

# Web サービス連携のための多粒度リポジトリ利用環境

## Web Services Coordination based on Multi-Grain Size Repository

寺井 公一<sup>\*1</sup>    福田 直樹<sup>\*2</sup>    和泉 憲明<sup>\*3</sup>    山口 高平<sup>\*2</sup>  
 Koichi TERAJ    Naoki FUKUTA    Noriaki IZUMI    Takahira YAMAGUCHI

<sup>\*1</sup>静岡大学大学院情報学研究科                      <sup>\*2</sup>静岡大学情報学部  
 Graduate School of Informatics, Shizuoka University    Faculty of Informatics, Shizuoka University

<sup>\*3</sup>産業技術総合研究所サイバーアシスト研究センター  
 Cyber Asist Research Center, AIST

Coordination of Web services is one of the most significant issues in order to realize the Semantic Web. Although current Web services provide only syntactic descriptions of their use, it is semantic descriptions for the coordination. In this paper we propose an environment for Web services coordination. Our environment is based on a multi-grain size repository. We divide knowledge modeling level into two levels which are semantics and implementation. The repository consists of several kinds of ontologies and models for each level. These ontologies provide semantic basis for Web services coordination. In order to realize our environment, we have developed supporting tools for each level. Finally, we confirmed that the environment supports us in coordinating Web services semantically.

### 1. はじめに

Semantic Web の実現に向けて Web サービスの連携が重要な研究課題の一つとなっている。Web サービスは分散コンピューティングの新たな潮流とされ、プログラミング言語やプラットフォームに依存しないことが特徴の一つとなっている。そのため、Semantic Web の研究においては、計算機が Web 上のサービスを利用する仕組みとして期待されている。

現在、Web サービスの連携を目的として、BPEL4WS[8] を始め、いくつかの仕様が提案されている。これらの仕様は、制御構造、データフローなどの Web サービス連携のフローを記述するための語彙を提供している。しかしながら、その記述はフローに留まっており、連携の目的(タスク)や実装に関する記述は不十分である。一方で、計算機に動的に Web サービスを連携させるために、その意味情報の記述に注目が集まっている。Web サービスの意味情報に関しては、最も重要な研究として DAML-S[1] が存在する。DAML-S は意味情報を記述するためのオントロジーとモデルを与えてくれるが、その記述は意味レベルに留まっており、Web サービス連携をソフトウェアとして実装するにはギャップが存在する。

本稿では、上記の問題を解決するために、Web サービス連携を支援するための環境を提案する。本研究では、ビジネスを対象領域とし、知識モデルをセマンティクス層と実装層へと分割する。セマンティクス層は、Web サービス連携に対して、ビジネスタスク、ビジネスプロセスという視点を与えることにより、意味付けを行うものである。実装層は、セマンティクス層から与えられるビジネスプロセスを、Web サービス連携として実装する層である。

知識モデルを記述するためには、モデルを記述するための最小要素が必要である。本研究では、これを与えるために、再利用可能な知識の分類としてタスク・推論・ドメインを与える[5]。各知識を記述するためにオントロジーやライブラリを構築する。そして構築されたオントロジーやライブラリの要素を知識モデルの最小要素とする。こうして構築される再利用可能

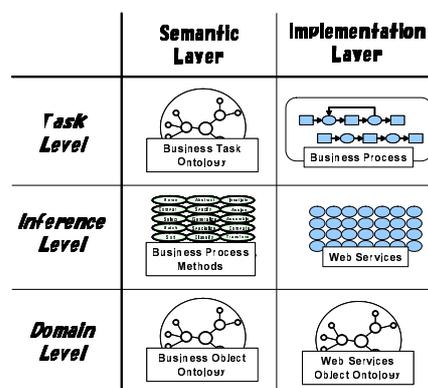


図 1: 再利用可能な知識の分類

な知識を含むセマンティクス層と実装層を、総称して多粒度リポジトリと呼ぶ。多粒度リポジトリを利用した Web サービス連携の概要を、図 2 に示す。Web サービスの連携は、リポジトリにより与えられる語彙を基に知識モデルを構築し、ユーザーとのインタラクションを交え、モデルの結合、展開を行うことにより実現する。

### 2. 多粒度リポジトリの構成

本節では、多粒度リポジトリの各粒度に含まれるオントロジーとモデルについて述べる。

#### 2.1 セマンティクス層、実装層の知識モデル

我々がある問題に取り組む時には、“何を行うべきか”、“どのように行うべきか”を最初に考え、最後にそれを“実行”する。セマンティクス層が与える知識モデルは前者 2 つに関する知識を与え、実装層の知識モデルはセマンティクス層から与えられるモデルを実行するための知識を与える。

セマンティクス層の与える知識モデルは、ビジネスタスクの階層構造と、ビジネスプロセスである。ビジネスタスク階層の上位 - 下位関係は、*has-a* の関係(タスク - サブタスクの関係)である。構築されるタスクの階層構造は、各々のビジネスモデルを表す。ビジネスプロセスモデルは業務手続きを表し、

連絡先: 寺井 公一, 静岡大学大学院情報学研究科, 〒 432-8011  
 静岡県浜松市城北 3-5-1, 電話:053-478-1473, FAX:053-473-6421, e-mail:terai@ks.cs.inf.shizuoka.ac.jp

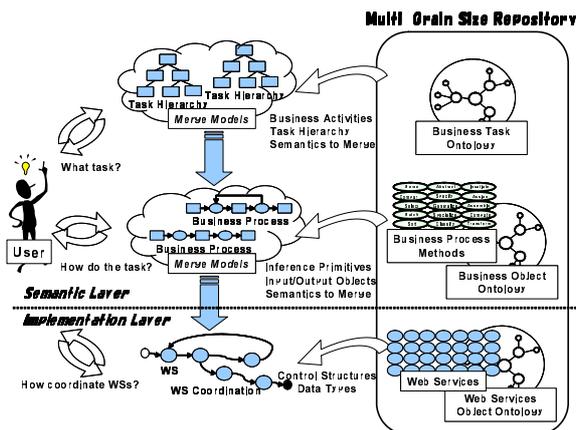


図 2: 多粒度リポジトリに基づく Web サービス連携の概要

表 1: CommonKADS Inference Primitives

Abstract	Assign	Classify	Compare
Cover	Critique	Evaluate	Generate
Group	Match	Modify	Operationalize
Propose	Predict	Select	Sort
Specify	Verify		
Obtain	Receive	Provide	Present

プロセスプリミティブであるビジネスプロセスメソッドと、そのメソッドの操作対象物であるビジネスオブジェクトから構成される。ビジネスプロセスメソッドは、CommonKADS [5] により与えられる推論プリミティブを基に設定する(表 1)。ビジネスオブジェクトはビジネスオブジェクトオントロジー(後述)中の要素から与えられる。実装層に対するモデルは、ビジネスプロセスモデルのビジネスプロセスメソッドとビジネスオブジェクトを、Web サービスを実行するためのデータ構造や、制御構造により拡張したものである。データ構造は、Web サービスオブジェクトオントロジー(後述)によって与えられ、制御構造はソフトウェアテンプレートとして与えられる。

2.2 タスク・推論・ドメインレベルのためのオントロジーとライブラリ

タスク・推論・ドメインは、それぞれ目的、方法、その方法を実行するために利用される静的な知識を与える。本研究では、この3種類の知識を与えるために、オントロジーやライブラリを構築する。オントロジーとライブラリは、知識モデルを記述するための最小要素を与える。特に、オントロジーはそれらのモデルを結合するためのセマンティクスの基礎を与えるものである。

セマンティック層では、タスクレベルに対するオントロジーはビジネスタスクオントロジー(図 3 参照)と呼ばれ、62個のビジネスタスクの has-a 階層構造として定義される。前節で述べたビジネスモデルは、このビジネスタスクオントロジーの部分構造となる。推論レベルでは、ビジネスプロセスを記述するための最小要素として、ビジネスプロセスメソッドのライブラリを与える。ドメインレベルに対するオントロジーは、ビジネスでの業務活動における操作対象物であるビジネスオブジェクトを is-a の関係により体系付けたものであり、ビジネスオブジェクトオントロジーと呼ぶ。図 4 は、このオントロジーの一部を示したものであり、Object をルートに、その下に person or organization、resource などが配置されていることが示されている。このオントロジーには 107 個のビジネスオブジェクトが含まれる。



図 3: ビジネスタスクオントロジー (一部)

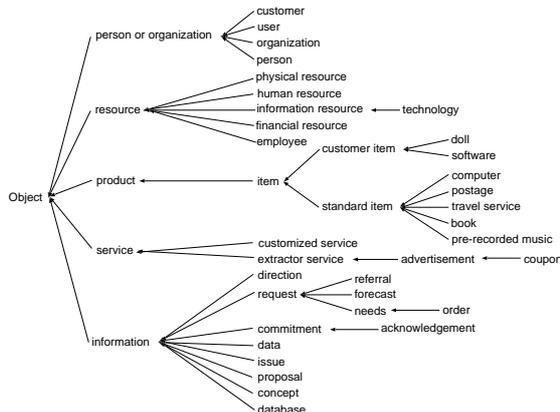


図 4: ビジネスオブジェクトオントロジー (一部)

実装層では、タスクレベルに対する知識は、セマンティック層で構築されるビジネスプロセスモデルとして与えられる。このモデルは Web サービス連携のビジネス上の目的を与えるものである。推論レベルに対しては、本環境で利用可能な Web サービスをライブラリ化したものを与える。実装に対するオントロジーは、Web サービスを実行するために利用されるデータ構造の has-a 階層構造を含んでおり、Web サービスオブジェクトオントロジーと呼ぶ。Web サービスを実行するためには、その入出力として様々なデータ構造を扱わなければならない、同じ情報を表す場合でも、Web サービスの所有者間で異なるデータ構造を利用する場合が考えられる。そこで、本研究では、このオントロジーにより統一されたデータ構造を与える。そして、前述したデータ構造の多様性の問題に関しては、変換メカニズムを与えることにより対応する。図 5 は、オントロジーの一部を示したものであり、例えば、Order というデータ構造が、Date、LineItem という他のデータ構造から構成されていることが示されている。これらのオントロジーは、Process Handbook[7] や WordNet[2] を、再構築することにより構築される [4][6]。

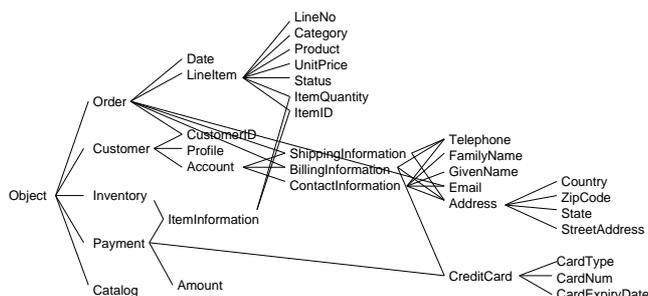


図 5: Web サービスオブジェクトオントロジー (一部)

### 2.3 ビジネスモデルライブラリとプロセスパターンライブラリの構築

本研究では、ビジネスモデル、業務プロセスの典型的なパターンをライブラリ化することにより、ユーザがテンプレートとして利用可能なリソースを提供する。このようなライブラリを構築するためには専門的な知識が必要であり、また、一から構築するのはコストがかかる。そこで、ここでは、既存のリソースを再構築することにより、それらを構築する。

ビジネスモデルライブラリは、Process Handbook に含まれるビジネスケーススタディを、ビジネスモデルとして体系化することにより得る。具体的には、ビジネスのタイプとして Distributor、Creator、Broker、Extractor、Service Provider の5つを設定し、各ケーススタディを分類する。次に各タイプ毎に、ビジネスの特徴分類階層を構築し、ケーススタディをビジネスモデルとしてその階層構造中に配置する。このライブラリには、現在、71個のビジネスモデルが存在する。プロセスパターンライブラリは、Assign、Assessment、Classification [5] などの一般的な推論に関するものを CommonKADS により与えられる推論カタログを基に設定する。そしてそれを基に、ビジネスオブジェクトやプロセスの拡張を行うことにより、ビジネス固有のプロセスパターンを設定する。本研究では30個の業務プロセスパターンを構築した。

## 3. 多粒度リポジトリを利用した Web サービス連携

### 3.1 ビジネスタスクの抽出

Web サービスを連携するためには、まず始めに、Web サービスの連携を行うタスクを同定する。本研究では、ビジネスモデルとしてビジネスタスクの階層構造が与えられている。そこで、この階層構造に含まれるビジネスタスクから、Web サービスの連携を行うタスクを選択する。現在、実際の選択に関して、本研究で提案する環境は、どのタスクを Web サービス連携として実現すべきかということを示唆しない。これは、多くの場合、あるタスク全体を支援するようなサービスの連携は不可能であり、タスクのどの部分的な作業を、サービスの連携として実現するかはユーザー自身が判断する必要があるためである。しかしながら、一度タスクを選択すれば、タスクの階層構造を利用することによりタスクを分解し、タスクのどの部分を Web サービス連携として実現すべきかという判断材料としてビジネスタスクの階層構造を利用することが可能である。また、ビジネスタスクオントロジーを参照することにより、複数のビジネスモデルを結合することが可能であり [4]、結合されたモデルにより、新たなモデルの展開が可能になる。

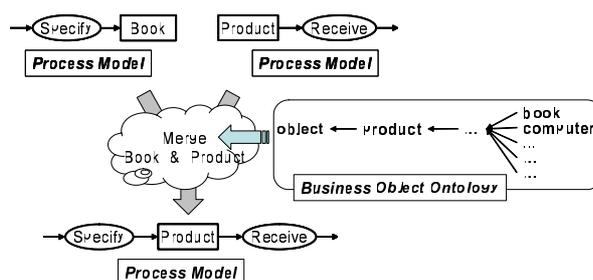


図 6: プロセスパターンの結合

### 3.2 ビジネスタスクからの Web サービスの同定

選択されたビジネスタスクは、各々、業務プロセスパターンへと展開される。本研究では、各ビジネスタスクに対して業務プロセスパターンの候補を手動で与えている。ユーザは与えられる候補の中から、自分の目的に合致するプロセスパターンを選択することができる。抽出された業務プロセスパターンは、その入出力オブジェクトを結合することにより、パターン全体を結合することができる。この際、異なる複数のオブジェクトを結合するためには、オブジェクト間の関係を明示し、その関係を利用してそれらを結合する枠組みを与える必要がある。本研究では、そのためにビジネスオブジェクトオントロジーを与える。ビジネスオブジェクトオントロジーはオブジェクト間の *is-a* 関係を与えるものであり、抽象度の差を埋めることができる。これにより、異なるオブジェクトに対して、共通なオブジェクトへの一般化、または一方のオブジェクトへと特殊化を行うことによりそれらを結合することが可能である。図 6 はその様子を示したものであり、Book が Product へと一般化されることにより、2つのプロセスパターンが結合される様子を示している。

次に、結合されたパターンにより表される業務プロセスのどの部分が Web サービスとして実現できるかを同定しなければならない。本研究では、このために、ビジネスプロセスメソッドにコミュニケーションメソッドと呼ばれる特殊なタイプを与えている。コミュニケーションメソッドは、そのプロセスが、ビジネス内外の他のプロセスとの情報のやりとりを必要とすることを明示するものである。ここでは、コミュニケーションメソッドとして Present、Receive を与え、その二つが Web サービスへの展開候補となる。コミュニケーションメソッドへの具体的な Web サービスの割り当てに関しては、現在、手動で行うものとしている。

### 3.3 プロセスモデルの入出力の同定

プロセスモデルを Web サービス連携として実行するためには、まず、その入出力を与える必要がある。本研究では、その入出力は、プロセスモデルに含まれるコミュニケーションメソッドにより明示される。コミュニケーションメソッドは、前節で述べた通り、ビジネス内外の他のシステム、ユーザ、そして Web サービスとの通信を明示するものであり、情報取得のための Receive と、情報供与のための Present がある。

現時点では、コミュニケーションメソッドの役割を、HTML による入力フォームからの値取得、結果の提示、そして入出力に対して意味付けされたいくつかの Web サービスの実行としている。具体的には、Receive の場合には、そのメソッドの出力オブジェクトに着目し、そのオブジェクトに対応する実行時

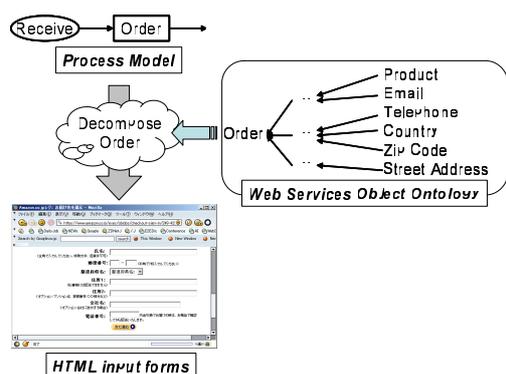


図 7: プロセスモデルの入力の同定

データ構造を取得する。得られたデータ構造は Web サービスオブジェクトオントロジーを参照することにより最小要素にまで分解される。そして、入力フォームの場合には、各要素の値を入力するためのフォームが生成される。図 7 は、Order ビジネスオブジェクトを分解し、各属性を入力する HTML フォームが生成される様子を示している。Web サービスの実行の場合には、その出力結果と、Web サービスオブジェクトオントロジーとの対応を与えることにより必要な情報を取得する。

### 3.4 Web サービスの実行

業務プロセスモデルは、Web サービスの実行に必要なとされるデータ構造や制御構造を詳細化することにより実行可能となる。データ構造に関する語彙は Web サービスオブジェクトオントロジーにより与えられる。一方で、制御構造はソフトウェアテンプレートとして与えられる。ビジネスプロセスメソッドとビジネスオブジェクトは抽象的な概念であり、それらをソフトウェアの部品として実現するためには複数の方法が考えられる。そこで、この過程はユーザとのインタラクションを交えることにより、実現方法を詳細化する。ビジネスプロセスメソッドに関しては、先行研究として CommonKADS における推論プリミティブのソフトウェア化に対する試みがなされている [3]。そこで、ここではそれに基づき、各ビジネスプロセスメソッドに対してソフトウェアのテンプレートを与え、そのテンプレートを拡張することにより制御構造を与える。

ビジネスオブジェクトは、XML により記述されるビジネスドキュメントテンプレートへと展開される。テンプレートは、e-ビジネスにおける、ビジネスドキュメントのいくつかの標準化仕様に基づいて設定されており、実行の際にはそれらのテンプレートをインスタンス化し、それを Web サービスの入出力とする。

## 4. 支援ツールの構築

本研究では、Web サービス連携を支援するためのツールを開発している。ツールの設計に当たっては、各モデルを GUI を通して扱えること、モデル間のマッピングの支援、ユーザとのインタラクションに重点を置いた。ツールは Java 言語により実装を行い、リポジトリのコンテンツは XML を用いて記述している。詳細な説明は紙面の都合上割愛する。

## 5. 関連研究と今後の課題

本研究では、現在、DAML-S との連携は扱っていない。DAML-S は Web サービスの意味情報を記述するためのモデルと、そのモデルを記述するためのオントロジーを与えて

くれる。しかしながら、意味情報間の関係を与えるような枠組みは有していない。このことは、例えば、DAML-S では Web サービスの実行プロセスのモデルを記述することはできるが、各実行プロセスがどのように関連し、時には結合することができるのかを示唆することはできないことを意味する。

本研究は、ビジネスというドメインを対象にし、タスクの *has-a* 構造を与えることにより、上記の問題に取り組んでいる。これにより、あるタスクが与えられた時にそのタスクをサブタスクへと分解することが可能となる。そして、各タスクがプロセスモデルと対応付くことにより、プロセスモデル間の *has-a* 関係を、暗黙的に与えている。さらに、ビジネスオブジェクトオントロジーを与えることにより、プロセスモデルの入出力オブジェクトを結合する仕組みを与えている。

## 6. まとめ

本稿では、Web サービスの連携を支援するための環境を提案した。提案した環境では、知識モデルをタスク、プロセス、実装という 3 つのレベルに分割した。そして、モデルを記述するためのプリミティブの提供と、モデルの意味的な結合を目的としてオントロジーを構築した。Web サービスの連携という目的に対して、現在多くの研究がなされている。本研究では、それら多くの研究が行っているような、Web サービスに付加された意味情報をどのように利用するかという視点ではなく、その情報を有効に利用するための枠組みを与えるものである。そのため、本研究により、Web サービス連携に対して新たな視点を示唆できると考えられる。

## 参考文献

- [1] DAML Services Coalition. : "Web Service Description for the SemanticWeb", In The First International Semantic Web Conference (ISWC), June 2002.
- [2] Fellbaum, C. (ed.): "WordNet", The MIT Press, 1998. see also URL: <http://www.cogsci.princeton.edu/~wn/>
- [3] 和泉 憲明、他 : "エキスパートシステム開発のための再利用可能メソッドライブラリの設計と実装", 人工知能学会誌 Vol.14, No.6, November, 1999.
- [4] Koichi Terai, et al.: "Managing Business Models for Business Application Development", PAKM 2002, LNAI 2569, pp.256-267, 2002.
- [5] Schreiber, G., et al.: "KNOWLEDGE ENGINEERING AND MANAGEMENT", The MIT Press, 1999.
- [6] Yamaguchi, T.: "Constructing Domain Ontologies Based on Concept Drift Analysis", IJCAI Workshop on Ontologies and Problem-Solving Methods: Lessons Learned and Future Trends, 13-1 - 13-7, August 1999.
- [7] The MIT Process Handbook Project: <http://ccs.mit.edu/ph/>
- [8] <http://www-106.ibm.com/developerworks/webservices/library/ws-bpel/>