南山大学 数理情報学部 情報通信学科 2004 年度卒業論文要旨集

BPEL 4 WS を用いた複合 Web サービスの研究

2001MT014 福永遂重 2001MT098 高木松雄

指導教員 青山幹雄

1. はじめに

現在, ビジネス環境はますます複雑さを増している. 増殖を続けるビジネスプロセスを柔軟に連携させてビジネスプロセスの自動化が望まれている. この要求を満たす新たなシステム連携技術として Web サービスが注目されている[4][5][6]. 本研究は, SOA (Service-Oriented Architecture) の概念を基に, Web サービスを利用したビジネスプロセスの自動化を目的として, 複合 Web サービスを動的に構築する方法を提案する.

2. Web サービスの問題と解決

2.1. ビジネスプロセスの自動化における問題点

ビジネスプロセスの自動化おいては, (1)サービスを利用する手順の不明確さ, (2) 異なるインタフェース間を動的に連携する困難さ, (3)サービスの意味定義ができない, という問題点が挙げられる.

(1)については, 複数の Web サービスを組み合わせて, 呼出し手順を定義する BPEL4WS(Business Process Execution Language for Web Services, 以下 BPEL と略記)[1]により解決できるので複合 Web サービスとして連携可能である. しかし, BPEL4WS では事前に記述したサービスしか利用できず, (4)サービスの追加変更の場合は, サービス情報を人手で変更しなければならない問題がある.

2.2. 解決案

ビジネスプロセスを連携する場合, 全く異なるサービスを連携させることは考えにくい. また, 問題点(2)について解決するため使用するサービスの機能, 属性の類似部分を関連付けることで, 同時に利用することが可能である. したがって, 類似の機能を持つサービスを連携する必要性が高い. 問題点(3)については現在の技術では困難なため, 機能が類似のサービスを集めてリクエスタに提供する. 問題点(4)については BPEL 文書をサービス選択時に自動生成することにより解決できると考える. 以上をふまえ BPEL を用いてサービスを動的に連携する方法を提案する. この方法を実現するための提案を以下に示す.

(1) 対話型サービス:リクエスタとプロバイダが対話形式でサービスを利用.
(2) 類似サービスの選択:クライアントが選択したサービスを選択サービスリストとして作成.
(3) 選択されたサービスの自動連携:クライアントが選択したサービス情報を取得し, 自動的に連携.
(4) BPEL 文書の自動生成:選択サービスを同時に使用するために並行実行する BPEL プロセスのテンプレートを作成. このテンプレートよりプロセスを生成.

以上の4つの課題を解決した開発の全体像を図 1 に示す.

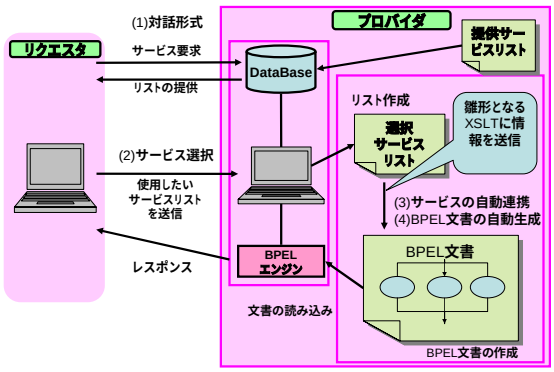

図 1 提案する BPEL 複合 Web サービス

3. BPEL 複合 Web サービス作成方法

本研究では, プロバイダ側の BPEL プロセスの自動生成に特化し開発する. BPEL プロセスの自動生成を実現するための提案方法を以下に示す.

(1)類似サービス間のデータ構造, 操作の対応付け.
(2)使用サービスの WSDL からのデータマッピング.
(3)選択サービスリストを生成.
(4)並行実行用 XSLT スタイルシート作成.
(5)XSLT プロセッサによる BPEL プロセス生成.

これらの提案方法を検討し, プロトタイプを開発する.

3.1. 類似サービスの対応付け

意味的に類似したサービスを対応付けするために各サービスの WSDL の type, message, operation に新たに属性を付加する. 同様に BPEL プロセス側の WSDL にも属性を付加する. これによって各サービスの機能ごとにサービスデータ構造と操作を対応づける[3].

3.2. サービスリストの作成

各サービスで関連付けた WSDL より, BPEL プロセスを

作成するためのデータリストを生成する．このデータリストは後述する XSLT の変換元になる XML インスタンスにあたる．

3.2.1. サービスリストデータ構造へのマッピング方法

本研究では BPEL プロセスを生成するために選択サービスリスト生成と，プロセス側の WSDL，選択サービスの WSDL を生成しなくてはならない．また選択されたサービスによって BPEL プロセスとプロセスの WSDL の内容も変更してリクエスタに提供しなくてはならない．この変更をする度に WSDL を読み込み，変更するための値を取得する処理が必要である．しかしサービスリストの生成，プロセスの WSDL の変更といった WSDL を読み込む処理を何回も行うことは効率的ではない．そこで WSDL の構造とほぼ同型な木構造へのマッピング[4]をすることを提案する．1 度マッピングを行い Java のオブジェクト化することでデータへのアクセスを容易に行うことができ，同じ処理の繰り返しを回避できる．図 2 に木構造へのマッピングを示す．

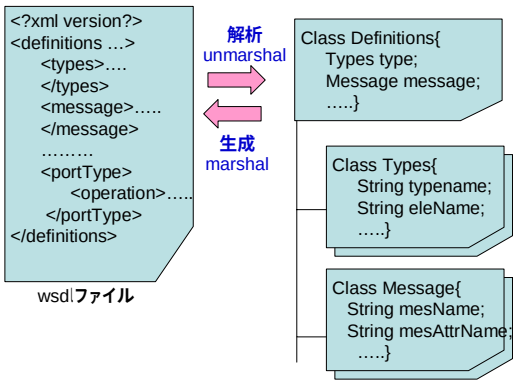

図 2 木構造へのマッピング

3.2.2. サービスリストのモデル

サービスリストの XML インスタンスを作成するために，XML インスタンスのモデルを考える．BPEL プロセスに必要な要素と出現回数を表すクラス図を図 3 に示す．

図 3 サービスリストのクラス図

BPEL プロセスに必要なデータだけで構成されており，並行実行用の XSLT スタイルシートが各サービスのデータにアクセスしやすいデータ構造とした．

3.3. BPEL プロセスの作成

サービスリストの XML インスタンスから BPEL プロセスを生成するための XSLT スタイルシートを用いる．
選択サービスにより動的に要素が組み込まれる時の処理として，BPEL プロセスの root 要素 process の子要素 <partnerLink>, <variables>, また main の<sequence>の中で処理される，各サービスに対する変数の初期化を行う <assign>， 並行実行を行う<flow>の要素に対してはそれぞれ組み込むテンプレートを作成することで動的に値を組み込めるようにする．template の呼び出し方法を図 4 に示す．

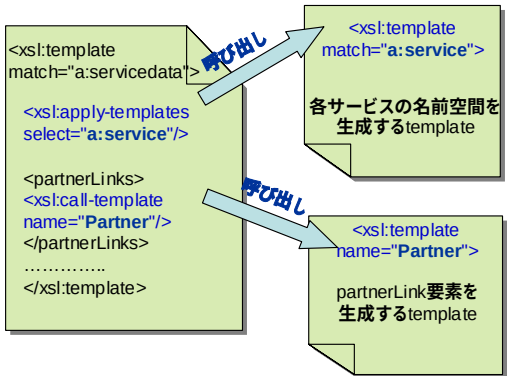

図 4 template 呼び出し図

3.4. BPEL 生成 ProcessController の作成

3.4.1. 開発目標

使用サービスリストから，動的に BPEL プロセスを生成する ProcessController を開発する．ここで，動的とは ProccesController が使用サービスの WSDL よりデータのマッピングを行い，サービスリスト生成，BPEL プロセスの生成までの処理を実行することを指す．

3.4.2. ProcessController のアーキテクチャ

選択サービスの情報を取得し，BPEL プロセス生成まで行う ProcessController のアーキテクチャを図 5 に示す．

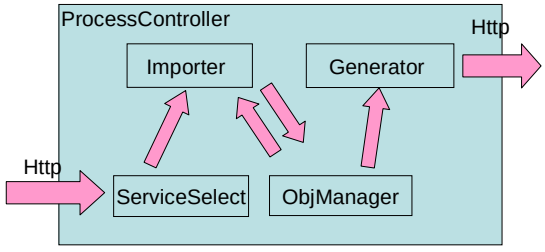

図 5 ProcessController のアーキテクチャ

ProcessControllerを構成する主要な4つのクラスとして外部ﾘｿｰｽを提供するServiceSelectクラス, 外部ﾘｿｰｽからモデルに登録するImporterクラス, モデルとしてデータオブジェｸﾄの管理を行うObjManagerクラス, モデルよりプロセス生成を行うGeneratorクラスを作成した.

ProcessControllerの制御であるBPELプロセス生成までのシーケンス図を図6に示す.

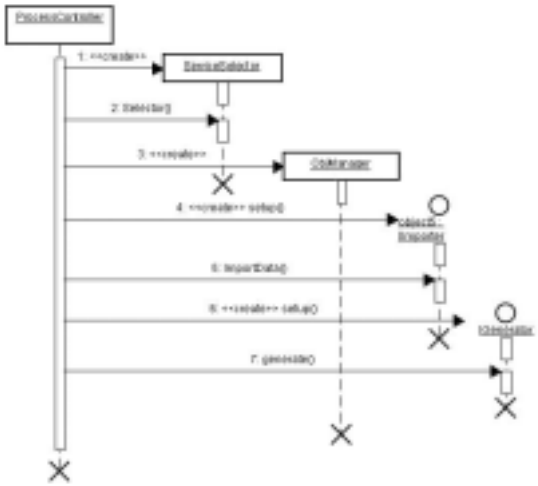

図6 BPEL **プロセス生成のシーケンス図**

3.5. BPEL **プロセス生成** ProcessController **の実装**

実行可能なBPELプロセスを提供するための生成物を挙げて, ProcessControllerで生成するように実装する. これらのクラスは全て, Generatorの共通インタフェースを定義interfaceであるIGeneratorにアドオンする.

表1 IGeneratorにアドオンするクラス

必要な生成物	クラス名
サービスﾘｽﾄ	ServiceListGeneratorクラス
BPELプロセス	BpelGeneratorクラス
使用するWSDL	WSDLGeneratorクラス

4. プロトタイプの開発

4.1. 単語張サービス

本研究ではBPELプロセスで扱う類似サービスとして単語帳サービスを使用する. 使用するサービスとして@ITのICD, 三省堂のNetDic, 我々が作成したSimpleサービスの三つとする[2].

(1) 各サービスのデータ構造の対応

三つのWebサービスを連携させるため, 三つのサービスのデータの構造の対応関係を図7に示す.

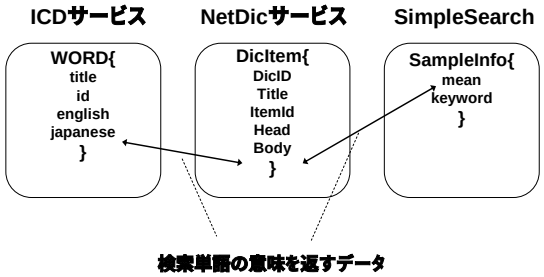

図7 各サービスのデータ構造の対応

(2) サービスのﾒｿｯﾄﾞの対応

三つのWebサービスのﾒｿｯﾄﾞの対応関係を図8に示す.

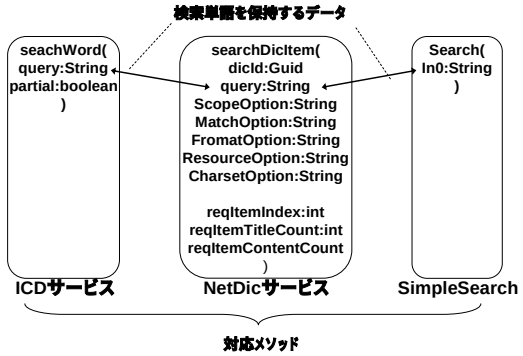

図8 ﾒｿｯﾄﾞの対応

(3) 類似サービスの対応の実装

WSDLの記述に新たに属性を追加することはWSDLの仕様で可能である. 追加する属性はXMLスキーマによって定義する. 作成するスキーマは, 属性名の定義とその属性値の定義になる.

5. 評価

5.1. サービスの対応付け

BPELプロセスに各サービスのWSDLを読み込ませ, 共通する機能のデータ構造と操作を各サービスのWSDLとBPELプロセスのWSDLのoperation, message, typeに属性を用いて対応付けた. これによりサービス間に対応関係を持たせ, BPELプロセスで扱うデータを指定できることを確認した.

5.2. BPEL **プロセスの自動生成**

本研究では, プロバイダはﾘｸｴｽﾀに結果が理解できるように類似のサービスを提供し, それらのサービスを同時に使用できるように並行実行するプロセスを生成した. 複数のサービス結果を同時に得るビジネスプロセスを自動生成はプロトタイプにより確認できた. 本研究の目的であるBPELプロセスの自動生成は実現できたといえる.

5.3. BPEL プロセス生成評価

本研究で作成した ProcessController は，WSDL を一度解析しオブジェクト化することでデータ解析部の処理を省略できる．サービス情報を記述し，ObjManager クラスにサービスのデータオブジェクトを登録することで，サービスの追加が容易にでき，拡張性が高いといえる．

次に BPEL プロセス生成において，DOM を用いた場合と，サービスリスト作成と XSLT により生成した場合の二つを，生成コード行数により比較したものを表 2 と図 9 に示す．

表 2 BPEL プロセスのコード行数の違い

生成方法	コード行数
DOM	553 行+約 140*サービス数
選択リスト+XSLT	1163 行+約 50*サービス数

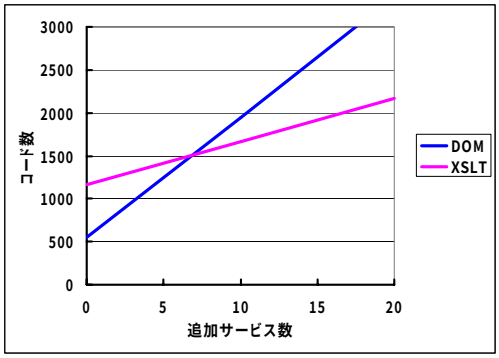

図 9 追加サービス数によるコード数の変化

BPEL プロセスの生成は DOM がサービスリストと XSLT を用いた場合の半分程度のコードですむ．しかし，選択サービスの追加を行うのに必要なコード数は選択サービスリストと XSLT を用いた方が DOM の時の半分以下である．このように，DOM で生成するよりも選択サービスリストと XSLT を用いた生成の方がサービスの追加が効率的となる．

本研究では並行実行のプロセスしか生成しないので，DOM で生成する BPEL プロセスでも対応できる．しかし，DOM で生成する時にサービス順序変更などの動的な構造の変換が行われる場合，一度生成した DOM 木からの構造変換が困難になり，XSLT を用いて構造を変換し直す必要がある．構造を柔軟に変更できる XSLT による BPEL プロセスの生成は効率的であるといえる．

6. 考察と今後の課題

6.1. サービスの対応付け

研究より単語帳サービス以外にも，類似の機能を持ったサービスの対応が行える．しかし，開発者が類似な機能を判断し，対応を考えなければならない．したがって，開発コストの増大も考慮する必要がある．

6.2. BPEL プロセス

作成した BPEL プロセス生成プログラムでは，提供できるサービス内ではプロセスを自動生成が可能ではあるが，新規サービスを追加するには，プロバイダ側が静的に行わなければならない．サービスを動的に追加することが今後の課題として挙げられる．また，作成した ProcessController の Web サービス化を行い，BPEL プロセスを提供することを今後の課題とする．プロバイダがリクエスタに提供サービスを公開し，リクエスタがプロバイダに選択サービスを送信するための XML フォーマットを提案し，有用性を評価することで実現できると考える．

BPEL プロセスの考察として，本研究では BPEL プロセス側がクライアントに対してリクエスト，レスポンスの各機能に portType を分けて考えた．しかし単語帳サービス以外の類似サービスを追加した場合，提供された側はどの portType が対応しているかが理解しにくいので，類似サービスに対して 1 つの portType を定義し，バリエーションの少ないインタフェースを提供することが望ましいと考える．

7. まとめ

本研究では，ビジネスプロセスの自動化を実現するため，BPEL を用いた複合 Web サービスの構築を提案した．各サービスの WSDL の拡張により類似サービスの対応を行い，インタフェース間でデータ構造と操作を対応させた．実験結果よりサービスの選択だけで BPEL プロセスと，使用するサービスの WSDL ファイルを自動生成することを確認した．DOM による BPEL プロセス生成方法と比較して，XSLT スタイルシートを用いた方法だと変更容易性が高いと考え，選択サービスリストを用いた方法を提案した．実験結果，実現方法の比較により BPEL プロセス生成プログラムの有用性の評価を行った．

参考文献

[1] IBM Web Services，ビジネスプロセス BPEL4WS について，http://www-6.ibm.com/jp/developerworks/webservices/021025/j_ws-bpelcol1.html#1，2002.

[2] 中村一仁ほか，価値に基づく Web サービスの動的連携ブローカとその評価，情報処理学会ソフトウェア工学研究会，Vol. 2004-SE-144, Mar. 2004，pp.123-130.

[3] 丸吉祐也，秋田太陽，Web サービスにおけるリクエスタプログラムの再利用に関する研究，2003 年度卒業論文，2004.

[4] 丸山 宏他，XML と Java による Web アプリケーション開発 第 2 版，ピアソン・エディケーション，1999.

[5] 本 俊也，Web サービス構築，ソフトバンク，2003.

[6] 本 俊也，最新 Web サービスマスタリングハンドブック，秀和システム，2004.