3K-4

Web アプリケーションの Web サービス変換アーキテクチャとその評価

加藤 房良† 林 佑亮† 青山 幹雄†
南山大学 数理情報学部 情報通信学科†

1. はじめに

近年，Web サービスの基盤技術(SOAP，WSDL，UDDI)が成熟しつつある．しかし，利用可能な Web サービスが少ないため普及に至っていない．本研究では既存の豊富な Web アプリケーションを Web サービスに変換する方法を提案する[3].

2. Web サービス変換アーキテクチャ

2.1. 変換アーキテクチャの提案

Web アプリケーションを Web サービスに変換するアーキテクチャとして，以下の 2 つの方法が考えられる．

(1) Web アプリケーションに直接変換機能を追加
Web アプリケーションを構築しているサーバ上に，直接 Web サービス変換プログラムを作成する．

(2) Proxy サーバによるラッピング
サービスリクエスタと Web アプリケーションの間に Proxy サーバを置く．Proxy サーバに仲介機能(ラッパー)を置いて変換する．Proxy サーバと Web アプリケーションを 1 つのサービスプロバイダと見なす．

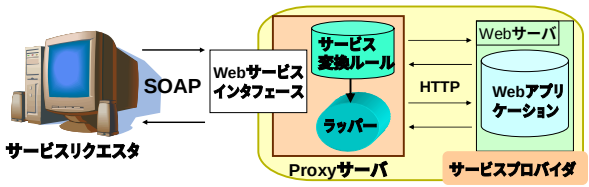

図 1 ラッピングアーキテクチャ

方法(2)は，Web アプリケーションの HTML レスポンスを利用して変換する方法をとるため，Web アプリケーションの実装言語に依存しない．また，方法(1)のように直接 Web サーバ内に変換プログラムを作成しなくて済むため，Web アプリケーションを一切変更せずに再利用できる．そこで本研究では，図 1 に示すように Proxy サーバによるラッピングアーキテクチャを用いる[1].

2.2. インタフェースの変換方法

ラッピングアーキテクチャは図 2 に示す，次の 2 つの要素で実現する．

(1) ラッピングタスク
ラッピングタスクは，Web アプリケーションと Web サービスのインタフェースの変換を行う．さらに Web アプリケーションから得られた HTML レスポンスを，Web サービスのレスポンスに変換する[2].

(2) Web サービス変換ルール
変換ルールは Web アプリケーションとの通信手順や Web アプリケーションと Web サービスのリクエスト，レスポンスの対応方法を規定する．変換ルールは XML で記述する．

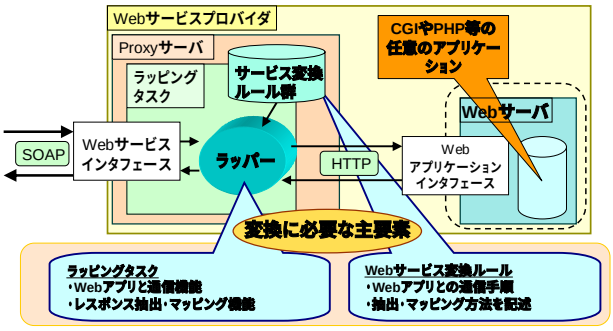

図 2 インタフェース変換方法

3. ラッピングタスクと変換ルール

3.1. ラッピングタスク

ラッピングタスクとして持つべき機能を以下に示す．また，図 3 にラッピングタスク内のデータフローを示す．

(1) リクエストジェネレータ
サービスリクエスタから受け取ったリクエストメッセージから Web アプリケーションへ送信する要素を抽出する．次に，Web アプリケーションの入力インタフェースである HTML 文書の FORM 要素の子要素のパラメータへ，対応づけて代入する．

(2) リクエストの送信
(1)で生成したリクエストを Web アプリケーションへ送信する．送信パターンは以下の 2 通りである．
1) リクエスト送信･レスポンス受信が一度だけ
2) 複数回にわたって送信･受信を繰り返す

(3) レスポンスの整形化
(2)によって得られた HTML レスポンスは必ずしも整形式であるとは言えない．レスポンスを抽出可能な形式にするために HTML 文書を整形式にする．本研究では HTMLTidy を使用し，HTML を走査し文法的な誤りを検出･修正し，整形式の XHTML にする．

(4) レスポンスパーサ
(3)によって整形式化された XHTML 文書からサービスリクエスタへのレスポンス要素を抽出する．

(5) レスポンスマッピング
(4)で抽出した要素をサービスリクエスタが処理可能な形式にマッピングする．

An Architecture of Translating from Web Applications to Web Services and its Evaluation
†Fusayoshi Kato, Yusuke Hayashi, Mikio Aoyama
†Faculty of Mathematical Sciences and Information Engineering, Nanzan University

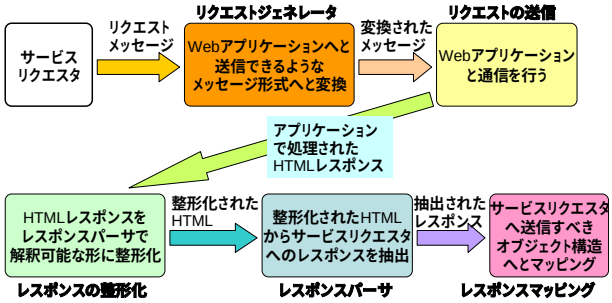

図 3　ラッピングタスク内のデータフロー

3.2. Web サービス変換ルール

Web サービス変換に必要なルールを XML で記述する. 変換ルールの適用プロセスを図 4 に示す.

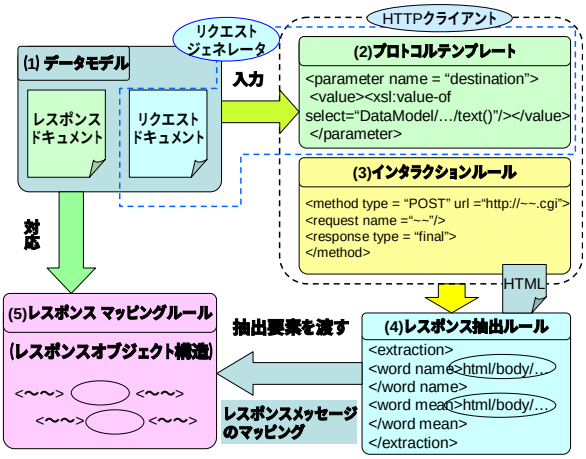

図 4　サービス変換ルールの適用プロセス

(1) データモデル

Web アプリケーションで使用されるデータを Web サービスで使用可能な形式のモデルとして記述する.

(2) プロトコルテンプレート

Web アプリケーションへ送信すべきリクエストを規定する. サービスリクエスタのリクエストを Web アプリケーションのリクエストであるパラメータ変数に対応付けてマッピングする.

(3) インタラクションルール

Proxy と Web アプリケーションとの通信を規定する.

(4) レスポンス抽出ルール

Web アプリケーションから得た HTML レスポンス内から, サービスリクエスタへのレスポンス要素が存在するロケーションへのパスを XPath で記述する. 抽出要素を XPath で指定した HTML レスポンスは XHTML 文書に整形式化する.

(5) レスポンスマッピングルール

HTML レスポンスの各要素をサービスリクエスタへのレスポンスの各要素にマッピングするために, 抽出要素間の対応関係を記述する.

ラッピングタスクとサービス変換ルールを用いた処理の流れを図 5 に示す.

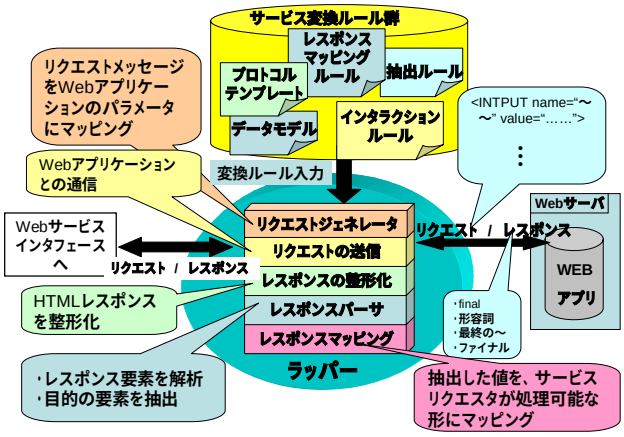

図 5　ラッパーの処理内容

4. 評価と考察

本研究で提案した変換方法を実現するラッパーのプロトタイプを Java で実装した. ここでは英単語の辞書 Web アプリケーションを例題とした. ラッピングタスクと変換ルールのコード行数を表 1 に示す.

表 1　ラッピングタスクと変換ルールの実装規模

サブシステム	ラッピングタスク	変換ルール
コード行数	約 400 行	約 200 行

ラッピングタスクは再利用できる. 異なる Web アプリケーションを変換する際には変換ルールのみを書き換えるだけ良いので, 開発コストの削減が期待できる. なお, 変換ルールの規模はラッピングする Web アプリケーションに依存する.

関連研究として, スクリプト言語を用いてラッピングをする方法がある. これは変換ルール記述がその実装言語に依存する. 本研究では実装言語に依存しない変換ルールを実現するため, XML で記述するアプローチを取った.

5. まとめと今後の課題

本研究では Web アプリケーションを Web サービスに変換するアーキテクチャとその実現方法としてラッピングタスク, サービス変換ルールに分離する方法を提案した.

今後の課題として, ラッピングルール作成の効率化を検討する.

参考文献

[1] 高橋 健一他, プロキシサーバによる WEB アプリケーションの WEB サービス変換, 情報処理学会ウィンターワークショップ 2005 論文集, 2005, pp. 95-96.

[2] TaMeX (A Task-structure Based Mediator for Information Integration through XML) , http://www.cs.ualberta.ca/~stroulia/TAMEX/

[3] 加藤 房良, 林 佑亮, Web アプリケーションの Web サービス変換に関する研究, 2005 年度南山大学数理情報学部情報通信学科 卒業論文, 2006.