

オントロジーを用いた Web アプリケーション統合プラットフォーム

A Platform to Integrate Web Applications with Ontologies

大澤 哲也*¹
Tetsuya OSAWA

丹羽 治隆*¹
Harutaka NIWA

福田 直樹*²
Naoki FUKUTA

飯島 正*¹
Tadashi IIJIMA

山口高平*¹
Takahira YAMAGUCHI

*¹ 慶應義塾大学
Keio University

*² 静岡大学
Shizuoka University

Today, there are too many Web applications on the Internet and it is difficult for a user to find applications good to user's task. To meet the requirement, a user should execute two or more Web applications. In this paper, our purpose is to reduce costs for users to do the task. We propose Web application composition system to deal with user's task automatically based on WSMO. For composing Web applications, we prepare the wrappers of some existing Web applications and light weight ontology. In order to select the compositions which are more useful for a user, the proposed system need to evaluate the quality of them. So, we conducted the evaluation experiment based on two case studies. We can show the possibility of saving the cost for user to discover the proper Web applications and also improve the trouble to execute two or more Web applications.

1. はじめに

現在多くの企業や研究者たちが、検索エンジンや分析ツール等を用いた Web アプリケーションを提供している。しかし、近年の Web の情報過多により、各 Web アプリケーションがどのような情報をデータベースに持っているかを把握し、タスクに対して適切な Web アプリケーションを見つけるのが困難となってきた。また、発見以前に、1つの Web アプリケーションでは、全てのユーザの要求に柔軟に対応できず、複数の Web アプリケーションを利用する必要がある。その場合、先の Web アプリケーションの出力をコピー & ペーストするなどして次のサービスに受け渡さなければならず、こうした手作業のプロセスに対し、多くの手間が生じてしまう。

一方で、ネットワーク上に散在する異なったプラットフォーム間のシステムをタスクに応じて自動的に発見・結合・実行させるセマンティック Web サービス[1]という研究分野に注目が集まってきた。特に、WSMO(Web Service Modeling Ontology)[2]というワーキンググループが、その実現のための概念モデルの基準を提供し、この 1 年で急速に拡大してきているが、現状では、オープンで利用可能な Web サービスは少なく、エンドユーザがセマンティック Web サービスの利益を享受できるにいたっていない。

本稿では、WSMO の概念モデルを参照し、エンドユーザのタスクに応じて、既存の Web アプリケーションを自動連携させ、タスクを自動処理できるようにすることで、ユーザの手間を軽減することを目的とする。[3]

Web アプリケーション連携を考慮する際に、情報の受け渡しの正確性や出力がユーザのタスクに合致しているかといった連携の質をシステムがどのように判断してユーザに提示するかという QoS の問題が発生する。こうした問題について WSMO のフレームワークだけでは、連携の中でどの Web アプリケーションを選択するかをシステムが判断するのは困難であり、多くの不要なパスを生成する可能性がある。このような連携の QoS に関する問題は、WSMO のモデルの中でも将来問題視されるものであると考える。

そのため本稿では、Web アプリケーション連携に関する評価

指標を独自に設け、複数の連携候補に対し、評価関数を用いた連携パスの並び替えを行い、上位の連携パスをユーザに提示して選択させることとした。

最後に、予備実験による評価指標の決定後、手間の改善という観点から、提案するシステムの有効性を検証するために、2つのケーススタディを用いて動作実験を行った。

2. セマンティック Web サービス

Web Service と Semantic Web との結合は E-commerce や B2B 統合における、次世代の革命的な Big Wave であると考えられている。Semantic Web とは WWW の次世代の姿といわれ、情報が機械可読な意味を持ち、オントロジーという機械と人が可読な概念間の仕様書(辞書)を用いて、機械と人間が共に作業することを可能にするものである。また Web Service とは、ネットワーク上に散在する異なったプラットフォーム間のシステムを連携させるための基本技術で、インターネットを経由して単純なリクエストから複雑なビジネスプロセスまでのあらゆる機能を果たすことが出来る。

その上で、セマンティック Web サービスでは、メタ情報が定義され、Web Service として利用可能なソフトウェアリソースが、タスクに応じて自動的にインターネット上から発見・結合・実行される。[4]本章ではセマンティック Web サービスの使用プロセスを紹介し、セマンティック Web サービスへの概念モデルを提供し、近年注目を浴びている WSMO(Web Service Modeling Ontology)の概要を説明後、本稿の WSMO での位置づけを述べる。

2.1 セマンティック Web サービスの使用プロセス

Web Service として利用可能なソフトウェアリソースが、タスクに応じて自動的にインターネット上から発見・結合・実行される、セマンティック Web サービスの実現のためには以下6つのプロセスが必要である。

- 登録(Publication)...サービスを使用可能にする
- 発見(Discovery)...タスクにあったサービスを見つける
- 選択(Selection)...利用可能な最も適切なサービスを選ぶ
- 合成(Composition)...サービスを連携させる
- 仲介(Mediation)...サービス連携のミスマッチを解決する
- 実行(Execution)...サービスを発動させる

上記プロセスにおいて、技術上の差異を解消し、複雑性、不均質性に対処するために WSMO は 4 種類の Mediator[6]を用いる。次節では WSMO の概要について述べる。

2.2 WSMO の概要

WSMO とはセマンティック Web サービスのためのオントロジーと概念モデルを提供していて、近年急成長しているワーキンググループで、WSMF(Web Service Modeling Framework)というフレームワークに基づいて設計されている。WSMO の特徴として、Web Service 統合の際の全ての問題に対し、出来る限り単純な解決策を提示していて実行可能性が高いといわれている。WSMO の根本となる概念図として図がある。

次章では図で示される WSMO の概念図を元に、提案システムについて述べる。

3. WSMO の概念モデルから見た提案システムの構成

本章では、初めに提案システムの流れと構成について述べた後、本システムにおける、前章で述べた WSMO の概念モデルの4つの概念について、詳細を述べる。

3.1 システムの構成

本節ではシステムの構成について述べる。はじめにユーザは自身のタスクに見合った入出力を選択する。システムは WSMO のモデルに従って、入力された入出力情報 (Goals) と使用可能な Web アプリケーションのサービス記述 (Web Services) とオブジェクトオントロジー (Ontologies) に基づき、多くの Web アプリケーション連携候補を作成する (Mediators)。そしてその後、作成された各連携候補に対し、評価関数を用いて連携の品質評価を行い、並び替えてユーザに表示する。次にユーザは、並び替えられた連携候補の中から一つの連携を選択し、入力フォームに入力値を入れる。その結果システムは、選択された各 Web アプリケーションのラッパーを通じて入力値を受け渡し、最終的にシステムはユーザに要求されたタスクの結果を表示する。

次節からは、WSMO の概念モデルを用いて連携候補を作成し、連携パスの並び替えを行う際の詳細について述べる。

3.2 Web Services

現在一般ユーザが使用可能な Web Service の数が少ないため、本稿では現在 Internet 経由で利用可能で、かつ複雑な処理を行うことができる Web アプリケーションを Web Service としてとらえ、それにラッパーを被せるという方法を用いて、サービスの自動連携を目指す。ラッパーを構築することで、各 Web アプリケーションを、Web サービスを用いたサービス連携における WSMO のモデルに対応させた。

その上で WSMO で規定されている Web Service 記述の中から、Web アプリケーションの連携と連携の品質に関するものとして、機能記述と非機能記述のみを記述することとし、さらに今回重要だと思われ、かつ表記しやすい指標を選んだ。

機能記述としては、低コストでシステムを作成するために、最低限必要とされるものである、各 Web アプリケーションの入出力だけを定義した。

非機能記述としては、WSMO の規定している中の「Availability」を 2 つの指標に分解して使用した。1 つ目は「情報量」(I)の指標で、これを各 Web アプリケーションが持つ情報量として定義した。2 つ目は「ユーザ評価」(U)の指標で、これは各 Web アプリケーションに対するユーザの評価値とした。

3.3 Goals

本稿では入力と出力で、ユーザの要求を表すことができるものとした。ただし、実際はユーザの要求は自然言語で表され、またあいまいなものが多いために、それらに対して対処していくことも WSMO では取り上げられているが、この課題に対しては本稿では扱わないこととする。

3.4 Ontologies

本稿では、システムで使用可能な各 Web アプリケーションの数が限られているために、本システム専用の固定のオントロジーを用意することとし、登録されている Web アプリケーションの入出力を収集し、全ての入出力を扱えるオントロジーとしてオブジェクトオントロジーを作成した。作成には、まず全ての Web サービスの入出力を列挙し、次に WordNet[7]を参考にして、階層構造を記述した。

3.5 Mediators

Mediator は、Web Service が協調する際に発生する数々の問題を緩和するための一般的なメカニズムである。WSMO では、セマンティック Web サービス実現のために4種類の Mediator を定義している。それぞれの Mediator は以下のように分類される。また、これらの Mediator と本稿のシステムとの位置づけを述べる。

・OO-Mediator...オントロジーの import と結合からタスクに必要なオントロジーを作成

本稿では、オントロジーは固定なので、この Mediator を考慮に入れないこととする。

・GG-Mediator...ユーザの目的と Goal とのマッチング

あるゴールを、Web アプリケーションで過去に実現可能であった代わりとなるゴールに変更する。本稿のシステムは GG-Mediator を考慮に入れないこととする。

・WG-Mediator...Web Service をゴールまで結ぶまでのミスマッチを解消

本稿のシステムは WG-Mediator をサポートする。オントロジーを用いた Web アプリケーションの入出力照合方法として、本稿では以下の二つを用いる。一つ目が厳密照合であり、後に実行されるサービスの入力集合が、前に実行されるサービスの出力集合の部分集合である照合方法をさす。二つ目が拡張照合であり、後に実行されるサービスの入力の子ノードに、前に実行されるサービスの出力があるといったパスが含まれる。拡張照合の場合、照合対象になるのは、後に実行されるサービスの入力の子ノードのみであり、子の子ノードや、親ノードの子ノードには適応しない。

・WW-Mediator...データ型、プロトコル、ビジネスプロセスのマッチング

本稿では、データ型、プロトコルのミスマッチは Wrapper によって解消されるため、この Mediator を考慮に入れないこととする。

3.6 連携の Ranking のための評価指標の定義

現在ユーザが利用できる既存の Web アプリケーションには同様のサービスを提供するものが多数あるのに加え、使用する Web アプリケーションが増加するにしたがって、本システムは多くの連携パスを生成する。それはつまり、多くのユーザのタスクにとって不要な連携が生成されることにもつながる。このような問題は WSMO の概念モデルの中で、将来問題視されるものであると見え、本稿では WSMO のモデルで生成した連携パスの QoS について、連携パスの生成後、評価関数を作成して、各連

携バスに得点付けをし、並び替えることとした。その際、3.2 節で述べたように各 Web アプリケーションが持つ記述として以下の 2 つを定義する。

Web アプリケーションの記述としての指標

- ・情報量 (\bar{I}) ... 各 Web アプリケーションの情報量の平均
- ・ユーザ評価 (\bar{U}) ... 各 Web アプリケーションに対するユーザの評価の平均

サービス連携に特化した指標

- ・パスの長さ (L) ... 連携バスの中の Web アプリケーションの数
- ・照合の種類 (M) ... 厳密照合か拡張照合
- ・連携としての評価 (C) ... ユーザの連携に対する評価

一つ目に連携バスの長さ (L) は、作成された連携バスの中で実行する Web アプリケーションの数のこととした。二つ目に照合の種類 (M) で、これは、前節で述べた 2 種類の照合 (厳密照合か拡張照合) のどちらを用いた連携かとした。三つ目に連携としての評価 (C) であり、これは各連携一つ一つに対するユーザの評価値とした。こうすることによりユーザに適した連携をシステムが選び出せると考えた。これら 5 つの指標を用いて、本稿では評価関数を作成し、連携バスの並び替えを行う。

これらの指標を考慮して、以下の評価関数を設定した。

$$V = \frac{\bar{I} + \bar{U} + M + C}{L}$$

4. 実験と評価

本章では提案システムを使った実験とその評価について述べる。はじめに実験の内容について述べ、次にシステムの実行結果と考察を述べる。

本実験では、手作業を自動化する手間の軽減の点から見た本システムの有効性と、評価関数を用いてサービス連携の QoS を計る有効性を考察することを目的とする。また、実験にあたり 17 個の Web アプリケーションを使用し、それぞれの Web アプリケーションについてあらかじめラッパーを作成した。また、それらの Web アプリケーションを基に、3.1 で定義したオブジェクトオントロジーをあらかじめ作成した。オントロジー作成にあたっては MR3[10]を使用した。

4.1 予備実験

本実験を行う前に、前章で既述した評価関数の各連携に対するユーザの評価値 (C) を決定する予備実験を行った。被験者は以下のとおりである。

- ・理系大学院生 5 人 (Internet 使用歴 5 年以上)

それぞれの被験者に好きなホテル名を一つ決めてもらい、ホテルから近くのレストランを検索するというタスクを行ってもらった。その際に、システムが作成する 24 本の連携それぞれに対して

表 1 各 Web アプリケーション連携の評価値

Web アプリケーション連携	C	Web アプリケーション連携	C
楽天 旅の窓口 Yahoo グルメ	0.82	Yahoo トラベル ロッチェリア	0.22
楽天 旅の窓口 ぐるなび	0.82	Yahoo トラベル ラーメン Bank	0.38
楽天 旅の窓口 モスバーガー	0.68	Yahoo トラベル ドトール	0.28
楽天 旅の窓口 マクドナルド	0.52	Yahoo トラベル 吉野家	0.34
楽天 旅の窓口 ロッチェリア	0.36	Aitel Yahoo グルメ	0.18
楽天 旅の窓口 ラーメン Bank	0.58	Aitel ぐるなび	0.16
楽天 旅の窓口 ドトール	0.42	Aitel モスバーガー	0.12
楽天 旅の窓口 吉野家	0.62	Aitel マクドナルド	0.08
Yahoo トラベル Yahoo グルメ	0.60	Aitel ロッチェリア	0.12
Yahoo トラベル ぐるなび	0.46	Aitel ラーメン Bank	0.16
Yahoo トラベル モスバーガー	0.24	Aitel ドトール	0.08
Yahoo トラベル マクドナルド	0.28	Aitel 吉野家	0.14

10 点満点で評価点をつけてもらい、その平均を 1 点満点で表したものを各連携の評価点である C の値として定めた。

4.2 実験内容

被験者を下記のようにそれぞれ、上級者、中級者、初級者に分け、実験を行った。このようにインターネットを現在使用している幅広いユーザに対して本システムを評価してもらうことにより、これからの本稿の課題を考察する材料とした。

- ・ Internet 使用歴 5 年以上 (上級者) 22 人
- ・ Internet 使用歴 3 年前後 (中級者) 10 人
- ・ Internet 使用歴 1 年前後 (初級者) 10 人

本実験で、被験者に行ってもらったタスクは以下の 4 つである。

答えが一意に決定するタスク

タスク 1.1: 「現在の GE の株価は日本円でいくらか？」

タスク 1.2: 「現在の IBM の株価は日本円でいくらか？」

ユーザによって答えが異なるタスク

タスク 2.1:

「品川プリンスホテルの近くで昼食を取りたいと思います。次の条件に当てはまるレストラン、または食事を取れる場所を一つずつ探してください」:

- ・ 700 円以下で食事が出る場所
- ・ 2000 円から 3000 円を予算とする比較的高級な場所

タスク 2.2:

「上野ファーストシティホテルの近くで昼食を取りたいと思います。次の条件に当てはまるレストラン、または食事を取れる場所を一つずつ探してください」:

- ・ 700 円以下で食事が出る場所
- ・ 2000 円から 3000 円を予算とする比較的高級な場所

タスク 1.1 と 1.2 は、複数のサブタスクに分解して行うタスクの中で、答えが一意に決定するものの例である。

また、タスク 2.1 と 2.2 は複数のサブタスクに分解して行うタスクの中で、検索するユーザによって答えが変わってくるタスクの例である。

被験者となる計 42 名の上級者、中級者、初級者それぞれを Group A と Group B に分け、以下の実験 1 と実験 2 を行った。

実験 1 として、タスク 1.1、1.2 を用いて 42 名の被験者に対し以下の手順で実験を行った。タスク 1.1 を普通の検索工程と同様に、検索エンジンを用いてタスクを実行した時間を計測し (手作業)、次に使用する Web アプリケーションを指定して、十分に使用方法について慣れてもらった上で、タスク 1.1 について手作業でタスクを実行した時間を計測した (手作業)。そして、最後にタスク 1.2 について、提案システムを使用した時間を計測した上で (システム利用)、それらの時間の差を計測した。

この実験において、手作業 と手作業 の差異が、検索エンジンを用いて求めるサービスを検索する時間を表している。具体的には、タスクのサブタスクへの分解と、各サブタスクに適切な Web アプリケーションを探索する時間である。また、手作業 とシステム利用 の差異が、タスクの実行とコピー & ペーストなどを用いた値の受け渡しの際にかかる時間を表す。

実験 2 として、タスク 2.1、2.2 を用いて、Group A、Group B それぞれに対し、実験順序をクロスさせて実験を行った。

Group A の人には、最初にタスク 2.1 について提案システムを使用して行ってもらった (システム利用 1) 後に、提案システムを使わずにタスク 2.2 を行ってもらった (手作業 2)。

Group B の人には、最初にタスク 2.1 についてシステムなしで行ってもらった (手作業 2) 後に、タスク 2.2 について提案システムを使って行ってもらった (システム利用 1)。

4.3 実験結果

本節では実験の結果を記す。

実験1:

タスク 1.1 およびタスク 1.2 において、各被験者がかかった時間の平均値(秒)と標準偏差は表2、表3のようになる。

表2 実験1の平均値(秒)

	手作業	手作業	システム利用
上級者	311.50	36.73	26.27
中級者	412.00	49.80	25.40
初心者	1039.50	78.60	36.90

表3 実験1の標準偏差(秒)

	手作業	手作業	システム利用
上級者	196.92	19.79	12.80
中級者	166.42	27.97	15.40
初心者	260.16	47.62	20.09

実験2:

タスク 2.1 とタスク 2.2 において、各被験者がかかった時間の平均値(秒)と標準偏差は表4、表5のようになる。

表4 実験2の平均値(秒)

	手作業	システム利用
上級者	374.45	180.32
中級者	602.10	246.50
初心者	949.90	310.70

表5 実験2の標準偏差(秒)

	手作業	システム利用
上級者	235.74	62.11
中級者	337.68	81.30
初心者	333.60	321.77

連携の評価:

ユーザの評価値の平均値の順番は表6である。また、評価関数を元に、システムが各 Web アプリケーションを並び替えた結果が表7である。

表6 ユーザ人気の上位パスと評価関数での順位

ユーザ人気ランキング		評価関数での順位
1位	楽天-Yahooグルメ	1位
2位	楽天-ぐるなび	2位
3位	楽天-マクドナルド	7位
4位	楽天-ドトール	9位
5位	Yahoo-ぐるなび	4位
6位	楽天-ラーメンBank	5位
7位	Yahoo-Yahooグルメ	3位
8位	Yahoo-マクドナルド	12位

表7 評価関数が作成した上位パス

評価関数が作成したランキング	
1位	楽天-Yahooグルメ
2位	楽天-ぐるなび
3位	Yahooトラベル-Yahooグルメ
4位	Yahooトラベル-ぐるなび
5位	楽天-ラーメンBank
6位	楽天-モスバーガー
7位	楽天-マクドナルド
8位	楽天-吉野家

4.4 考察

下記の2つの点について考察する。

1. 手作業の自動化による、ユーザの手間の軽減について
2. 本稿で提案した評価関数について

(1) 手作業の自動化による、ユーザの手間の軽減について

ルーチンワークの自動化

上級者のみに限り、実験1の手作業 とシステム利用 の所要時間の差が、有意水準 0.05 での t 検定において、有意な差異を検定することが出来た。その結果、インターネット使用の上級者は、提案システムの利用により、ルーチンワークとしての作

業の手間を自動化することによる、定量的なシステムの有効性を計ることができた。

Web 探索の自動化

全ての被験者に対し、実験2の手作業とシステム利用の所要時間の差が、有意水準 0.05 での t 検定において、有意な差異を検定することが出来た。また、本システムはルーチンワークの手間を改善するよりも、Web 探索の手間を改善する実験2に大きな有効性を示すことが出来た。

(2) 本稿で提案した評価関数について

合計 24 本の連携パスの中から、ある程度主要なものを検出することが出来、Web アプリケーション連携を WSMO のモデルで考える際に、連携パスの QoS の評価を元にした Ranking を行う必要性を確認することが出来た。

5. おわりに

4.3 節の実験1より、ルーチンワークのような決まりきった作業を自動化することにより、意味のある手間の改善を行うことが出来た。また、実験2より、インターネットを用いた情報検索に要する手間を、本システムを用いることで大幅に改善することが出来る可能性を示すことが出来た。

また、今回設定した評価関数が、24 本の連携パスの中で被験者の評価が高い主要な連携を検出していたことから、Web アプリケーション連携を評価することが意味のあることであると確認された。

今後の課題として、システム内で使用できる Web アプリケーションを増やすことと、評価関数の自動学習について考察する予定である。

参考文献

- [1] Sheila A. McIlraith et al., "Semantic Web Services", IEEE Intelligent Systems, Mar./Apr.2001, pp.46-53.
- [2] WSMO <http://www.wsmo.org/>
- [3] Kimihito Ito, and Yuzuru Tanaka: "A visual environment for dynamic web application composition", ACM PressPages: pp.184-193, 2003.
- [4] Arroyo, S., Lara, R., Gomez, J. M., Berka, D., Ding, Y. and Fensel, D: "Semantic Aspects of Web Services" in Practical Handbook of Internet Computing. MunindarP. Singh, editor. Chapman Hall and CRC Press, Baton Rouge. 2004.
- [5] OWL-S 1.0: <http://www.daml.org/services/>
- [6] Massimo Paolucci, Naveen Srinivasan, and Katia Sycara: "Expressing WSMO Mediators in OWL-S", 3rd International Semantic Web Conference (ISWC2004), 2004.
- [7] WordNet: <http://www.cogsci.princeton.edu/~wn/>
- [8] Liangzhao Zeng, Boualem Benatallah, Marlon Dumas, Jayant Kalagnanam, and Quan Z. Sheng: "Quality driven web services composition", International World Wide Web Conference, pp.411-421, 2003.
- [9] Christoph Bussler, and Katia Sycara et. al.: "Tools and Technologies for Semantic Web Service", 3rd International Semantic Web Conference (ISWC2004), 2004.
- [10] T.Morita, N.Izumi, N.Fukuta, and T.Yamaguchi: MR3: Meta-Model Management based on RDFs Revision Reflection, Joint Conference on Knowledge Based Software Engineering, Frontiers in AI and Applications, IOS Press, Vol.108, pp.228-236 (2004)