F-040 連携品質を考慮したオントロジーに基づくWebサービスの自動連携
Ontology-based Automated Service Coordination with QoS

福田 直樹 † 肥塚 八尋 ‡ 和泉 憲明 § 山口 高平 ¶
Naoki Fukuta Yahiro Koezuka Noriaki Izumi Takahira Yamaguchi

1. はじめに

インターネット技術を生かしたコンポーネント（ソフトウェアの部品）の利用法の一つとして提案されたのがWebサービスである．Webサービスを利用することで，システム開発者は開発コストを抑え，短時間でシステムを開発することが可能となる．また，Webサービスを利用して作られたシステムを利用するユーザも，複数のタスクを一気に実行可能なシステムの利用を介してWebサービスの恩恵を受けることができる．

Semantic Web Services[1]では，ウェブ上に多数存在するサービスに対してメタデータを付け，エージェントがユーザにとって重要なサービスを効率よく的確に発見し，それらを合成・連携させることを目標としている．Semantic Web Servicesに関しては，フレームワークや実装[2]に関して研究が進みつつある．OWL-S[3]（以前はDAML-Sと呼ばれていた）は，サービスやその連携を記述するための標準の１つとなりつつある．OWL-Sで記述されたサービスの照合機構[4]が提案されている．これは，ソフトウェア工学におけるコンポーネント同士の照合技術[8]から派生していると見ることができるが，Semantic Web Servicesに関する研究では，単なる類似性の判定を越えて，それらを自動で連携させることに焦点をおいている点で異なっている．本研究分野における最も盛んな研究課題の１つが，自動サービス合成である．自動サービス合成に関しては，プランニングに基づくアプローチ[6]，および照合に基づくアプローチ[4]が研究されている．

サービス連携に関する機構の研究は盛んであるが，実在するサービスを連携させようとした際に，どのような問題が起きるかについては，まだ実在するWebサービス自身が少なく，それらのサービスに対するオントロジーも整備されていないため，十分な考察が行われているとは言いがたい．本論文では，出張業務支援システムを構築した際の知見に基づき，実在するサービスに対してオントロジーを用意し，そのサービス間の連携に関する考察を述べる．

2. 出張業務支援におけるサービス連携と課題

2.1 出張業務支援システムにおけるサービス連携

業務による出張の際には，交通手段・乗り継ぎの検索，ホテル予約，および社内決済等，複数の作業を行う必要がある．これらの作業で用いる情報は相互に関連しており，たとえば列車乗り継ぎ検索で得られた乗り継ぎ情報や運賃を出張旅費の社内決済書類に自動で転記する等，必要な情報を相互に連携させることで，業務上の作業負担を軽減できると考えられる．

一方で，出張業務に利用可能なWebアプリケーションは，列車乗り継ぎ検索，特急券予約，ホテル予約等が存在し，社内決済等のイントラネットアプリケーションもWebアプリケーションされている場合が多い．我々は，これらのWebアプリケーションを連携させながら一連の出張関連業務の遂行を支援するシステムを開発した．本システムでは，ユーザは専用のWebページからシステムにアクセスする．システムは，必要に応じて外部のWebアプリケーションを呼び出し，必要なデータを内部に格納するとともに，ユーザに結果を提示する．ユーザは，本システムを利用することにより，他のWebアプリケーションを直接呼び出すことなくそれらの機能を利用でき，必要なデータをシステム内部で自動的に連携させることにより，先の転記に相当するようなユーザの無駄なデータ入力が軽減される．

本システムから利用するWebアプリケーションとしては，ジョルダン（列車乗り継ぎ検索），旅の窓口（ホテル予約），えきねっと（特急座席予約）および社内出張決済システムを利用し，それらの相互連携をコア機能として実装した．コア機能の利用時に，MapFan(地図情報サービス)，CabStation(タクシー情報サービス)等，連携が可能なサービスを提供する他のWebアプリケーションとも，必要に応じて連携可能とした．図１は，本システムのユーザインタフェースである．図１では，ユーザの入力した出張予定に基づいた列車乗り継ぎ経路の探索結果が表示されている．ユーザが列車乗り継ぎ経路の確定ボタンを押した後に，その乗り継ぎ経路が連係データとして利用され，乗り継ぎ経路を考慮した（たとえば，乗り継ぎ経路沿線上での最寄り駅の付近となるような）ホテル空室検索が行われる．

2.2 サービス連携システム構築における課題

一見すると，どのサービス同士を連携させたら有益かは，システムの適用ドメインを決めれば自明にわかりそうに思える．我々が出張業務支援システムを開発する際に，サービス連携プロセスを作ろうとしたときには，連携可能と思われるようなサービスを探してくること自体が容易ではなかった．課題となったのは，サービスの機能同定問題と，補完サービスの存在による探索空間の爆発であった．

2.2.1 サービスの機能同定問題

あるWeb上のサービスがあったときに，そこが潜在的にどんなサービスを提供可能かを判断すること（機能の同定）自体が，困難な場合がある．特に，多数の入力と出力を持つようなサイトの場合，それらをどのように

†静岡大学情報学部
‡静岡大学大学院情報学研究科
§産総研 サイバーアシスト研究センター
¶慶應義塾大学理工学部管理工学科

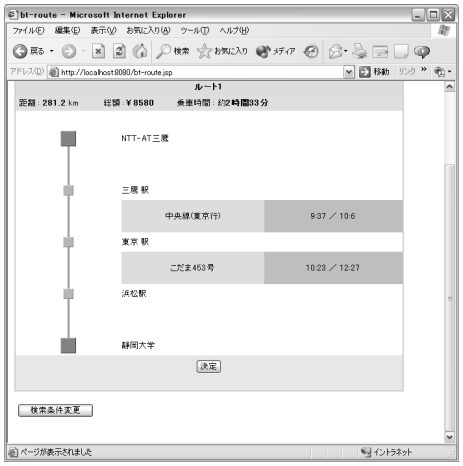


図 1: 出張業務支援システム

組み合わせたときに，どんなことに使えるのかを，容易には想像できないことがある．例えば，Web 上のサービスの１つである MapFan を使って,「特定の場所の周辺にあるコンビニを調べられる」ということは,「施設の住所から地図が表示できること」,「地図上の周辺施設を表示できること」,「周辺施設の表示の際にその施設の一覧が含まれること」,「その周辺施設にコンビニが含まれること」の４つの情報を総合して，はじめてわかることであった．たくさんの入力と出力を持つサービスがあるようなときには，その多様な用途を人間が頭で考えて連携パターンを作っていくのには限界があると考えられる．

2.2.2 補完サービスの存在による探索空間の爆発

一見すると連携できそうにないサービス同士でも，それらを連携させられるような「つなぎ」となるサービスが見つかることがあり，目的とするサービス同士が直接連携できないからといってすぐにあきらめてしまうのがもったいないことだということがわかった．例えば，会社から出張先への経路を提示する際に，出張先の最寄り駅から出張先までの経路も提示したいと考え，MapFan の経路探索機能が利用できないかと考えた．MapFan による検索では正確な住所がわかっている必要があるが，出張先の正確な住所の入力を求めるのはユーザに対しての負担が大きい．ここで，Yahoo 電話帳を用いて施設名から住所の検索を行う機能と連携させてやることにより，出張先名称をキーとして，会社の最寄り駅から出張先までの列車乗り継ぎ経路と，最寄り駅から出張先までの経路や距離をあわせて提示することが可能となった．このようなサービス連携を考えることにすると，サービス連携として考慮する対象は，直接的に必要なサービスを提供するもののみに限定されないため，サービス連携を考える際の探索範囲が膨大になる．補完サービスを考慮したサービス連携プランを人間が考えるのというのは，システムの構築時には意外と難しいものであることがわかった．

図 2: サービスオントロジーの一部

3. オントロジーに基づくサービス連携候補の探索

本節では，出張業務支援システム構築で得られた知見を元に，実在する Web 上のサービスを連携する際の，サービス連携候補の探索について考察する．本ドメインで Web サービスとして利用可能なものが現状では十分になく，Web サービスをオントロジーに基づいて連携させるための方式も議論が行われている最中であり，Web サービス自身の信頼性の保証のための機構も発展途上であるため，サービスの自動連携機構の実現，およびその連携の正当性の保証を議論することは現状では困難であると考えられる．

本論文では，サービスの自動連携機構，および連携の正当性の保証を行うことは扱わず，そのかわりに，Web 上のサービスが Web サービス化され，十分なオントロジーによる記述が準備された状況を想定し，適切なサービスの連携候補を提示するための手法について考察することを目標とする．

3.1 オントロジーの構築

出張業務支援に役立ちそうなサービスを Web 上から収集し，サービスオントロジーを構築した．各 Web 上のサービスが持つ入出力を分析して，オブジェクトオントロジーを構築した．以下で，それぞれのオントロジーについて概説する．

サービスオントロジー　サービスオントロジーを構築するために，Web サービスとなる動的にコンテンツを提供するサイトをウェブ上から収集した．一つの Web サービスは，入力パラメータと出力パラメータを持つモデルとして記述される．構築したサービスオントロジーの一部を図 2 に示す．収集した Web サービス群から，入出力パラメータを抽出し，次にオブジェクトオントロジーを構築する．

オブジェクトオントロジー　動的なサービス連携パスの組み合わせを考えるためには，各サービスの入出力データ型の同定を行う必要がある．それを実現するために，対象とするサービスの全入出力パラメータを扱うオブジェクトオントロジーを構築した．構築したオブジェクトオントロジーには，Web サービスの入出力パラメータに登場する全ての属性を明記した．まず今回の実験に

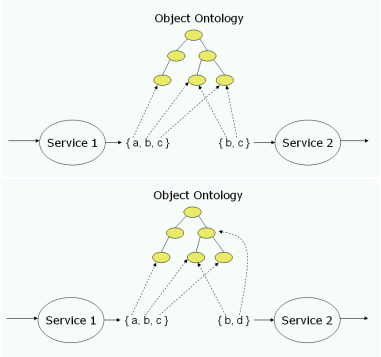


図 3: 厳密照合（上）と拡張照合（下）

使えそうな Web サービスを収集し，各 Web サービスが持つ入出力パラメータを列挙した．次に，洗い出した全属性に対し，WordNet[5] との対応関係をマッピングし，必要な部分のみを切り出したものをオブジェクトオントロジーとした．

構築したオブジェクトオントロジーは，約１５０個のノードを持つ．

3.2 概念の上位関係に基づくサービス入出力の照合

サービスを相互連携させるのに十分なサービス自身が提供されていなかったり，連携に必要な入出力が厳密には一致しないという理由から，サービス連携が困難となるような状況に対処するために，入出力パラメータの照合方法を２種類用意した．一つは「厳密照合」であり，後に実行されるサービスの入力属性集合が，前に実行されるサービスの出力属性集合の部分集合であり，かつ２つのサービスの共通集合に属する属性集合は，全てオブジェクトオントロジー内の同じノードにマップされる（図 3 上）．もう一つは「拡張照合」であり，後に実行されるサービスの入力属性集合が，前に実行されるサービスの出力属性集合の部分集合とはならずに，後に実行されるサービスの入力属性の子ノードに，前に実行されるサービスの出力属性があるといったパスが含まれる（図 3 下）．拡張照合の場合，照合対象となるのは跡に実行されるサービスの入力属性の子ノードのみであり，子の子（孫）ノードや，親ノードの子（兄弟）ノードには適応しない．

本照合方法は，MapFan のような，非常に多くの具体的要素を含む抽象的な出力を持つサービスに対する入出力の照合にも役立つと考えられる．

3.3 入出力に基づくサービス連携候補の探索

タスクの遂行に際してユーザから与えられるもの（初期属性集合）と，タスク達成の目標となる事項（目標属性集合）を設定し，初期属性集合から最終的にゴールとなる属性の集合を出力するまでのサービス連携パスを組み立てる．サービス連携パスを探索する際のパスの長さ（深さ）は，ユーザによって与えられるものとする．

初期属性集合と，ゴールとなる最終属性集合を設定し，得られた結果の一例を図 4 に示す．図 4 の (a) および (b) にはそれぞれ，初期属性集合・終了属性集合・拡張照合の有無・探索回数を示してある．図 4 中の楕円は挿入された Web サービスを表し，そこから伸びる矢印は次に実行可能な Web サービスを指す．矢印に付けられている文字列は，拡張照合によるものか，厳密照合によるものかを「extension」,「strict」として表現する．図 4 を含む 6 つの条件と，その条件下で得られたサービス連携パスの総数を，表 1 に示す．ここで，パスの総数とは，解の総数を指す．

表 1: 結果に含まれるパスの総数

	(a)	(b)	(c)	(d)	(e)	(f)
開始点／終了点	(1)	(1)	(2)	(2)	(2)	(2)
拡張照合	アリ	アリ	ナシ	ナシ	アリ	アリ
探索数	3	5	5	7	5	7
パス数	16	827	62	248	70	280

開始点／終了点: (1) 出張先→駅　(2) 駅→ホテル予約

表 2: 各評価項目の重み

	パスの長さ	情報量の豊富さ	UI 評価値	拡張照合の有無
重み	0.8	0.6	0.5	1.0

3.4 サービス連携品質を考慮した絞り込み

パスの評価項目として，生成されたパスの長さ，情報量の豊富さ，ユーザビリティ評価値 (Usability Index)，拡張照合の有無の４つを設定した．また，各評価項目に対して，表 2 に示す重みを設定した．

ユーザビリティ評価値は，そのサービスの利用のしやすさであり，与える入力パラメータの準備の容易さや，サービス内容の人間からの理解の容易さを基準に主観的な評価値を与えている．

今回収集した各サービスに対して，情報量の豊富さ，ユーザビリティ評価値に関し，1 点満点の点数を主観的評価により与えた．表 3 にその値を示す．

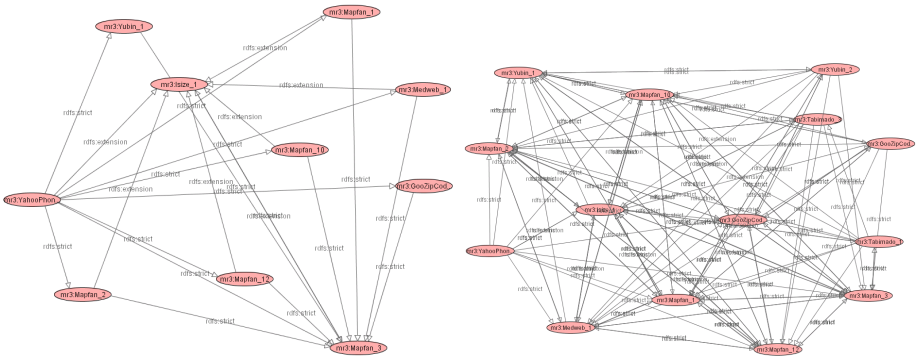
表 3 で示した点数と，表 2 を元に，3.3 節で得られた連携パス全てに関して，次の評価関数を用いて評価値（V）を得る．

$$V = \frac{\sqrt{V_a}}{\sqrt{(\frac{L-(max+6)}{1-(max+6)})^2 * 0.8 + 2.1}} \qquad (1)$$

ただし，V_a は次のように定義される．

$$V_a = (\frac{L-(max+6)}{1-(max+6)})^2 * 0.8 + (\frac{\sum_{k=1}^{L} I_k}{L})^2 * 0.5 + (\frac{\sum_{k=1}^{L} U_k}{L})^2 * 0.6 + (\frac{\sum_{k=1}^{L} C_k}{L-1})^2 * 1.0$$

式 1 の中で，L はパスの長さを表し，I は情報量，U は UI の良さをあらわす．C は次へのサービスが拡張照合であれば 0.5，厳密照合であれば 1.0 が入る．分母は，この関数の分子で得られる最大値をあらわす．また，max はユーザにより設定された探索回数である．



(a)

開始:出張先，終了：駅，拡張照合アリ，探索回数：3

(b)

開始:出張先，終了：駅，拡張照合アリ，探索回数：5

図 4: サービス連携パスの例

表 3: サービス情報

サービス名	情報量	UI
Mapfan	0.91	0.76
旅の窓口	0.87	0.65
Yahoo!電話帳	0.78	0.54
国土地理院	0.64	0.62
タクシーサイトドットコム	0.78	0.7
国内線.com	0.85	0.77
キャブステーション	0.68	0.5
HORNET	0.76	0.74
ジョルダン	0.86	0.83
goo郵便番号	0.68	0.62
郵便ホームページ	0.68	0.64
AMAZON	0.95	0.72
えきねっと	0.79	0.48
紀伊国屋	0.81	0.67
イサイズじゃらん	0.83	0.65
PRONET	0.67	0.44
測量計算	0.66	0.45
るーとMap	0.9	0.67
Webcat	0.75	0.59
7桁郵便番号検索	0.63	0.61
MEDWEB	0.73	0.71
便利コム	0.82	0.61
GSTART	0.77	0.77
PAR72 PLAZA	0.77	0.76
INET	0.69	0.83
国内線時刻表	0.79	0.65

表 4: 最高評価値

	(a)	(b)	(c)	(d)	(e)	(f)
得点	0.881	0.888	0.904	0.909	0.904	0.909

式 1 によって得られた各パスの点数から，3.3 節で示したそれぞれの結果について，最も評価値の高かった Web サービス連携パスの点数を表 4 に示す．どのケースでも，評価値が 0.9 以上かそれに近い連携パスが得られており，それらは人間の直感にも大きく反しなかった．

4. おわりに

本論文では，サービス自動連携システム構築に向けた，サービスの連携候補の自動生成および連携品質を考慮した候補の絞り込み方法について述べた．

サービス自動連携を実現するためには，サービスに対するオントロジーの準備方法の確立，サービス連携の信頼性に対する検討など，解決すべき問題が多くある．これらの問題に対する考察は，今後の課題としたい．

参考文献

[1] Sheila A. McIlraith et al.,“Semantic Web Services”, IEEE Intelligent Systems, Mar./Apr.2001, pp.46-53.

[2] E. Motta, J. Domingue, L. Cabral, and M. Gaspari: “IRS-II: A Framework and Infrastructure for Semantic Web Services”, *In Proc. of 2nd International Semantic Web Conference(ISWC2003)*, pp.306–318, 2003.

[3] OWL-S 1.0: http://www.daml.org/services/

[4] M. Paolucci, T. Kawamura, T. R. Payne, and K. Sycara : “Semantic Matching of Web Services Capatilibities”, *In Proc. of International Semantic Web Conference (ISWC2002)*, pp.333–347, 2002.

[5] WordNet: http://www.cogsci.princeton.edu/ wn/

[6] D. Wu, B. Parsia, E. Sirin, J. Hendler, and D. Nau : “Automating DAML-S Web Services Composition Using SHOP2”, *In Proc. of 2nd International Semantic Web Conference (ISWC2003)*, pp.195–210, 2003.

[7] www.xmethods.net: http://xmethods.com/

[8] A. M. Zaremski and J. M. Wing: “Specification Matching of Software Components”, *ACM Transactions on Software Engineering and Methodology*, Vol.6, Issue4(October 1997), pp.333–369, 1997.