

研究種目：若手研究 (B)  
 研究期間：2007 ～ 2008  
 課題番号：19700155  
 研究課題名 (和文) ユーザとエージェントの協調作業によるドメイン依存型の Web サービス連携  
 研究課題名 (英文) Domain-Specific Web Service Composition by Coordinating Users and Agents

## 研究代表者

村上 陽平 (MURAKAMI YOHEI)  
 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・研究員  
 研究者番号：00435786

研究成果の概要： Web サービスを連携する際、各 Web サービスの入出力を合わせるだけでは、正しい結果を得られない場合がある。例えば、日英翻訳サービスと英独翻訳サービスを接続し独訳の結果を得ることができるが、中間の英語に多義語が含まれると、英独翻訳は日本語での本来の意味が分からず誤った訳語を選択することがある。このような問題を解決するために、各サービスの処理コンテキストを伝搬させるエージェントの協調作業によるサービス連携技術を開発した。

## 交付額

(金額単位：円)

	直接経費	間接経費	合計
2007年度	2,000,000	0	2,000,000
2008年度	1,000,000	300,000	1,300,000
総計	3,000,000	3,000,000	3,300,000

研究分野：総合領域

科研費の分科・細目：情報学・知能情報学

キーワード：エージェント, Web サービス, ワークフロー, オントロジー

## 1. 研究開始当初の背景

Web サービスは、Web 上で提供されているアプリケーションを、人間ではなくソフトウェアから利用可能にする技術のことである。近年、Google API に代表されるように多くの Web サービスが開発されてきたことで、Web サービスを連携させ複合サービスを構築するサービスコンピューティングが注目されている。既に IBM や Microsoft といった企業を中心となって、Web サービスをワークフローによって連携する技術を開発しており、ワークフローの記述言語として WS-BPEL が OASIS の標準となっている。また、WS-BPEL の実行エンジンも多数開発

されており、Web サービスを連携し複合サービスを構築する環境は整ってきている。

このような実行に重きを置いた産業界の試みを基に、学術界では複合サービスの自動構築に焦点が当てられている。複合サービスを自動的に構築する手法には、ワークフローを自動生成する垂直型サービス連携と、ワークフローに当て嵌めるサービスを選択する水平型サービス連携がある。それぞれプランニング技術や制約最適化技術などを用いて多数の研究が行われ、Web サービスの自動連携は、当該分野においてもっとも重要な課題の一つとして位置づけられている

## 2. 研究の目的

Web サービスの連携を行う際、これまで、各 Web サービスのインターフェースの整合性と、各 Web サービスに設けられたポリシーを満たすことだけが考慮されていた。しかしながら、サービスの入出力を合わせるだけでは、ユーザの満足のいく正しい結果が得られない場合がある。例えば、日英翻訳サービスと英独翻訳サービスを接続し独訳の結果を得ることができるが、中間の英語に多義語が含まれると、英独翻訳は日本語での本来の意味が分からず誤った訳語を選択することがある。日本語で「彼女の欠点は大きな問題だ。」と入力すると「Her fault is a big problem.」という英訳を介して「Ihre Schuld ist ein grosses Problem.」と独訳され、「欠点」が「責任」と翻訳されてしまう。これは、最初の日英翻訳でどの単語をどの訳語に訳したかという処理コンテキストが、英独翻訳に伝達されなかったことが原因である。このような問題に対処するためにも、各 Web サービスに処理コンテキストを扱うサービスエージェントを配置し、各サービスエージェントを全体で有機的に連携させる連携エージェントとの協調作業により、ユーザのゴールを満たすサービス連携を実現する。

## 3. 研究の方法

(1) 処理コンテキストを考慮した Web サービス連携を行うため、連携エージェントがサービス実行の文脈情報を集約してサービスエージェントを連携させるマルチエージェントアーキテクチャを翻訳サービス連携に適用した例を図 1 に示す。各々のサービスエージェントは、自分の処理できる内容以外の情報は何も知らず、受け取ったデータのみを見て処理を行い、結果を次のエージェントに渡す。このようなサービスエージェントに、データ処理時に考慮したコンテキストを連携エージェントに報告する機能と、連携エージェントから他のサービスエージェントのコンテキストを受け取ればそれに基づいた処理を行う機能を追加する。

連携エージェントは受け取ったコンテキストを元にサービスエージェントに指示を出し、コンテキストに沿わない処理が行われ

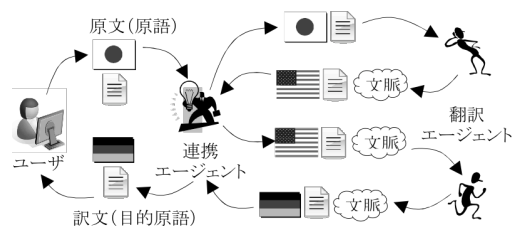


図 1 : コンテキストに基づくサービス連携

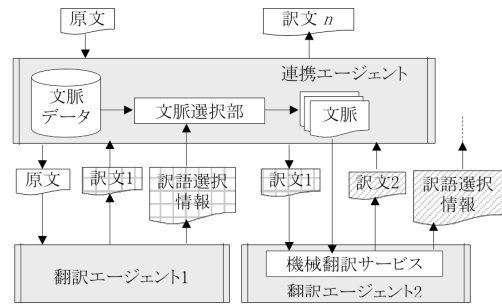


図 2 : マルチエージェントアーキテクチャ

ることがないよう連携を行う(図 1)。

図 2 にコンテキストに基づくサービス連携のためのマルチエージェントアーキテクチャを示す。サービスエージェントである翻訳エージェントは、データ処理時の訳語選択情報を処理コンテキストとして連携エージェントに送る。連携エージェントは、取得した処理コンテキストを基に次のサービスエージェントに合った処理コンテキストを、ドメインに特化したコンテキストレポジトリから選択し、次の翻訳エージェントに与える。連携エージェントが保持するコンテキストレポジトリは、翻訳サービスを連携する場合には、翻訳対象の文の文脈に関する情報や、翻訳対象の文を含む文書全体の文脈、訳語選択に用いられる多言語同義語辞書などがあり、何らかの手法で獲得されているものとする。翻訳エージェントは入力データを受け取り、連携エージェントから与えられた文脈に合わせて処理を行う。そして、処理時に考慮した処理コンテキストを連携エージェントに報告し、次の翻訳エージェントに出力データを渡す。このような処理を繰り返して目的の結果を取得する。サービスエージェントに連携エージェントが一括して指示を出すことで、全体として一貫性のある処理を行うことができる。以降ではこのモデルの実現として、機械翻訳ドメインを取り上げ、連携エージェントが訳語選択情報を処理コンテキストとして使用し、コンテキストレポジトリとして多言語対訳辞書を用いる場合の連携手法を述べる。多言語対訳辞書は多言語の同義語データを記載した辞書であり、サービスエージェントである翻訳エージェントが同義語情報を利用して訳語選択を行うよう連携エージェントが連携を行う。

(2) 文脈を用いた機械翻訳サービス連携を実現するマルチエージェントアーキテクチャのアルゴリズムを図 3 と図 4 に示す。これらはマルチエージェントモデルの最もシンプルな実装であり、原文の単語を含む同義語の集合を処理コンテキストとみなしている。

---

**Algorithm 1: COORDINATOR-AGENT CA**

---

```
1:  $s_i$  /* Source sentence */
2:  $o_i$  /* A word in sentence  $s_i$  */
3:  $MTA$  /* An ordered list of translation agents
   ( $MTA = \{MTA_1, \dots, MTA_n\}$ ) */
4:  $MTA_i = \{(s_i, s_{i+1})\}$  /* A translation agent;
   a set of pairs of
   sentence  $s_i$  and  $s_{i+1}$  */
5:  $T_i$  /* A set of  $n$ -tuples  $(w_1, w_2, \dots, w_n)$ , where  $w_k$ 
   is included in  $s_k (k \leq i)$ ; All  $n$ -tuples are
   registered in  $n$ -tuple repository */
6:  $Q_i$  /* A set of pairs  $(o_i, m_{i+1})$ , where  $o_i \in s_i$  and
    $m_{i+1}$  is the modified translated
   word for  $o_i$  */
7: when received (ask,  $s_1$ ) from user do
8:    $T_1 \leftarrow \{(w_1, w_2, \dots, w_n) | w_1 \in s_1\}$ ;
9:   for each  $MTA_i$  in  $MTA$  do
10:    send (request,  $(s_i, T_i)$ ) to  $MTA_i$ ;
11:    when received (response,  $(s_{i+1}, Q_i)$ ) do
12:       $T_{i+1} \leftarrow \text{SELECT-POSSIBLE-N-}$ 
        TUPLES( $T_i, Q_i$ );
13:    end do;
14:  end loop;
15:  send (reply,  $s_{n+1}$ ) to user;
16: end do;
```

---

図 3 : 連携エージェントのアルゴリズム

つまり、原文の単語がその文の文脈を表していると考え、その語と同義の語を翻訳に利用することにより、訳語の意味が変わることを防いでいる。

まず用語の定義を行う。  $i$  番目の機械翻訳サービス  $MT_i$  は入力文  $s_i$  と訳文  $t_i$  のペアで表す。  $MT_i$  を内部に持つ翻訳エージェント  $MTA_i$  は、入力文  $s_i$  を受け取り、訳文  $t_i$  のうち不適切な単語を置き換えて修正済みの訳文  $s_{i+1}$  を出力する。  $s_{i+1}$  は次の翻訳エージェント  $MTA_{i+1}$  の入力となる。 連携エージェント  $CA$  は、初めにユーザから原文  $s_1$  を受け取り、翻訳エージェント  $MTA_1$  から  $MTA_n$  までの連携を行い、最終的に目的言語の訳文  $s_{n+1}$  を受け取ってユーザに返す。 ある単語の  $n$  言語での同義語は、  $n$  個の単語の組である  $n$ -tuple  $(w_1, \dots, w_n)$  として表す。  $i$  番目の翻訳の時点の処理コンテキスト  $T_i$  は  $n$ -tuple の集合で表す。 各々の  $n$ -tuple のうち  $i$  番目の単語  $w_i$  は、  $i$  番目の翻訳エージェント  $MTA_i$  に渡される入力文  $s_i$  に含まれている。 つまり  $T_i$  は、原文の語と 1 番目から  $i-1$  番目までの翻訳で選択された単語  $(w_1, \dots, w_i)$  合計  $i$  個を含んでおり、残りの  $n-i$  個の単語  $(w_{i+1}, \dots, w_n)$  は  $i$  番目以降の翻訳で使用されるべき単語を示している。 これは、ある  $n$ -tuple  $(w_1, \dots, w_n)$  において、  $w_2, \dots, w_n$  は  $w_1$ 、ひいては原文  $s_1$  と同じ意味を持っているという考え方に基づいた実装である。

---

**Algorithm 3: SERVICE-AGENT  $MTA_i$** 

---

```
1:  $t_i$  /* Translated sentence */
2:  $MT_i = \{(s_i, t_i)\}$  /* A translation service; a set
   of pairs of source sentence  $s_i$ 
   and translated sentence  $t_i$  */
3:  $c_{i+1}$  /* A word in sentence  $t_i$  */
4:  $P_i$  /* A set of pairs  $(o_i, c_{i+1})$ , where  $o_i \in s_i$  and
    $c_{i+1} \in t_i$  */
5: when received (request,  $(s_i, T_i)$ ) from CA do
6:    $t_i \leftarrow MT_i(s_i)$ ;
7:    $P_i \leftarrow \text{GET-WORD-PAIRS-USED-}$ 
     BY-MT( $s_i, t_i$ );
8:    $Q_i \leftarrow \text{CREATE-WORD-PAIRS-TO-BE-}$ 
     USED( $P_i, T_i$ );
9:   if  $Q_i \neq P_i$  then
10:      $s_{i+1} \leftarrow \text{MODIFY-TRANSLATED-}$ 
       SENTENCE( $t_i, P_i, Q_i$ );
11:   else
12:      $s_{i+1} \leftarrow t_i$ ;
13:   end if;
14:   send (response,  $(s_{i+1}, Q_i)$ ) to CA;
15: end do;
```

---

図 4 : サービスエージェントのアルゴリズム

まず初めに、連携エージェントはユーザから受け取った原文  $s_1$  もとに、  $n$  言語の同義語の集合が蓄積された  $n$ -tuple レポジトリから同義語を抽出して最初の処理コンテキスト  $T_1$  を作成し、翻訳を開始する。  $i$  番目の翻訳エージェント  $MTA_i$  は翻訳処理を終え、訳文  $s_{i+1}$  と、入力文  $s_i$  の単語とそれに対して選択された訳語のペアの集合である  $Q_i$  を送り返してくるので、連携エージェントはそれを受け取り、次の  $i+1$  回目の翻訳のために処理コンテキストを更新して  $T_{i+1}$  を作成する。 処理コンテキストの更新は、  $T_i$  のうち  $i+1$  番目の語が訳文  $s_{i+1}$  に含まれる  $n$ -tuple のみを残すよう絞り込む処理となり、SELECT-POSSIBLE-N-TUPLES で行っている。  $T_{i+1}$  には曖昧性が含まれており、  $j$  番目の語  $(1 \leq j \leq i+1)$  が同じで  $i+2$  番目以降の語が異なる  $n$ -tuple が複数存在する場合がある。 そのような場合には、次の翻訳エージェント  $MTA_{i+1}$  が最も適切な語を選択する。 選択結果は訳語選択結果  $Q_{i+1}$  によって連携エージェント  $CA$  に通知され、  $CA$  はそれをさらに次の翻訳に反映する。

処理コンテキストと入力文を受け取った翻訳エージェント  $MTA_i$  は、Algorithm 2 に示すように、まず内部に持つ機械翻訳サービス  $MT_i$  を用いて翻訳を行い、GET-WORD-PAIRS-USED-BY-MT により入力文の原語  $o_i$  と訳語  $c_{i+1}$  のペアの集合  $P_i$  を獲得する。 この処理は、原文と訳文をそれぞれ形態素解析し、辞書を用いるなどして原語から訳語へのマッピングを行うことにより実装できる。 次に、CREATE-WORD-PAIRS-TO-BE-USED によ

り、受け取った処理コンテキスト  $T_i$  を用いて  $P_i$  の訳語を修正する。  $T_i$  にはそれまでの翻訳で使用された訳語とその同義語が含まれているため、  $T_i$  に含まれない訳語  $c_{i+1}$  は原文と異なる意味を持つと判断できる。 よって、そのような語は  $T_i$  に含まれる語で置き換えて、原語  $o_i$  と修正済みの訳語  $m_{i+1}$  のペアの集合  $Q_i$  を作成し、MODIFY-TRANSLATED-SENTENCE で訳文と  $P_i$ 、  $Q_i$  から修正済みの訳文  $s_{i+1}$  を作成する。 上述のように  $m_{i+1}$  の候補が複数ある場合の選択方法は様々であり、単語の頻度情報や優先度情報に基づいて選択することもできる。 また、翻訳対象文を含む文書全体や対話のログなどが利用できる場合には、その情報は連携エージェント  $CA$  が最初の文脈  $T_1$  を作成する際に反映される。

#### 4. 研究成果

実際に日英翻訳サービス・英独翻訳サービス・独英翻訳サービス・英日翻訳サービスを翻訳エージェントの協調作業による Web サービス連携技術で連結させ、日独折り返し翻訳サービスを構築した。提案した Web サービス連携技術の動作を検証するために、この折り返し翻訳サービスに対し予備評価を行った。品質の評価は、日本語の原文 A と、連携を行わずに翻訳した折り返し日本語文 B、連携を行って翻訳した折り返し日本語文 C をそれぞれ比較し、原文 A の意味を B および C がどの程度表現できているかを 5 段階 (5: All, 4: Most, 3: Much, 2: Little, 1: None) で評価するという方法で行った。原文は NTT の自然言語グループが提供する機械翻訳の性能評価用例文 (<http://www.kecl.ntt.co.jp/mtg/resources/index.php>) の 3718 文を用い、訳文 B と C が異なる結果となった 458 文から 100 文をランダムに選択した。評価者は日本人 3 人とした。訳文 B の評価結果と訳文 C の評価結果には、Welch の検定により 98% の信頼性で有意差が現れた。

連携技術を用いることで、平均で 100 文中 41 文 (41%) で評価値の向上が見られ、評価値は 100 文全体で 0.47 ポイント上昇した。図 5 の例では、日本語の単語“道”が“方法”

( 評価値が 4(Most) から 5(All) に改善された例 )

原文 A: トラックが 道を塞いでいた。  
 連携技術なしの訳文 B: トラックは 方法を妨げた。  
 連携技術ありの訳文 C: トラックは 道を妨げた。

( 評価値が 3(Much) から 5(All) に改善された例 )

原文 A: 社長は労働者を使う。  
 連携技術なしの訳文 B: 大統領は労働者を使う。  
 連携技術ありの訳文 C: 社長は労働者を使う。

図 5 : サービス連携技術の適用例

に誤訳されていた部分、および“社長”が“大統領”に誤訳されていた部分が改善されている。これらの誤訳は仲介した英単語“way”と“president”が多義であったために生じていた。また、連携技術を適用した訳文 C では、訳文 B において評価値が 4 であった文の 34%、3 であった文の 32%、2 であった文の 49%、1 であった文の 60% で評価値が改善された。

#### 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)

[雑誌論文] (計 1 件)

Rie Tanaka, Yohei Murakami, Toru Ishida, “Towards Coordination of Multiple Machine Translation Services,” in New Frontiers in Artificial Intelligence: JSAI 2008 Conference and Workshops, LNAI 5447, pp.73—86, 2009. 査読無し。

[学会発表] (計 1 件)

田仲理恵, “文脈情報を用いた機械翻訳サービスの連携” 第 22 回人工知能学会全国大会, 2008 年 6 月 12 日, 旭川市ときわ市民ホール。

[図書] (計 0 件)

[産業財産権]

○出願状況 (計 0 件)

○取得状況 (計 0 件)

[その他]

特になし

#### 6. 研究組織

##### (1) 研究代表者

村上 陽平 (MURAKAMI YOHEI)

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・研究員

研究者番号 : 00435786

##### (2) 研究分担者

( )

研究者番号 :

##### (3) 連携研究者

( )

研究者番号 :