

## 概念間距離に基づく連携パス評価手法を用いた Web サービス自動連携システムの提案

高林 裕矢<sup>†</sup> 丹羽 治隆<sup>†</sup> 種田 光晴<sup>†</sup> 福田 直樹<sup>‡</sup> 山口 高平<sup>†</sup>

<sup>†</sup>慶應義塾大学 〒223-8522 神奈川県横浜市港北区日吉 3-14-1

<sup>‡</sup>静岡大学 〒432-8011 静岡県浜松市城北 3-5-1

E-mail: <sup>†</sup>{yuuya, haru\_tk, tanem, yamaguti}@ae.keio.ac.jp, <sup>‡</sup>fukuta@cs.inf.shizuoka.ac.jp

あらまし セマンティック Web サービスの実現を目指す中心的なワーキンググループの1つである WSMO は4つの柱をもつ概念モデルにより、ユーザのタスクを満たす Web サービス連携パスが一意に特定できるとしている。しかし、利用可能な Web サービスが増加するにしたがって、多くの連携パスが生成され、求めるタスクにとって不要な連携パスが生成されることが予想される。このような問題は将来 WSMO の概念モデルの中でも問題視されるものであると考えられる。本論文では、WSMO の概念モデルを参照して、概念間距離に基づいた評価関数を適用したユーザタスクに応じて Web サービスを自動連携するシステムを提案し、後半部ではインタラクションを用いたシステムの改善方法について示す。

キーワード セマンティック Web サービス, 概念間距離, 評価関数

## On Implementing an Automated Web Service Composition System Using Combination Path Evaluation Based on Conceptual Distances

Yuya TAKABAYASHI<sup>†</sup> Harutaka NIWA<sup>†</sup> Mitsuharu TANEDA<sup>†</sup>

Naoki FUKUTA<sup>‡</sup> and Takahira YAMAGUCHI<sup>†</sup>

<sup>†</sup>Keio University 3-14-1 Hiyoshi, Kohoku-ku, Yokohama, 432-8011 Japan

<sup>‡</sup>Shizuoka University 3-5-1 Johoku, Hamamatsu-shi, Shizuoka, 223-8522 Japan

E-mail: <sup>†</sup>{yuuya, haru\_tk, tanem, yamaguti}@ae.keio.ac.jp, <sup>‡</sup>fukuta@cs.inf.shizuoka.ac.jp

**Abstract** Recently, Web services technology is widely spreaded, and the number of Web services are increasing. Semantic Web services are aiming to discover, compose and execute Web services automatically. WSMO (Web Service Modeling Ontology) assumes that we can find only one Web service path satisfying the task of user by using Ontologies, Goals, Web Services and Mediators. However, when the number of available Web services are increased, many number of web service cooperation paths are generated. They may contain unwanted cooperation paths for the task. This problem is considered as a future work of general idea models of WSMO. In this paper, we extend a general idea model of WSMO and developed a recommendation system which can generate Web services cooperation paths to achieve the user's task automatically by applying an evaluation function for generating cooperation paths.

**Keyword** Semantic Web Service

### 1. はじめに

近年、分散システム連携技術としての Web サービス [1] が注目を集めている。また、その技術により提供されるサービスとしての Web サービスも普及してきており、そのサービス数は増加してきている。ソフトウェア・システム開発者らにとっては、Web サービスの普及により作業負担が大幅に軽減されることが期待できるが、今後 Web サービスの数が爆発的に増加していくと新たに別の問題が発生してくる。それは、現状の Web の世界で生じているのと同様に Web サービス自動

連携実行のための基盤が提供され、Web サービスの数が莫大なものとなり、タスクを達成するために必要な Web サービスを探索し、それらを適切に組み合わせて使用することが困難になってしまうという問題である。

一方で、セマンティック Web [2] の技術を応用してタスクに応じて自動的に Web サービスを発見・連携・実行することを目的とした、セマンティック Web サービス [3] という研究分野に注目が集まってきている。この技術により Web サービス自動連携実行のための基盤が提供され、システム開発者らの負担は軽減されてい

くことが期待されている。セマンティック Web サービスの実現を目指す中心的なワーキンググループとして WSMO (Web Service Modeling Ontology) [4]がある。WSMO では、Ontologies, Goals, Web Services, Mediators の4つを柱としたセマンティック Web サービス実現のための概念モデルを提案している。WSMO の概念モデルを用いると、ユーザのタスクに合わせた Web サービスの自動連携を行うことで、ユーザのタスクを満たす Web サービス連携パスが一意に特定できるとしている。しかし、現在では利用可能な Web サービスが加速的に増加しつつあるため、Web サービス連携を実行することで非常に多くの連携パスが生成され、求めるタスクにとって不要な連携パスが生成されることが予想される。このような問題について現状では WSMO では十分に考慮されていないが、近い将来 WSMO の概念モデルの中でもこのような問題が生じてくるものと思われる。

本研究では、サービスを自動連携統合するシステムをサービス連携パス生成部とサービス連携実行部の2つから構成し、サービス連携パス生成部では WSMO の概念モデルを参照し、ユーザの実行したいタスクに対して、多数の Web サービスから多数の Web サービス連携パスを作成するシステムを作成し、それらの Web サービス連携パスに対して概念間距離に基づいた評価関数を適用することで、莫大な数の連携パスの中からユーザのタスクを満たす連携パスを抽出する手法を提案する。サービス連携実行部においては、必要に応じてユーザーとインタラクションをとるための機構を提案する。

## 2. 提案システムの構成

本章では、システムをサービス連携パス生成部における WSMO の概念モデルの4つの概念との関わりと評価関数について述べた後、サービス連携実行部における構成について述べる。全体の構成を図1に示す。

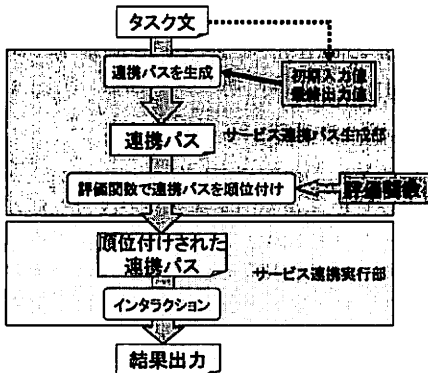


図1 システム全体の構成

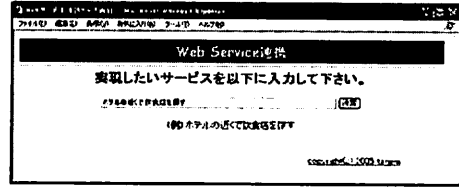


図2 タスク入力画面

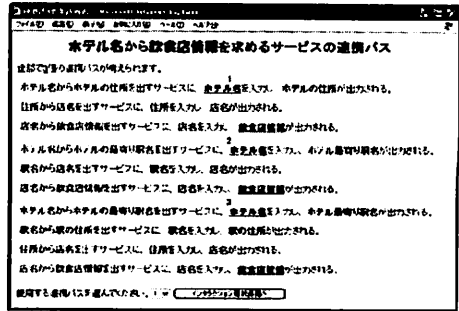


図3 候補選択画面

### 2.1. サービス連携パス生成部の構成

サービス連携パス生成部では、はじめにユーザは自身の満たしたいタスクを入力する。システムは入力されたタスクを分析し、タスクの入力値、および出力値をとりだす(図2)。次に、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 サービスのサービス記述 (Web Services) とオブジェクトオントロジー (Ontologies) に基づき、Web サービス連携パス候補を作成する (Mediators)。その後、作成された各連携パス候補に対し、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(図3)。

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。

#### 2.1.1. Web Service

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、利用可能なサービスを増やすことにする。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。また、実験・評価部分においては、評価指標の適切さを評価するために Web サービスの数を増やす必要があり、架空のサービスを多数設定し、シミュレーションを行った。

### 2.1.2. Goals

本稿では自然言語でユーザの要求を表すことができるものとした。ただし、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を 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。

### 2.1.3. Ontologies

本稿では、Web サービスのサービス記述として入出力のデータ構造を「対象 (O)」「属性 (A)」という形で持たせることとした。これは、例えば「どこかの住所」を入力するとその近辺の「飲食店の住所」を出力するサービスの場合、入力の場合の「住所」はどこかの住所でもよいが、出力の場合の「住所」は「レストラン」のものに限定されている。このように、サービスの入手値には具体的な対象が決まっている場合があり、「何の (対象)」「どんな値 (属性)」であるのかという情報を持たせた方がよいと考えたためである。

また、オブジェクトオントロジーとして、照合の種類と概念間距離を求めるために EDR 電子化辞書[5]を用いることとした。

### 2.1.4. Mediator

Mediator は、Web Service が協調する際に発生する数々の問題点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。

- 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
- GG-Mediator…ユーザの目的と Goal とのマッチング
- WG-Mediator…Web Service をゴールまで結ぶまでのミスマッチを解消
- 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

これらの Mediator のうち本稿のシステムは WW-Mediator をサポートする。オントロジーを用いた Web サービスの入出力照合方法として、本稿では以下の二つを用いる。一つ目が記号照合であり、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(図 4)。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(図 5)。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。

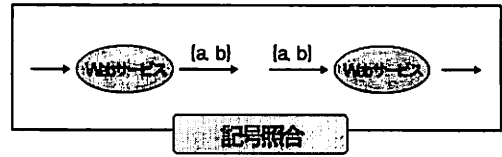


図 4 記号照合

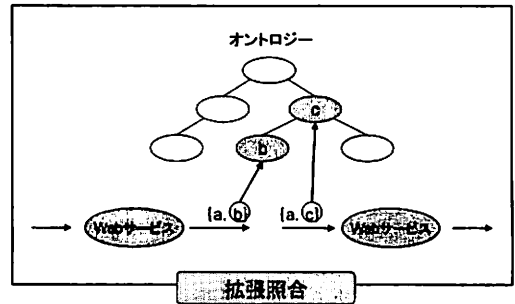


図 5 拡張照合

## 2.2. 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たがって、本システムは多くの連携パスを生成する。この結果、多くのユーザのタスクにとって不要と思われるような連携が生成されるという状況がおきる。この問題は WSMO の概念モデルの中では具体的な解決法は示されず、将来的に解決されるべき問題として扱われている。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、われわれが提案する評価関数を用いて、各連携パスに得点付けをし、並び替えることとした。評価関数で具体的に用いる指標として、関連研究[6]では、Web アプリケーションの記述としての指標として、

- 情報量 (I) …各 Web アプリケーションのデータベースが持つ情報量
- ユーザ評価 (U) …各 Web アプリケーションに対するユーザの評価

また、サービス連携に特化した指標として、

- 連携パスの長さ (L) …連携パスに含まれる 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を用いている。

本研究では、このうち、Web アプリケーションに特化した場合の指標であるものを除いて、Web サービスにも汎用的に使用できると考えられる、連携パスの長さ (L)、照合の種類 (M) を用いることとした。

連携パスの長さ、照合の種類といった指標だけを用

いて Web サービスの連携パスの評価を行う場合、連携パスとしては文法的には成立しているが、パス内で使用している Web サービスに、ユーザのタスクには本質的に関連のないものが含まれていたり、ユーザが欲しい情報に関するサービスがまったく使用されていないといった、的外れな連携パスを含まないようにするための、連携パスの評価方法が別途必要となる。本論文では、生成された Web サービス連携パスとユーザのタスクとの間の意味的な差を評価するための評価指標の利用を考える。

### 2.2.1. 概念間距離に基づく指標

ユーザのタスクから抽出したキーワードと連携パス中の Web サービス間での連携の中間値とのオントロジー上の語彙の概念距離を、評価指標として考える。ここで使った中間値とは図6に示されるような前のサービスから後ろのサービスへと受け渡される「住所」、「郵便番号」、「金額」といった値のことを指す。

キーワードと中間値のオントロジー上の概念距離を計測する際、ユーザのタスクからキーワードを抽出する方法に関しては、ユーザのタスクを単一の文にし、その文章を形態素解析することで、意味のある単語を抽出し、それを前から順に2つずつキーワードペアとして採用していく。ここでキーワードのペアとしたのは、比較対象となる Web サービスの入出力地のデータ構造を、前述したように「対象」と「属性」のペアとしたためである。

そして、得られた各キーワードペアに対して、連携の中間値とのオントロジー上の概念の間の距離を順に見ていき、距離が最小となるものをそのキーワードペアと連携パスとの距離  $d$  とした。それをすべてのキーワードペアに行い、距離  $d$  の合計値  $D$  をキーワードペアと連携パスとの距離とした。評価指標  $D$  はその値が小さければ小さいほど、ユーザのタスク内容と近い、つまり、ユーザのタスクを満たしている可能性が高い連携パスとみなされる。

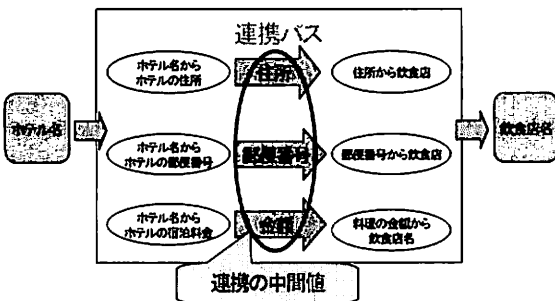


図6 中間値の概念

### 2.2.2. 評価関数

以上の指標を用いて、(1)のように、評価関数を設定した。

$$V = \frac{M}{L + D} \dots (1)$$

この評価関数は、評価関数値  $V$  が大きいほど評価が高いとするときに、この  $V$  の値に対して正の効果を与える評価指標を分子に、負の効果を与える評価指標を分母に配置し、それぞれの和をとった形となっている。

### 2.3. サービス連携実行部の構成

次にサービス連携実行部では、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、その連携に対してシステムは、ユーザに対してサービスを連携したときにサービスによる出力が複数になる部分を表示し、その部分でユーザがインタラクションをとる必要があるかないかマークし、入力フォームに入力値を入れる(図7)。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、必要な部分でユーザとのインタラクションをとり(図8)、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。

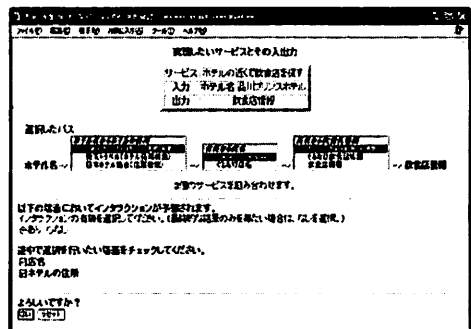


図7 インタラクション選択画面

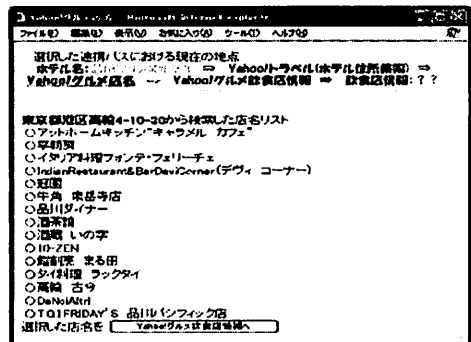


図8 インタラクション画面

### 3. 連携パス生成に関するケーススタディ

本章では、2章で述べた評価関数を用いてシミュレーションと評価を行ったケーススタディについて述べる。

#### 3.1. 方法

評価用のサービス連携パス生成システムを作成し、それに対してタスク文を与え、システムにより生成された連携パスに対して、評価関数を適用することで連携パスに対して順位付けを行い、その順位付けされた結果に対して検証を行った。

##### 3.1.1. システム

与えられたユーザタスクと Web サービスから、連携パスを生成する試作システムを作成した。このシステムにより、ユーザのタスク、および初期入力値、最終出力値をもとに、Web サービスを連携していくことができる。

##### 3.1.2. ユーザの行うタスク

Web サービス連携パスを生成する際の目的にあたるものであり、このユーザの行うタスクを満たすような連携パスを評価関数で抽出していく。本研究では、ユーザの入力したタスク文からキーワードを抽出し評価に使用するため、ユーザタスク文としては、単一の文であることが望ましい。また、初期入力値と最終出力値もタスク文から抽出する。そこで、本ケーススタディで使用するユーザの行うタスク文は

タスク 1: 「ホテルの周辺のレストランの地図がほしい」

タスク 2: 「あるメニューの料理を出すレストランの近くの駅の名称が知りたい」という二つのものとした。

ユーザの入力したタスク文からのキーワード抽出には、自然言語処理技術の適用が必要となる。本研究の提案は、連携パスに対する評価関数に主眼をおいているので、自然言語処理の部分は簡略化する。本論文では、上記の一つ目のタスク文に対して形態素解析を行い、「ホテル」、「周辺」、「レストラン」、「地図」という単語が抽出されたものとし、これらの単語を前から順に選択し、「ホテル、周辺」、「レストラン、地図」というキーワードペアが作成できるものとする。二つ目のタスクに対しても、形態素解析の結果、「メニュー」、「料理」、「レストラン」、「近く」、「駅」、「名称」という単語が抽出され、「メニュー、料理」、「レストラン、近く」、「駅、名称」というキーワードペアが作成されるものとする。

#### 3.1.3. Web サービス

本研究では、現在使用可能な Web サービスに限界があることから、仮定の抽象的な Web サービスを利用する。この仮定の抽象的な Web サービスを定義する際、以下の5つの点を考慮した。

- ① 本研究のタスクを満たす連携が行える Web サービスを含めること
- ② ①のサービスとの連携が可能であるが、タスクを満たすには関係のないサービスを含めること
- ③ ②のサービスとの連携が可能なサービスを含めること
- ④ 全てのサービスに対して、現実世界のサービスとして実在する内容のサービスであること
- ⑤ できる限り多くのサービスを定義すること

以上5つの点を考慮し、50個の仮定の抽象的な Web サービスを定義した。表1にその一例を示す(表中の\*は対象が何でもよいということを表す)。

表1 サービスの種類

入力		出力		サービス内容
対象(O)	属性(A)	対象(O)	属性(A)	
ホテル	名称	ホテル	住所	ホテル名→ホテルの住所
ホテル	名称	ホテル	郵便番号	ホテル名→ホテルの郵便番号
*	住所	レストラン	住所	住所→レストランの住所
*	郵便番号	レストラン	住所	郵便番号→レストランの住所
*	住所	*	地図	住所→地図
*	金額	レストラン	名称	金額→レストラン名
駅	名称	レストラン	名称	駅名→レストラン名
料理	種類	レストラン	名称	料理名→レストラン名
レストラン	名称	レストラン	住所	レストラン名→レストラン住所
*	住所	駅	名称	住所→最寄り駅名
著書	名称	書籍	名称	著者名→書籍名
ホテル	名称	株	金額	ホテル名→ホテルの株価
病院	名称	病院	住所	病院名→病院の住所
書籍	名称	書籍	番号	書籍名→書籍番号
企業	名称	株	金額	企業名→企業の株価

#### 3.2. 結果

タスク 1: ホテルの周辺のレストランの地図がほしい  
このタスクに対して、本システムにより 54546 個の連携パスが生成された。その後、評価関数により順位付けを行った結果の上位 5 パスは表 2 の通りである。

タスク 2: あるメニューの料理を出すレストランの近くの駅の名称が知りたい

このタスクに対して、本システムにより 234 個の連携パスが生成された。その後、評価関数により順位付けを行った結果の上位 5 パスは表 3 の通りである。

#### 3.3. 考察

「ホテルの周辺のレストランの地図がほしい」というタスク 1 では、54546 個の連携パスが生成され、生成された連携パスに対して評価関数を適用した。結果、評価関数により最高評価値を得たパスが三つ抽出され、

その三つの連携パスともユーザのタスクを十分満たすような、意味のある連携パスとみなせるものであった。「あるメニューの料理を出すレストランの近くの駅の名称が知りたい」というタスク2では、234個の連携パスが生成された。こちらの連携パスに対しても評価関数を適用した結果、評価関数により最高評価値を得たパスが一つ抽出され、この連携パスもユーザタスクを十分に満たす意味のある連携パスとみなせるものであった。

以上のことから、ユーザタスクから生成された連携パス候補に対して、本研究で提案した評価関数を適用することにより、多数の連携パスの中からユーザのタスクを満たす意味のある連携パスを抽出し、上位に提示することができたと考えられる。

表2 タスク1の結果

パス順位	評価値	連携内容	
		連携順	サービス内容
1	0.1	1	(ホテル名 から ホテルの住所)
		2	(住所 から レストランの住所)
		3	(住所 から 地図)
2	0.1	1	(ホテル名 から ホテルの住所)
		2	(住所 から レストランの電話番号)
		3	(電話番号 から 地図)
3	0.1	1	(ホテル名 から ホテルの住所)
		2	(住所 から レストランの郵便番号)
		3	(郵便番号 から 地図)
4	0.090909091	1	(ホテル名 から ホテルの住所)
		2	(住所 から レストランの郵便番号)
		3	(郵便番号 から レストランの住所)
		4	(住所 から 地図)
5	0.090909091	1	(ホテル名 から ホテルの住所)
		2	(住所 から レストランの住所)
		3	(住所 から レストランの郵便番号)
		4	(郵便番号 から 地図)

表3 タスク2の結果

パス順位	評価値	連携内容	
		連携順	サービス内容
1	0.028316	1	(メニュー名 から 料理ジャンル名)
		2	(料理名 から レストラン名)
		3	(レストラン名 から レストラン住所)
		4	(住所 から 郵便番号)
2	0.025641	1	(メニュー名 から 料理ジャンル名)
		2	(料理名 から レストラン名)
		3	(レストラン名 から レストラン住所)
		4	(住所 から レストランの住所)
		5	(住所 から 郵便番号)
3	0.025	1	(メニュー名 から 料理ジャンル名)
		2	(料理名 から レストラン名)
		3	(レストラン名 から レストラン住所)
		4	(住所 から レストランの電話番号)
		5	(電話番号 から ホテルの住所)
		6	(住所 から 郵便番号)
4	0.025	1	(メニュー名 から 料理ジャンル名)
		2	(料理名 から レストラン名)
		3	(レストラン名 から レストラン住所)
		4	(住所 から レストランの郵便番号)
		5	(郵便番号 から レストランの住所)
		6	(住所 から 郵便番号)
5	0.02439	1	(メニュー名 から 料理ジャンル名)
		2	(料理名 から レストラン名)
		3	(レストラン名 から レストラン住所)
		4	(住所 から レストランの住所)
		5	(住所 から レストランの郵便番号)
		6	(郵便番号 から レストランの住所)
		7	(住所 から 郵便番号)

#### 4. 今後の課題

本研究では、Web サービス連携パスに対する評価関数の提案を行い評価関数の有用性を示すことができた。

本論文で評価に使用したサービスは仮定の抽象的なWeb サービスであり、実際に何かの仕事を行える実サービスではない。これは、現状使用可能なWeb サービスの数が少なく、現実の問題に対して使用できるWeb サービスは限られたものしかないという理由に基づくものである。今回提案した評価関数に使用している評価指標は、パスの長さ、Web サービス間の入出力値の照合方法、サービス連携パスの中間値とキーワードとの概念距離など、抽象的な指標しか用いておらず、実サービスに基づいた評価指標は含まれていない。今後Web サービス技術の普及により、使用できるWeb サービス数が増加し、Web サービス連携評価に十分なサービスが利用可能となったときには、実サービスに固有の問題から、今回提案した評価関数を改良する必要が生じると考えられる。将来の実サービスを用いたサービス連携の実現時においても、Web サービス連携パスに対して有効な評価ができるようにするために、実サービスを使用することを想定した評価指標を取り入れ、新たな評価関数を作成することが今後の課題である。

#### 文 献

- [1] Web Services: <http://www.w3.org/2002/ws/>
- [2] T. Berners-Lee, J. Hendler, and O. Lassila, "Semantic Web", Scientific American, vol. 284, pp.34-43, 2001
- [3] S. McIlraith and T. Son and H. Zeng, "Semantic Web services", IEEE Intelligent Systems. Special Issue on the Semantic Web, vol16, num.2 pp.46-53, March/April, 2001
- [4] WSMO: <http://www.wsmo.org/>
- [5] 日本電子化辞書研究所, "EDR 電子化辞書 (第2版) 仕様説明書", 2001
- [6] 大澤哲也, 丹羽治隆, 福田直樹, 飯島正, 山口高平, "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", 第19回人工知能学会全国大会, 1A4-03, 2005