

機関番号：82636

研究種目：研究活動スタート支援

研究期間：2009～2010

課題番号：21800095

研究課題名（和文） サービススーパービジョンによる水平型Webサービス連携

研究課題名（英文） Horizontal Web Service Composition using Service Supervision

研究代表者

田中 正弘 (TANAKA MASAHIRO)

独立行政法人情報通信研究機構・知識創成コミュニケーション研究センター言語基盤グループ・専攻研究員

研究者番号：70534444

研究成果の概要（和文）：

サービスコンピューティングでは、同じビジネスロジックに異なるサービスを割り当てることで、様々な要件を実現できる。しかし、十分なサービスが無く、どのように Web サービスを組み合わせても要件を満たすことができないことがある。そこで本研究では、Web サービスの選択と実行時適応を包括的に行うプロセスを提案した。サービス選択を制約充足の拡張問題として定義し、実行時適応が必要となるサービスを効率的に特定するアルゴリズムを示した。

研究成果の概要（英文）：

Services computing technologies enable us to configure properties of a composite service by assigning various services to the same business logic. When enough services are not available, however, any combination of services cannot satisfy given requirements. Therefore we proposed a comprehensive process which realizes service selection and runtime adaptation. We also showed an efficient algorithm which identifies services to be required by the service selection as an extension of CSP (Constraint Satisfaction Problem).

交付決定額

(金額単位：円)

	直接経費	間接経費	合計
2009年度	1,080,000	324,000	1,404,000
2010年度	850,000	255,000	1,105,000
年度			
年度			
年度			
総計	1,930,000	579,000	2,509,000

研究分野：工学

科研費の分科・細目：情報学・知能情報学

キーワード：サービスコンピューティング

1. 研究開始当初の背景

サービスコンピューティング技術の広がりに伴い、様々な組織によってサービスが公開されている。サービスコンピューティング技術のメリットの一つは、オープンな環境において異なる組織が公開したサービスを容易に横断的に連携させられることである。例

えば、言語資源と呼ばれる翻訳プログラムや辞書などをサービスとして連携させる、言語グリッドと呼ばれるプラットフォームが開発され、運用されている。そこでは、様々な組織の持つ言語資源を、標準化されたインターフェースを持ったサービスとすることで相互連携を実現されている。インターフェ

スのみの抽象的なサービス定義に基づいて定義された複合サービスが提供されており、実行時に、実際に呼び出すサービスの指定（バインディング）を与える。これにより、非機能的性質の異なる複合サービスを容易に構成できる。

このような手法による Web サービス連携を、水平型 Web サービス連携と呼び、AI プランニングなどによって実行フローを生成する垂直型 Web サービス連携と区別する。従来研究では、制約最適化技術を用いて水平型 Web サービス連携を行う技術が提案されてきた。そこでは、ユーザの嗜好を表す目標関数を最大化するようにサービスの選択が行われる。機能的に等価で QoS などの非機能的性質のみ異なる Web サービスの増加に伴い、水平型 Web サービス連携の重要性は増している。

2. 研究の目的

複数の Web サービスを連携させた複合 Web サービスを構築するにあたって、構成要素となる原子 Web サービスの組み合わせの決定が困難なことがある。これは、各原子 Web サービスの様々な属性が、どのように複合 Web サービス全体の実行結果の品質に影響を与えるかを、ユーザが把握することが困難であることによる。また、従来のプログラムコンポーネントと異なる Web サービスの特性として、スループットや実行結果の品質などの性質が常に変化することが、適切な Web サービスの選択をより困難なものにしている。

そこで本研究では、適切な原子 Web サービスの組み合わせを実行時に発見することを目的とする。実行時にサービスの属性を取得し、実際にサービスを実行して結果を評価するサイクルを繰り返すことで、適切なサービスを選択する。

3. 研究の方法

サービス連携を扱った多くの従来研究では、非常に多くのサービスが提供されている中から、適切な組み合わせを発見することを前提としていた。しかし実際には、要件を満たす適切なサービスが存在しないことが多い。そこで本研究では、サービススーパービジョンと呼ぶ複合 Web サービスの実行制御技術をサービスの適応に用いることによってこの問題を解決する。サービスの実行時適応によって既存のサービスの性質を変化させることができ、十分な数のサービスが存在しない場合にも、要件を充足させながら動作させることが可能となる。

複合サービスは通常、インターフェースのみを定めた抽象サービスに基づいて構成され、実際に呼びだされる具象サービスのエン

ドポイントは実行時に割り当てられる。このような複合サービスの実行に当たっては、以下に示す3種類の制約を考慮する。

1つ目は、サービスの組み合わせについての制約である。ある辞書サービスの提供者がポリシーとして、ある翻訳サービスと連携させることを禁止していることがある。従って、明示的にユーザから具象サービスを指定されなかった抽象サービスについては、システム側で全てのサービス提供者のポリシーが満たされるように具象サービスを選択する必要がある。

2つ目は、原子サービスの仕様についての制約である。例えば、翻訳サービスに定められた文字列長の上限を超える入力を与えられるなど、実行時に前段のサービスの出力が後段のサービスの仕様に適合しないことがある。このような場合、入力を分割してから翻訳を実行するなどの適応処理が必要になる。

3つめの点は、複数のサービスの実行状態に関する制約である。複合サービスの中で、2つの辞書サービスを並列実行するように定義されているとする。しかし、選択したサービスによっては、サーバへの負荷を避けるため、同じクライアントからの複数の同時アクセスを禁止しているとする。このような場合、複合サービスで並列実行として定義された処理を、直列的に実行されるように制御する必要がある。

これらの問題を解決することを目的に、アーキテクチャとアルゴリズムの提案を行う。

4. 研究成果

3種類の制約をそれぞれ解決する3つのレイヤが、相互に連携するアーキテクチャを提案する（図1）。

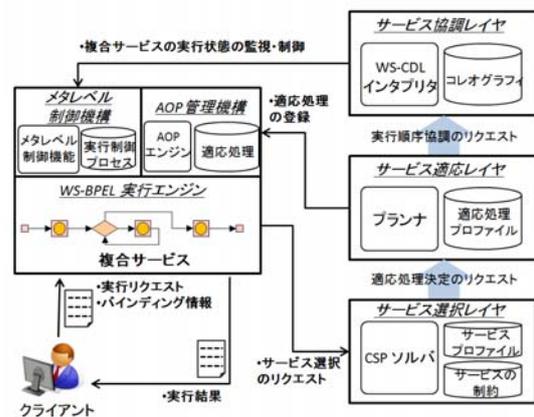


図1 アーキテクチャ

このアーキテクチャは以下のように動作する。まず、クライアントが複合サービスの実行リクエストを、サービスのバインディング情報と共に WS-BPEL エンジンに送信する。次

に、具象サービスの指定されていなかった抽象サービスについての具象サービスの選択、サービス提供者のポリシーに適合しない具象サービスの適応、及び実行順序の制御が図1右側の3つのレイヤによって行われる。各レイヤの主な機能は以下の通りである。

サービス選択レイヤ サービスの組み合わせについての制約を満たすように、それぞれの抽象サービスに対応する具象サービスを選択する。選択は後述の制約充足問題としての定式化に基づいて行われる。制約を満たすサービスの組み合わせが存在しない場合には、上位レイヤでサービスの適応を実行する。

サービス適応レイヤ 複合サービスに含まれる原子サービスの適応を行うことによって、連携される他のサービスの提供者のポリシーを満たすように、サービスの属性を変更する。プランニングによって登録された適応処理を組み合わせた上で、適応の対象となるサービス呼び出しの前後にアスペクト指向プログラミングに基づいて処理を挿入することによって実現する。このレイヤで制約を満たすことができない場合、上位レイヤでサービスの協調を実行する。

サービス協調レイヤ これまでの研究で提案された複合サービスのメタレベル制御機能を用いて、各サービスの実行がWS-CDLで記述されたサービス実行プロトコルに従うよう実行順序を変更する。このレイヤでの制御によって制約を満たすことができない場合、複合サービスの実行は失敗する。

サービス選択レイヤでは、制約を満たすサービスの組み合わせを発見するだけでなく、制約を満たす組み合わせがない際に適応を行う必要があるサービスを特定する。そこで、サービス選択を制約充足問題 (CSP) として以下のように定式化する。

変数: 複合サービスに含まれる抽象サービスに相当する。

ドメイン: 抽象サービスに対応する具象サービスの集合に相当する。

制約: サービス提供者によってポリシーとして課された制約に相当し。サービスのプロファイルや実行時のサービスの入出力に基づいて定義される述語として表現する。

アーキテクチャ図に示したように、全ての制約を満たすサービスが存在しないとき、適応処理が適用される。しかし、プランナを用いた適応処理の発見は、実行のパフォーマンスを低下させる可能性がある。従って、適応処理の対象となるサービスを効率的に特定する必要がある。

そこで本研究では、Open CSP の解法を適用する。Open CSP は、CSP の拡張問題であり、既存のドメインでは解が得られない場合に、ドメインを拡張して新たな値を獲得する。このとき、ドメインの拡張が必要となる変数は、バックトラックによって解の探索を行ったとき、探索木で最も深い位置にあったものであることが示されている。そこで、その変数に相当するサービスの適応が必要となる。図2に探索過程の例を示す。各ノードは抽象サービスに相当し、枝の分岐は具象サービスの選択に相当する。

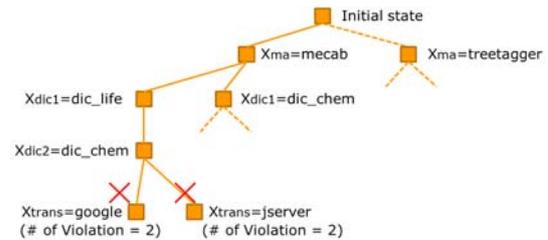


図2 Open CSPにおける探索木

このとき、図のように深さ4に相当する、翻訳サービスの選択においてバックトラックが行われ、またそれ以上の深さに探索木が伸びなかった場合、このノードでドメインの拡張を行う。

また、いずれの具象サービスに適応処理を行うかを決定するため、Partial CSP のアイデアを適用する。すなわち、選択時にもっとも制約違反度が小さくなるような具象サービスに対して適応を試みる。

これを行うアルゴリズムを図2に示す。

```

function: searchCombination(X, D, C)
入力: X: 変数  $\{X_1, \dots, X_n\}$ , D: ドメイン  $\{D_1, \dots, D_n\}$ , C: 制約
1:  $i \leftarrow 1, k \leftarrow 1, N \leftarrow \infty$ 
2: while  $(i > 0)$ 
3:   if  $D_i$  の全値をチェック済み then reset  $x_i, i \leftarrow i-1$ 
4:   else
5:      $x_i \leftarrow D_i$  の次の値
6:     if  $\{x_1, \dots, x_i\}$  が C を満たす then  $k \leftarrow \max(k, i+1)$  end if
7:      $i \leftarrow i+1$ 
8:   if  $i > n$ 
9:     if ViolationCount $\{x_1, \dots, x_n\} = 0$ 
10:      return  $\{x_1, \dots, x_n\}$ 
11:     end if
12:     if  $N > \text{ViolationCount}\{x_1, \dots, x_n\}$ 
13:        $N \leftarrow \text{ViolationCount}\{x_1, \dots, x_n\}$ 
14:       currentSelection  $\leftarrow \{x_1, \dots, x_n\}$ 
15:     end if
16:      $i \leftarrow i+1$ 
17:   end if
18: end if
19: end while
20:  $x_k \leftarrow x_k$  に適応を適用して得られたサービス
21:  $D_k \leftarrow D_k \cup \{x_k\}$ 
22: 変数の順序を変更し  $x_k$  を  $X_1$  とする
23: return searchCombination(X, D, C);

```

図3 適応処理の対象となるサービスの決定アルゴリズム

このアルゴリズムは、深さ優先探索でサー

ビスの組み合わせを探索する。もっとも深いノードのインデックスは k として記録される (6 行目)。また探索の過程で、最小の制約違反度とそれを与えるサービスの組み合わせはそれぞれ N 及び *currentSelection* として記録される (13, 14 行目)。サービス xk がサービス適応レイヤに渡される。適応処理によって性質が変化した xk' は、新たな変数としてドメインに追加される (20, 21 行目)。

また、複数の組織が適応処理を適用した結果、コンフリクトが生じ、それぞれの組織の目的や、本来の複合サービスの要件が果たされなくなることがある。そこで、本研究ではモデルチェッカを用いた検証を行う。

初めに、制御用ワークフローの実行モデル (WS-BPEL で記述) が、モデルチェッキングのための検証モデルに変換される。この変換は、従来研究によって提案された手法で可能であり、本研究項目の対象外とする。制御対象となるサービス連携ワークフローの実行中に、制御用ワークフローによってビジネスロジックが変更されると、このモデルの修正と再検証が必要となる。

そこで、ビジネスロジック変更 API のそれぞれに対して、検証モデル修正の操作を定義した。モデルの修正は、SPIN によるモデルチェッキングを用いた検証の過程で生成される、実行状態に対応する探索空間を変更することによって実現される。また、サービス提供者は、提供するサービスの実行ポリシーを与える。このポリシーは、比較的記述の容易な LTL (Linear Temporal Logic) によって表現される。ポリシーの例として、負荷の集中を避けることを目的とした同一提供者によるサービスの同時実行の禁止や、競合する他の提供者によるサービスとの連携の禁止などの制約が記述可能である。最終的に SPIN モデルチェッカに対して、修正された検証モデルと LTL で記述されたサービス実行ポリシーが入力され、サービス実行ポリシーの違反の可能性があるかどうかを検証される。ビジネスロジックの変更の結果、サービス実行ポリシーの違反がありえる場合には、制御対象のサービス連携ワークフローと制御用ワークフローの動作が停止し、サービス実行は中止される。

提案された実行制御の検証機構では、制御用ワークフロー中でビジネスロジックの変更 API が実行されるたびに、モデルの修正と再検証が必要になる。このため、サービス連携ワークフローの実行パフォーマンスが低下する可能性がある。そこで、SPIN を改良し、再検証時の計算コストを小さくするアルゴリズムも示している。検証モデルに基づいて生成される探索空間のうち、ビジネスロジック変更 API によって変化した部分のみの探索を行う。

5. 主な発表論文等

(研究代表者、研究分担者及び連携研究者には下線)

[学会発表] (計 1 件)

Masahiro Tanaka. Service Supervision for Service-oriented Collective Intelligence, 7th IEEE International Conference on Services Computing (SCC 2010), pp. 154-161, Hyatt Regency Miami, Miami, USA, July 7, 2010.

[その他]

ホームページ等

<http://langrid.nict.go.jp/~mtanaka/>

6. 研究組織

(1) 研究代表者

田中 正弘 (TANAKA MASAHIRO)

独立行政法人情報通信研究機構・知識創成
コミュニケーション研究センター言語基
盤グループ・専攻研究員

研究者番号：70534444

(2) 研究分担者

()

研究者番号：

(3) 連携研究者

()

研究者番号：