価値モデルに基づく Web サービスの動的選択と価値保証

M2004MM031　中村 一仁

指導教員 青山 幹雄

1. はじめに

SOA(Service-Oriented Architecture)は，ソフトウェアをサービスとして定義し，大規模システムを柔軟に構築できる技術として注目されている．

しかし，現在の技術では，実行時に柔軟にサービスを組み替えることは困難である．ユーザにとって最適なサービスを自動的に連携し，品質保証を行う技術が必要である．

本稿では，高い抽象度でサービスを定義する VSDL (Value-Based Service Description Language)を提案し，価値ブローカを用いたサービスの動的連携の方法を提案する．また，プロトタイプを用いた評価結果を報告する．

2. Web サービスの提供モデル

本研究で考えるサービスの提供モデルを図1に示す．リクエスタとプロバイダ間を四段階でモデル化する．コンテンツとはサービスを利用した結果，入手可能な商品や情報である．プロビジョニングは Web サービスで実現されたサービスの提供手段である．

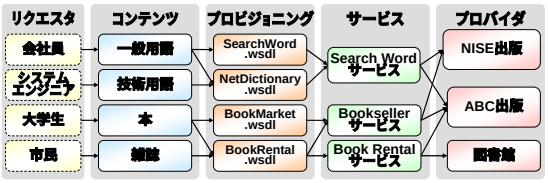

図 1　Web サービスの提供モデル

従来の Web サービス技術では，サービス選択はプロビジョニングの機能特性に基づいている．しかし、サービスを利用時に連携(動的連携と定義)する場合には，コンピュータがすべての段階を総合的に評価する必要がある．

3. Web サービスの動的連携の問題点

Web サービスをサービスブローカを用いて動的に連携し利用するには以下の問題が挙げられる．

3.1. 非機能特性によるサービス選択の必要性

Web サービスの動的連携には，リクエスタの要求するサービスを発見し連携する基準が必要である．この要求基準には，機能特性と非機能特性の二つがある．機能特性は，サービスのインタフェースや UDDI により定義できる．一方，非機能特性はサービスのプロビジョニングやプロバイダの属性などに依存し多様であると考えられる．非機能特性の定義は現在の技術では困難である．そこで，非機能特性をモデル化する新たな方法と，リクエスタの要求するサービスを自動的に選択し，利用する仕組みが必要である．

3.2. 高い抽象度でのサービス記述の必要性

SOA では，要求分析工程でビジネスプロセス導出し，それを実現するサービスを組み合わせてシステムを構築する．しかし，WSDL によるインタフェース記述は，ビジネスプロセスと比べ抽象度が低い．サービスの動的連携の場合には，サービスブローカがサービスを組み合わせるため，抽象度の違いを吸収し，ビジネスプロセスからサービス抽出を行う技術が必要である．

3.3. Web サービスの品質保証の必要性

サービスブローカが動的連携したサービスをビジネスで利用するには，図1の Web サービスの提供モデルに従って，コンテンツ，プロビジョニング，サービス，プロバイダの4つの要因を総合的に評価する仕組みが必要である．

4. 価値に基づく Web サービスの動的連携

4.1. 価値交換によるサービス連携アプローチ

サービスの入出力データを価値として抽象化すると，サービスは入力として要求した価値を出力として提供する価値に変換する活動として定義できる．そこで，サービスリクエスタとサービスプロバイダ間のメッセージのやり取りを価値交換として定義し，サービス連携を価値交換の連鎖としてモデル化する(図 2).

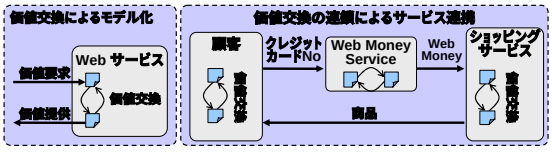

図 2　価値交換によるサービス定義とサービス連携

4.2. 価値モデルと価値ブローカ

価値交換によるサービス連携の実現には，サービスを総合的に評価する基準が必要である．サービスを包括的に価値として定義し，価値に基づくサービスの動的連携と品質保証を実現する価値モデルと価値ブローカを提案する．

(1) 価値モデル：サービスに入出力される価値をモデル化する．機能特性と非機能特性を総合的に評価する．

(2) 価値ブローカ：価値交換の連鎖によるサービスの動的連

携と価値保証を行う仲介サービスを提供する.

4.3. 価値に基づく Web サービスの利用の流れ

価値に基づく Web サービスの利用は登録フェーズと利用フェーズに分類でき, 利用の流れは図 3 のようになる.

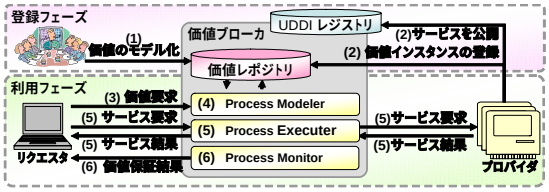

図 3 価値モデルのモデル化と利用

1)サービスの価値のモデル化: 価値はアプリケーションドメインやプロバイダの業種により異なる. そこで, 価値のスコープを名前空間として定義し, スコープ内で有効な価値を価値要素として抽出し, 価値モデルの構造や意味を定義する. 価値モデルは価値レポジトリに登録する.

2)サービス定義と価値モデルのインスタンス化: プロバイダは次の手順でサービスを公開する. 1)サービスのインタフェースを WSDL で定義し UDDI へ登録する. 2)サービスの価値を定義し価値レポジトリへ登録する.

3)価値要求の作成: リクエスタは価値モデルに基づきリクエスタのコンテキストに応じた価値要求を作成し, 価値ブローカに送信する.

4)複合サービスの生成と提供: 価値ブローカは価値レポジトリと UDDI レジストリを検索し最適なサービスを選択し, 複合サービスを生成してリクエスタに提供する.

5)サービス要求の生成と実行: リクエスタはサービス要求を生成し価値ブローカに送信する. ブローカは要素サービスを起動するためプロバイダにサービス要求を行い, サービス結果を統合してリクエスタに返す.

6)サービスの価値保証: 価値ブローカはリクエスタとプロバイダ間で交換されるメッセージを監視し価値を評価する. これによりリクエスタに対して価値を保証できる.

5. 価値モデル

価値によるサービス記述の実現のため, サービスに入出力される価値を価値モデルとして定義した. 価値を定義するためのフレームワークである価値メタモデルを定義し, 木構造による価値のモデル化を行った. 価値モデルは, インスタンス化され価値要求と価値提供として価値によるサービス記述に使用される.

5.1. 価値のメタモデル

価値と価値の利用に影響を与える要因を定義する価値メタモデルの定義を図 4 に示す. 価値モデルは, サービスの総合評価のため, 図1に示す Web サービスの提供モデルに従い 4 つに分類する. サービス選択に有効な価値は, アプリケーションドメインなどに異なると考えられるため, スコープ毎に価値モデルを定義する. 価値モデル内は, サービス選択に有効な品質属性を価値要素として抽出し, 価値要素間の関係を記述することで価値モデルを定義する. 価値モデルを評価するリクエスタやブローカなどのアクタは, サービスの利用時間や利用場所などのコンテキストを持っており, 評価基準はコンテキストに依存すると考える.

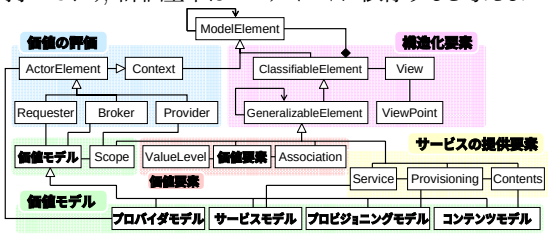

図 4 価値メタモデル

5.2. 価値モデル

価値モデルは価値要素とその間の集約関係を木構造で定義し, UML のクラス図を拡張して記述する方法を提案する. 商社の価値モデルの例を図 5 に示す.

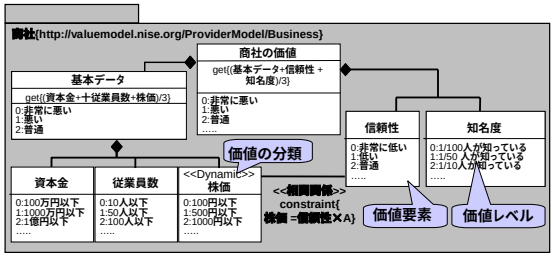

図 5 価値モデルの例

価値モデルのスコープはパッケージで表現し, 価値要素はクラスとして表現する. ある価値要素はより詳細な特性を集めたものとして考えられるため, 価値モデルの木構造を集約関係で表現する. 価値要素は, 価値レベルとして品質属性の尺度を持ち, 分類はステレオタイプで記述する. また, 木構造の節点となる価値要素は操作「get()」を持ち, 下位の価値要素の価値レベルを評価して返す.

価値モデルの XML 記述として VM-XML(Value Model-XML)を提案する. VM-XML で記述された価値モデルは価値ブローカ内の価値レポジトリに, 価値要求と価値提供のテンプレートとして格納される.

価値モデル記述を行うツールを Eclipse プラグインとして開発した. 開発規模は, 76 クラス 7,962 行である.

5.3. 価値提供と価値要求

サービスのリクエスタとプロバイダは価値モデルのテンプレートを用いて, 価値提供と価値要求を定義する. 価値提供とは, サービスから出力される価値であり, サービス結果や他のサービスへの入力として利用される. 一方, 価値要求とは, サービスへ提供される価値に対する条件記述であり, 価値モデルで定義された価値レベルに対して要求す

る価値の条件を付加し定義する. 価値要求の条件として以下の二つの水準を定義した.

1) **当たり前価値要求**:指定した条件を必ず満たさなければならない価値の条件. 条件を満たさないサービスは利用サービスの候補から除外される.
2) **魅力的価値要求**:条件を必ず満たす必要はないが, サービスの利用順位に影響する.

図 5 の価値モデルに従って定義された価値提供と価値要求の例を図 6 に示す.

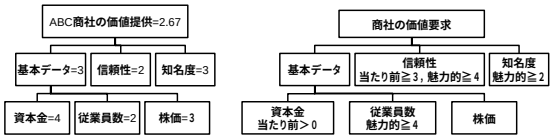

図 6 価値提供と価値要求の例

6. 価値交換によるサービスの動的連携

6.1. 価値に基づく Web サービス記述(VSDL)

価値交換の連鎖としてサービスの動的連携を実現するため, 価値に基づく Web サービス記述言語として VSDL (Value-Based Service Description Language)を提案する. e³value Model[1]の表記法を拡張した VSDL の図的表現を図 7 に示す.

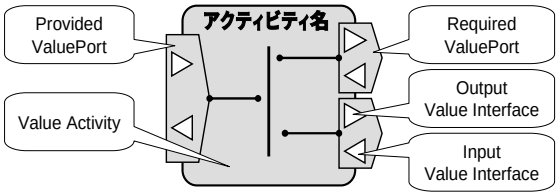

図 7 VSDL(Value-Based Service Description Language)

Value Interface:Web サービスの入出力を価値として定義する. サービスへの入力は価値要求として Input Value-Interface で定義する. サービスからの出力は価値提供として Output Value-Interface で定義する.

Value Port:一つの意味のある価値の交換を行う Value Interface の集合である. 他のサービスから起動される PP(Provided Port)と他のサービスを起動する RP(Required Port)の二つがある.

VA(Value Activity):価値交換を行う主体であり Value Port のセットを持つ. Value Port の違いにより, 一つのプロビジョニングは複数の VA を持つ.

VSDL で記述されたサービス記述は価値ブローカ内の価値レポジトリに格納され, サービス利用時におけるサービスの動的連携で利用される.

6.2. 価値交換による複合 Web サービスの自動生成

価値ブローカは, 価値レポジトリ内に格納された VSDL 記述を参照し, 以下の手順で価値交換の連鎖を実現する Value Exchange プロセスを自動生成する(図 8).

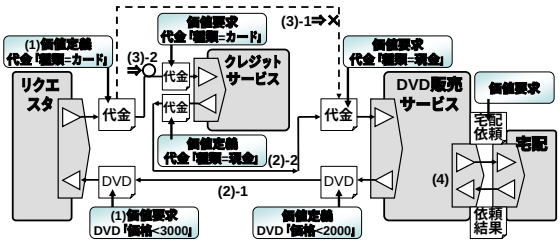

図 8 複合 Web サービスの生成手順

(1)リクエスタのポート定義:リクエスタからの価値要求と価値提供からリクエスタの RP を作成する.

(2)サービスの検索:価値レポジトリを検索し RP の価値要求を満たす価値提供を持つ VA を検索する.

(3)価値提供との比較:(2)で検索された VA の価値要求がリクエスタの価値提供を満たしているか確認する. 満たさない場合は VA の価値要求を RP として(2)と(3)を繰り返し価値交換のループを完成させる.

(4)Provided Port の解決:(3)で完成したループ内で PP を持つ VA があった場合には, (2)と(3)の手順を繰り返し価値交換の完全なループを完成させる.

7. 価値ブローカのアーキテクチャ

既存の Web サービスプラットフォームを拡張し, 価値を評価する価値ブローカのアーキテクチャを図 9 に示す.

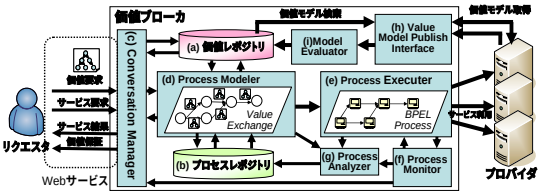

図 9 価値ブローカのアーキテクチャ

(a) 価値レポジトリ: VM-XML で記述された価値モデルと VSDL で記述された価値によるサービス記述を格納する木構造の XML データベースである.

(b) プロセスレポジトリ: 利用履歴による最適サービスの選択精度向上と Value Exchange プロセスの再利用のため, 実行済みプロセスを四階層に分けて格納する.

(c) Conversation Manager: 一つのリクエスタの価値要求で利用するサービスが一意に決定しない場合があるため, リクエスタと価値ブローカ間の対話をサポートする.

(d) Process Modeler: Value Exchange プロセスを自動生成する. Process Modeler は, 以下の三つのデータベースを検索できる. 1)サービスの機能特性を検索する UDDI レジストリ. 2)価値検索を行う価値レポジトリ. 3)プロセスの再利用に使うプロセスレポジトリ.

(e) Process Executer: Value Exchange プロセスから実行可能な WS-BPEL プロセスを生成し, BPEL エンジンを

用いて実行する[2].

(f)Process Monitor: サービスの価値保証の実現のため, Process Executer で実行される複合 Web サービスでやり取りされる価値を監視する.

(g)Process Analyzer:複合サービスのプロセスを Process Monitor での価値の評価に基づき分析しプロセスレポジトリに利用履歴として格納する.

(h)Value Model Publish Interface: プロバイダに対して価値提供や価値要求の検索や登録サービスを提供するインタフェースであり, Web サービスで実現する.

(i) Model Evaluator: プロバイダにより申請された VSDL 記述を価値ブローカの担当者が評価するためのユーザインタフェースを提供する.

提案した価値ブローカを Java, Apache Axis, XIndice, WSDL4J, BPEL4J, Oracle BPEL を用いて実装した. 開発規模は, 46 クラス, 6,810 行である.

8. プロトタイプによる評価

価値による Web サービスの動的連携の有用性を評価するため, 次の 3 つの辞書サービスを用いて評価した.

(1) NetDicV08: 三省堂が提供する辞書サービスであり, 英和, 和英, 国語辞書から総計 190,000 語が利用できる. (http://btonic.est.co.jp/NetDic/NetDicv08.asmx)

(2) Insider's Computer Dictionary (ICD): @IT が運営する辞書サービスであり, IT の専門用語が検索できる. (http://www.iwebmethod.net/icd1.0/icd.asmx)

(3) Simple Word Book:研究室で開発した辞書サービスであり, ソフトウェア工学分野の専門用語が検索できる.

このうち, NetDicV08 の英和と和英は日本語と英語の言語の変換を行っていると考えられる. その他は, 単語の意味の深化を目的としている. そこで,「単語」と「単語の意味」の二つのコンテンツの価値モデルを定義し VSDL による記述を行った. これを図 10 に示す.

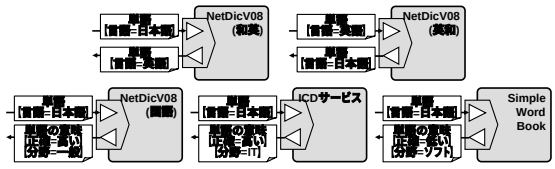

図 10 3つの辞書サービスの VSDL 記述

これらの Web サービスを価値交換に対応させるため C#を用いてラッパーブローカを作成し, 価値ブローカを用いて動的連携させ評価した. 開発規模は, 1,708 行である.

価値モデルによりプロバイダの信頼性やサービス結果の単語の分野や意味の正確性の評価尺度が定義できた. また, 価値モデルを木構造で定義したことにより, リクエスタは必要に応じた詳細度で価値要求を作成でき, 同一辞書サービスでもユーザの利用状況や職業などのコンテキストから, ユーザに適した辞書の提供が可能となった. 価値モデルにより辞書サービスの利用価値が高まったといえる.

また, 価値ブローカに単語の意味を要求し, 英語の単語を提供すると, 価値交換の連鎖を自動的に生成し NetDicV08 の英和辞書サービスと SimpleWordBook を連携できた. リクエスタはサービス連携を意識せずに, 英単語から SimpleWordBook に登録された意味が検索できる. なお, 生成された複合 Web サービスの WS-BPEL 文章は 64 行であり, 生成時間の平均は約 7,765 ミリ秒であった.

9. 関連研究

Web サービスをシステムが動的連携する技術は様々な技術が研究されている. Zeng らの研究[3]では, サービスの品質属性として 6 つをあげ, 品質特性を利用した複合プロセスの生成方法を提案している. しかし, 品質特性はサービス提供者などの属性に影響を受けると考えられ, サービスの包括的な定義と品質特性のモデル化が必要である.

10. 今後の課題

価値による Web サービスの動的連携の課題を挙げる.

(1) 多次元的価値モデル:サービスの価値は様々な分類方法が考えられるため, 多次元的な価値のモデル化と評価方法が必要である.

(2) ユーザ要求の価値モデルへのマッピング: コンテキストにより変化するユーザの価値要求を価値モデルにマッピングし評価する方法の提案が必要である.

11. まとめ

本稿では, Web サービスの動的連携を目標とし, 価値によるサービス記述である VSDL の提案を行った. また, これを実現する技術として価値モデルと価値ブローカの提案を行った. 価値モデルとは, サービスの価値を包括的に表現するものである. 価値ブローカは, 価値に基づく複合サービスの自動生成と価値保証サービス提供を行う. 最後に, 3つの辞書システムを用いて評価を行った.

参考文献

[1] J. Gordijn, et al., E-Business Value Modeling Using the e³-Value Ontology, W. Currie (ed.), *Value Creation, Elsevier*, 2004, pp. 98-127.

[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, No. 30, Mar. 2004, pp. 123-130.

[3] L. Zeng, et al., QoS-Aware Middleware for Web Services Composition, *IEEE Trans. on Software Eng.* , Vol. 30, No. 5, May 2005, pp. 311-327.