南山大学 大学院 数理情報研究科 2006 年度修士論文 要旨集

複合 Web サービスのモデル駆動開発方法の提案

M2005MM003 福永 遂重

指導教員 青山 幹雄

1. はじめに

ユーザ要求の多様化に対し，最適な Web サービスを実行時に連携する必要がある．現在の Web サービス技術は，ユーザの非機能要求による連携に対応していない．本研究ではユーザの機能，非機能要求を満たす複合 Web サービスのモデル駆動開発方法を提案する．機能，非機能特性が表現可能な Web サービスのインタフェースモデルを定義する．モデルから BPEL を自動生成する複合 Web サービス開発支援環境のプロトタイプを開発し，有用性を示す．

2. 動的な複合 Web サービス構築の問題点

2.1. サービス選択問題

システムが自動的にサービス選択を行うには，選択基準となるサービスの機能・非機能特性を表現し，その基準に基づきシステムがサービスを決定する方法が必要である．しかし，現在の Web サービス技術では機能特性は UDDI のカテゴリ情報や，WSDL のインタフェース情報などで定義可能であるが，非機能特性を定義することは困難である．よって，ユーザ要求を満たすサービス選択を自動的に行うには，機能・非機能特性の両方を扱う必要がある．そこで機能・非機能特性をモデル化し，モデルに基づくサービス選択をする仕組みが必要である．

2.2. サービスの自動連携問題

現在，複数の Web サービスを組み合わせる仕様として BPEL などが挙げられる．しかしこれらの仕様はあらかじめ連携する Web サービスの実行順序を決定する必要があるため，自動的にサービス選択・連携を行うことは困難である．自動的なサービス選択・連携を実現するために，実行時にサービスのインタフェースを抽出し，サービス間のインタフェースを実行可能な形式に変換する必要がある．そこで Web サービスのレベルより高い抽象度でサービスの機能とその入力，出力のモデル化が必要である．

3. 関連研究

モデルに基づき Web サービスを組み合わせる研究が行われている．文献[2]では，QoS 特性を付加した Web サービスのインタフェースを定義している．Web サービスの QoS 特性は UML Profile で定義されている QoS カタログと Schedulability，Performance and Time の UML Profile に基づく記述方法により定義されている．

文献[1]では，抽象的なオペレーションとオペレーションの状態遷移をモデル化した対話モデルが提案されている．このモデルに基づき BPEL の変換ルールからスケルトンを自動生成する方法も提案している．しかしこれらの研究は Web サービスの動的な選択，連携には対応していない．

4. 複合 Web サービスの開発方法論

4.1. モデル駆動開発(Model-Driven Development)

現在のシステム開発ではシステム要件の複雑化，実装技術の多様化に対して迅速な対応が求められている．そこで，モデル中心のシステム開発アプローチである MDA (Model-Driven Architecture)が提唱されている．MDA ではビジネスプロセスのようなプラットフォームに依存しない PIM (Platform-Independent Model)と実装環境のようなプラットフォームに依存する PSM (Platform-Specific Model)のモデルを定義している．さらに，PIM から PSM への変換や PSM からソースコードへの変換を自動化，または半自動化にするモデル駆動開発手法が提案されている．

4.2. 複合 Web サービスにおけるモデル駆動開発方法

本研究では，Web サービスのモデル定義とその開発プロセス[3]を提案する．モデルから複合 Web サービスを自動生成するモデル駆動開発方法を提案する．図 1に示すモデル駆動 Web サービス開発プロセスでは PIM と PSM を定義し，このモデルから複合 Web サービスを自動生成する．

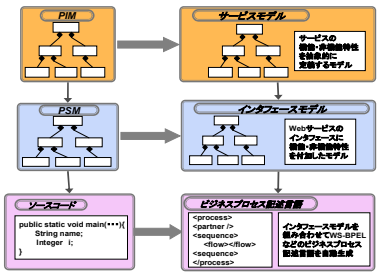


図 1 モデル駆動 Web サービス開発プロセス

Web サービス開発における PIM は，Web サービスに依存せずに，高い抽象度でサービスの機能・非機能特性を定義する．以後，このモデルをサービスモデルと呼ぶ．PSM は Web サービスのインタフェース (例:WSDL など)に準拠し，機能・非機能特性を付加したモデルを考える．以後，このモデルをインタフェースモデルと呼ぶ．

4.3. モデル駆動開発プロセスの提案

図 2 に示す複合 Web サービスのモデル開発プロセスでは，モデル定義とサービス実行の 2 段階で実行する．

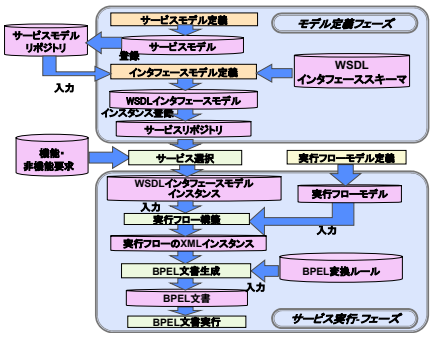


図 2 モデル駆動開発プロセス

4.3.1. モデル定義フェーズ

モデル定義フェーズでは，サービスモデル定義とサービスインタフェースモデル定義の 2 段階で実行する．サービスモデル定義では，開発者がサービスメタモデルに基づき XML Schema で記述されるサービスモデルを定義し，サービスモデルリポジトリに登録する．インタフェースモデル定義フェーズでは，開発者がサービスモデルリポジトリの中のサービスモデルと Web サービスのインタフェーススキーマを入力し，XML Schema で記述されるインタフェースモデルを生成する．本研究では，Web サービスのインタフェーススキーマに WSDL の XML Schema を適用し WSDL インタフェースモデルを生成する．以後，このスキーマを WSDL インタフェーススキーマと呼ぶ．そして生成した WSDL インタフェースモデルからインスタンスを開発しサービスリポジトリに登録する．

4.3.2. サービス実行フェーズ

サービス実行フェーズでは，Web サービスの実行順序を定義する実行フロー構築と，ビジネスプロセス記述言語 BPEL を用いて BPEL 生成，BPEL 実行の 3 段階で実行する．実行フロー構築フェーズは開発者が定義した実行フローモデルと，ユーザ要求によって選択された WSDL インタフェースモデルのインスタンスを組み合わせて実行フローの XML 文書を生成する．この実行フローにより様々なビジネスプロセス記述言語にも対応可能となる．BPEL 生成フェーズは，実行フローの XML 文書と BPEL 変換ルールを入力して BPEL と複合 Web サービスの WSDL を自動生成する．最後に BPEL エンジンを用いて BPEL を実行する．

5. サービスの機能･非機能モデル

5.1. サービスモデルの概念

本研究では，サービスは提供する機能とサービスがユーザに提供する商品や情報などのコンテンツから構成されるものと考えモデルを定義する．モデルは XML で記述する．さらにモデル開発，利用コストを考慮するために，UML のような視覚的に構造が理解しやすい図式表現を用いたものが必要であると考える．そこでサービスモデルはプロバイダの信頼性などのプロパティと，提供する機能，提供するコンテンツの 3 つの要素を UML のクラス図を拡張した図式表現を用いて集約関係で定義する．

5.2. サービスモデル

サービスモデルは抽象的な機能と，サービスが提供するコンテンツ，サービスプロパティを定義する．図 3にサービスメタモデルを示す．

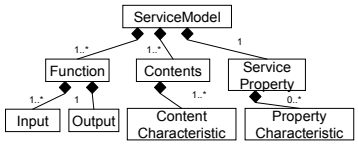


図 3 サービスメタモデル

サービスメタモデルはルートに 1 つの ServiceModel 要素を持つ．その子要素には 1 つ以上の FunctionCharacteristic である Function 要素，ContentsCharacteristic である Contents 要素，PropertyCharacteristic である ServiceProperty 要素の 3 つの要素を持つ．FunctionCharacteristic には複数の FunctionDemeintion，ContentsCharacteristic には複数の ContrentsDemention 要素を持つ．サービスモデルでは要素間を全て集約関係で定義する．

5.3. WSDL インタフェースモデル

サービスモデルと WSDL インタフェーススキーマを入力し，WSDL インタフェースモデルを出力する．そこで，WSDL インタフェースモデルを生成するために構造，属性，制約を定義する．また，WSDL インタフェースモデルは，UML のクラス図を拡張した図式表現を用いて定義する．

5.3.1. WSDL インタフェースモデルの構造

図 4に示す WSDL インタフェースモデルの構造はルート要素に WSDL の Definition を定義し，子要素に Types，Service，interface を持つ．Service 要素の子要素にはサービスモデルのルート ServiceModel 要素を持ち，サービスモデルの ServiceModel 要素以下の要素はサービスモデルと同様の木構造で変換する．

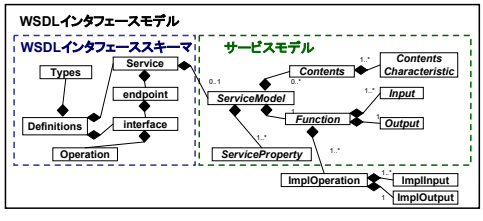


図 4 WSDL インタフェースモデルの構造

Function 要素の子要素には実行するオペレーションを記述する ImplOperation 要素を持ち，その子要素には引数，返り値を定義する ImplInput 要素と ImplOut 要素を持つ．

5.3.2. WSDL インタフェースメタモデル要素の属性定義

WSDL インタフェースモデルの各要素がもつ属性を定義する. WSDL インタフェーススキーマで定義されている要素には WSDL で定義されている属性を利用する. サービスモデルの要素には UML Profile の SPT Profile で定義されている記述方法を適用する. 表 1に定義する属性を示す.

表 1 サービスモデルの要素の属性

属性名	パラメータ
value	Characteristic, Dimension の値
unit	Characteristic, Dimension の単位
source	required,assumed,predicted,measured,undeifned
type	maximum,minimum,mean,variance
direction	increasing, decreasing, undefined

5.3.3. WSDL インタフェースモデル要素の制約定義

WSDL インタフェースモデル要素に記述するステレオタイプ, タグの定義を行う. ステレオタイプ<<sequence>>は実装する Web サービスのオペレーションの実行順序を定義する. <<CWSInput>>は変数の値のコピー先を指定する. <<CWSOutput>>は変数の値にコピー元を指定する. <<extends>>は ServiceModel 要素のみ適用し, 継承しているサービスモデルの URI を記述する. <<attribute>>は入出力に付加的な意味定義が可能になる. 表 2, 表 3に定義した制約を示す.

表 2 ステレオタイプの定義

ステレオタイプ	WSDL インタフェースモデル要素
<<sequence>>	FunctionCharacteristic
<<CWSInput>>	Input
<<CWSOutput>>	Output
<<extends>>	ServiceModel
<<attribute>>	Input, Output

表 3 タグと式の記述方法

ステレオタイプ	タグ	式
<<sequence>>	ope	ope=ope1:ope2
<<CWSInput>><<CWSOutput>>	var	var="var"
<<extends>>	namespace	namespace="uri"
<<attribute>>	attr	attr="attribute"

6. WSDL インタフェースモデルによる Web サービスの構築

実行フェーズでは, WSDL インタフェースモデルのインスタンスを入力し, ビジネスプロセス記述言語の自動生成を行う. WSDL インタフェースモデルのインスタンスから BPEL 生成までの詳細プロセスを図 5に示す. WSDL インタフェースモデルの XML 文書を用いて Web サービスの実行順序を規定する実行フローを構築する. そこで実行フローの XSLT スタイルシートである実行フローモデルを用いて, WSDL インタフェースモデルの XML 文書から実行フローXML 文書を生成する. 最後に実行フローXML 文書と XSLT スタイルシートで定義する BPEL 変換ルールを用いて BPEL を生成する.

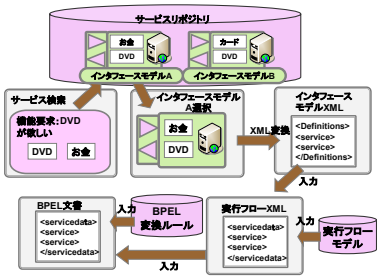


図 5 実行フェーズの詳細プロセス

7. プロトタイプの実現

7.1. ServiceModeler の実現

図式表現によるモデル編集機能と図形モデルから XML への変換機能を持つ ServiceModeler を実装した. グラフィカルなエディタを実現するために GEF(Graphical Editing Framework)を利用した. 図 6に要素の図形を表現するコンポーネントモデルクラスと要素間を結ぶコネクタモデルクラスを示す. 開発規模はクラス数 46, コード数 2236 行となった.

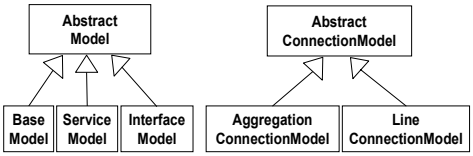


図 6 コンポーネントモデルとコネクタモデルのクラス図

7.2. ProcessContorller の実現

サービス実行フェーズを自動化するために ProcessContorller を実装した. 図 7に ProcessContorller のアーキテクチャを示す. 開発規模はクラス数 97 個, コード数 5004 行となった. ServiceSelect クラスが WSDL インタフェースモデルを選択する. 次に ObjManager クラスによって WSDL インタフェースモデルの XML 文書から Java のデータモデルにマッピングする. 選択したサービスのデータモデルを Importer クラスに渡し, Generator クラスによって複合 Web サービスを自動生成する.

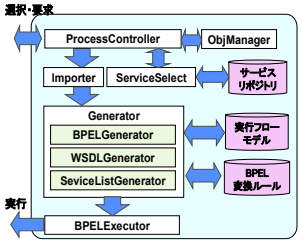


図 7 ProcessContorller のアーキテクチャ

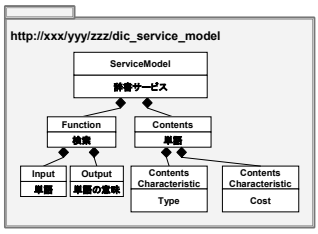
7.3. 例題を用いた評価

本研究では, BPEL で扱う類似サービスの例として表 4に示す 3 つの辞書サービスを利用する. また, 図 8に 3 つの辞書サービスにおけるサービスモデルの定義を示す.

表 4 利用するサービスの概要

サービス名	特徴 / URL
ICD	コンピュータ用語事典
http://www.iwebmethod.net/icd1.0/icd.asmx	
NetDic	英和・和英・国語辞典サービス
http://btonic.est.co.jp/NetDic/NetDicv09.asmx	
SimpleSearch	本論文で作成したサービス
http://app.nise.org/axis/services/SimpleSearchService	

単語を入力し単語の意味を返す FunctionCharacteristic”検索”と, コンテンツ”単語”を定義した. コンテンツ”単語”ではサービスが扱う単語の種類 Type, サービスを利用するための料金 Cost の 2 つを定義した.



辞書サービスモデルに基づき, 図 9のように3つの辞書サービスの WSDL インタフェースモデルを定義する.

図 8 辞書サービスのサービスモデル

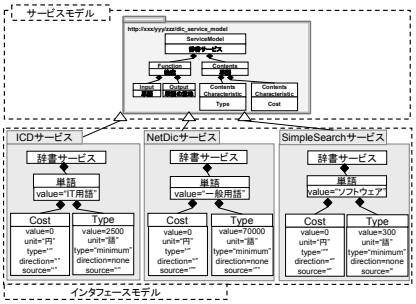


図 9 3つの辞書サービスの WSDL インタフェースモデル

この WSDL インタフェースモデルに基づき ProcessController を用いて非機能特性によるサービス選択を行った. リクエスタは単に辞書サービスを使用したい場合は, サービスモデルを用いて”FunctionCharacteristic=検索, Input=単語, Output=単語の意味”を記述し ProcessController に要求する. そして条件の満たすサービスモデルを実装している WSDL インタフェースモデルを検索し該当する 3 つのサービスを提供する. また, リクエスタが”IT 用語”の辞書サービスを利用したい場合は, サービスモデルを用いて”FunctionCharacteristic=検索, Input=単語, Output=単語の意味”を記述し, ContentsCharacteristic に”IT 用語”を記述し ProcessController に要求する. そして条件の満たすサービスモデルを実装している WSDL インタフェースモデルを検索し, 該当するICDサービスを提供する. よってサービスモデルで機能と入出力を抽象的に定義することで, サービスの機能的特性によりサービスの絞込みが可能となる. また, サービスモデルのコンテンツを定義することで, 非機能特性によるサービス選択が可能となる. さらに ProcessController にサービスモデルを用いて要求を与えることで, リクエスタは利用するサービスを意識しなくてもよい. 結果より定義したサービスモデルと WSDL インタフェースモデルを類似機能の Web サービスに適用することでユーザの要求に対応したサービス選択が可能となり, プロトタイプの有用性が評価できた.

8. 考察

本研究では, サービスは提供する機能とユーザに提供する商品や情報などのコンテンツから構成していると捉えモデル化を行った. 抽象度の高いモデルから Web サービスレベルのモデル定義により, 機能・非機能特性を付加した Web サービスのインタフェースの表現が可能になり, このモデルから並行実行する複合Webサービスを実現した.

9. 今後の課題

サービスの実行時間や応答時間, セキュリティなどの QoS 特性に対応していない. このような QoS 特性によるサービス選択も考えられるため, QoS 特性も表現可能なモデルを提案する必要がある. また, サービスの機能特性として入出力のデータの型を規定する方法は提案したが, 適切なデータ型の Web サービスを選択・連携する仕組みは考慮していない. ユーザ要求を満たし, 実行可能を保証するサービス選択・連携を実現する仕組みが必要である.

10. まとめ

本研究の目的はユーザ要求を満たす複合 Web サービス構築することである. そのためにはユーザ要求からサービス選択・連携を動的に可能にすることが必要である. そこで本研究では MDA の概念を取り入れ, 動的に Web サービスを選択, 連携を可能にするために機能・非機能的特性をモデル化し, そのモデルから駆動する複合 Web サービス開発方法を提案した. また, サービスモデルを記述するためのモデル定義エディタを Eclipse プラグインとして開発し, 開発したモデルから複合 Web サービスを自動生成するプロトタイプを開発し, その有用性を評価した.

参考文献

[1] K. Baina, et al., Model-Driven Web Service Development, *Proc. of CAiSE 2004*, Jun. 2004, pp. 290-306.

[2] A. D’Ambrogio, et al., A Model-Driven WSDL Extension for Describing the QoS of Web Services, *Proc. of ICWS* 2006, Sep. 2006, pp. 789-796.

[3] 福永 遂重, 青山 幹雄, 複合 Web サービスのモデル駆動開発方法と支援環境, SES2006 論文集, Oct. 2006, pp. 221-222.