO), pp.437-440, 2006
- [7] 東真樹・佐藤智哉・小泉寿男:「Web サービス連携のためのフロー記述方式」,情報処理学会 第 123 回 DPSWS, 2005
- [8] 高橋健一・立堀道昭・柴合治:「Web アプリケーションの Web サービス変換」,情報処理学会 研究報告, pp.1-8, 2006
- [9] 加藤勇良・林祐亮・青山幹雄:「Web アプリケーションの Web サービス変換アーキテクチャとその評価」,情報処理学会 第 68 回 全国大会, pp.255-256, 2006
- [10] ThinkIT:「UML の現状」,http://www.thinkit.co.jp/free/compare/12/1/
- [11] IBM:「BPEL4WS2.1」,http://alphaworks.ibm.com/tech/bpws4/