XML 文書の差分を用いた SOAP 高速化

竹内　陽一† 　岡本　隆史†
株式会社　NTT データ　技術開発本部

1. はじめに

近年，電子商取引，省庁/企業間のシステム連携を行うための基盤技術として Web サービスが注目を集めている．ここで，Web サービスの標準プロトコルとして SOAP が用いられている．しかし，SOAP を用いたメッセージングでは，SOAP メッセージのパース処理部分がボトルネックとなるため，処理速度が遅く高速化が課題とされている．ここで，通常の SOAP 処理では，SOAP メッセージの XML のタグを全て解析している．しかし，SOAP メッセージ中の，サービスを実行するために必要な情報が含まれる部分は限られており，各 SOAP メッセージ全体を毎回パースするのは，処理に無駄が多い．

本稿では，SOAPメッセージの中で値が変化する部分が少ないことに着目し，変化しない部分からテンプレートを作成しておき，SOAPメッセージとテンプレートとの差分のみを抽出することによる，SOAPメッセージの処理の高速化を提案する．

2. SOAP 高速化の従来手法の問題点

2.1 従来手法の問題点

Web サービスにおいて，クライアント，サービスプロバイダの両側で，SOAP メッセージのパージングと生成の処理が行われる．そして，この二つの処理が SOAP メッセージ処理のボトルネックになると言われている．よって，SOAP メッセージの処理（生成／パージング），つまり，XML の処理（生成／パージング）を改善する事が課題となる．ここで，SOAP 高速化の従来手法としては，XML パーサを高速化する手法[1]，タグの省略による手法[2]がある．ここでは，妥当性検証を伴う，XML のパース処理における問題として，以下の三つの問題点を挙げる．

問題点 1　XML 文書のパース処理において，各タグを一つ一つ処理する点

XML 文書を利用した場合，アプリケーションに必要な値が書き込まれている部分(例えば<info><name>竹内陽一</name></info>における「竹内陽一」の部分)は，XML 文書全体に比べると少ない．しかし，XML のパース処理において，全てのタグを１つ１つ処理する必要がある．

問題点 2　XML 文書のタグ毎に妥当性検証を行う点

XML 文書のタグ毎に，XML スキーマのフォーマットを定義したと一致しているかどうか調べるため，処理速度が遅い．

問題点 3　XML 文書のパース処理と妥当性検証を別々に行う点

従来の手法では，XML 文書のタグの一つ一つを抽出してから，XMLSchema を用いて妥当性の検証を行っていた．つまり「XML 文書のパース処理と妥当性検証を別々に行う事」が問題である．

A Method of Accelerating SOAP Messaging Based on Difference of XML Documents
†Yoich Takeuchi, Takashi Okamoto
†NTT DATA CORPORATION Research and Development Headquaters

3. テンプレートに基づく SOAP 高速化の提案

3.1 提案手法の基本的アプローチ

Web サービスで利用される SOAP メッセージの性質に着目すると，メッセージの中でシステムが処理に利用する情報は値部分のみであり，メッセージの中で極一部に過ぎない．そして，値以外の残りのタグ情報は，この文書構造を表すために利用されているため，システムには必要ない．また，SOAP メッセージの用途が決まれば，そのフォーマットは多くの場合固定的であり，殆んど同じ SOAP メッセージがシステムで利用される．ここで，殆ど重複した SOAP メッセージに対して，毎回 XML のパースを行うと無駄が多い．

そこで，本稿では，SOAP メッセージの中で値が変化する部分が少ないことに着目し，変化しない部分からテンプレートを作成しておき，SOAP メッセージとテンプレートとの差分のみを抽出することによる，SOAP メッセージの処理の高速化を提案する．全体の処理の概要は，最初に，SOAP メッセージの中で値が変化しない部分を定型化し，変化する部分を変数で表現する事により，テンプレートを作成する．テンプレートは，実際に値が変化する部分を，＄の記号と変数名で表している．そして，このテンプレートを用いて，受信した SOAP メッセージに対して，テンプレートと比較を行い，＄変数名に該当する値部分のみを取り出す．（図 1参照）

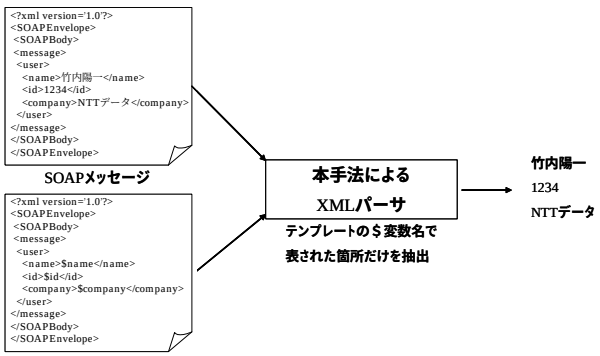

図 1　本手法の基本的アイデア

次に，具体的なパース処理のイメージを図 2を用いて説明する．

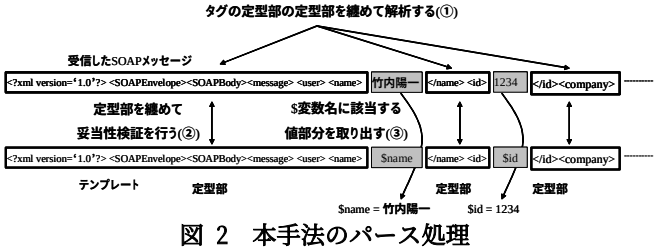

図 2　本手法のパース処理

図 2の様に，受信した SOAP メッセージに対して，タグの定型部を纏めて読み込む（図 2-①参照）と同時に，テ

ンプレートの定型部とマッチングするかどうか調べる事により妥当性の検証を行う（図 2-②参照）．そして，マッチングした場合には＄変数名に該当する値部分のみを取り出す（図 2-③参照）事によりパース処理を行う．

続いて，提案手法により，従来の問題点をどのように解決できるか説明する．

問題点 1 に関して，本手法では，テンプレートを用いる事で，定型部を纏めて解析する事ができるため，高速化を図る事が可能となる．例えば，$name に該当する値部分である“竹内陽一”と，$id に該当する値部分である“1234”の間のタグの文字列“</name><id>”を定型部とみなし，纏めて解析する（図 2-①参照）．

問題点 2 に関して，本手法では，XMLSchema を用いる代わりに，テンプレートを用いて妥当性検証を行う．テンプレートを用いる事で，定型部を纏めて妥当性検証を行う事ができるため，高速化を図る事が可能となる（図 2-②参照）．

問題点 3 に関して，本手法では，テンプレートを用いる事で，タグの定型部を纏めて読み込むと同時に，テンプレートの定型部とマッチングするかどうか調べる事により，妥当性の検証を行う事が可能である．つまり，本手法では，「XML 文書のパース処理と妥当性検証を同時に行う事が可能」であるため，高速化を図る事ができる．

3.2 テンプレートの表現方式

本手法を実現するにあたり，テンプレートを固定的に設計すると，XML 文書に頻繁に見られるタグの繰り返しに対応できないという問題がある．そこで，オートマトンを利用する事により，タグの繰り返し部分に対応させた．本手法では以下の方法によりテンプレートをオートマトンであらわす．

- テンプレートの各変数をオートマトンの各状態に対応させる
- 変数と変数の間にあるタグの文字列の定型部をオートマトンの遷移条件とする

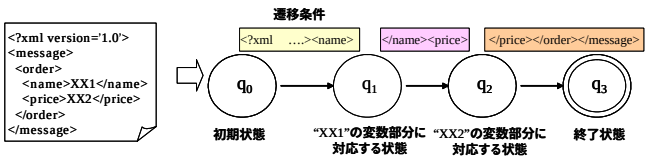

図 3 オートマトンにおける状態と遷移条件

例えば，図 3を用いて説明する．XML 文書の中で値「XX1」と「XX2」の部分は，テンプレートにおいて変数で表現され，オートマトン上ではそれぞれ，状態 q_1，状態 q_2 に対応する．また，「XX1」と「XX2」間の文字列「</name><price>」が状態 q_1 から状態 q_2 への遷移条件となる．なお，初期状態と終端状態は，XML 文書の開始と終了を表す．この方法により，例えば，図 4の左の XML 文書は，右のオートマトンであらわされる．ここで，図 4の XML 文書において，タグ<order>の後に，タグ<name>とタグ<price>の組み合わせが三回繰り返して現れている．そして，この部分は，オートマトンにおいて，状態 q_1→状態 q_2→状態 q_1 の遷移を繰り返す事に置き換えられる．よって，タグの繰り返しに対してもオートマトン上での遷移の繰り返しにより表現できるため，テンプレートを汎用的に用いる事ができる．

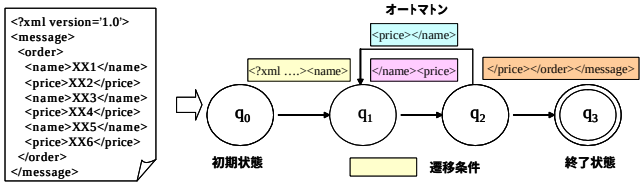

図 4 XML 文書からオートマトンへの変換

3.3 パース処理

上記により作成したオートマトンを利用して SOAP 文書のパース処理を行う処理方法について述べる．パース処理は，オートマトンで文字列を受理する事による状態遷移により行う（図 5参照）．処理手順は，最初に，オートマトンの先頭のノードから次のノードへの遷移条件を取り出す．そして，その値と SOAP メッセージの先頭部分からの文字列を比較し，一致するかどうか調べる（図 5-①参照）．一致した場合には，値部分を取り出し（図 5-②参照），オートマトンの次のノードへ遷移する（図 5-③参照）．そして，次の状態へ遷移し文字列を受理すると言った操作を繰り返し，終端の状態へ辿りつけば，パース処理は終了となる．

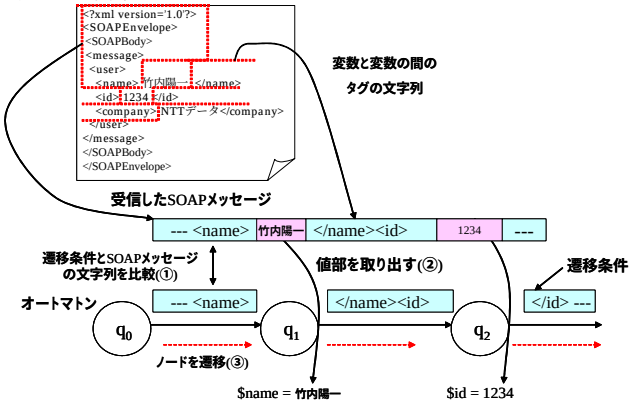

図 5 パース処理

本手法の有効性を図 5の例を用いて確認する．図 5において，受信した SOAP メッセージは，18 個のノードから構成される．そして，従来手法を用いてパース処理を行った場合，18 個のノードを一つ一つ解析する必要がある．しかし，本手法を用いるとタグの定型部を纏める事で，7 個のブロックを解析するだけでよい．よって，高速化を図る事が可能となる．

4. おわりに

本稿では，SOAP メッセージの中で値が変化する部分が少ないことに着目し，変化しない部分からテンプレートを作成し，SOAP メッセージとテンプレートとの差分のみを抽出することによる，SOAP メッセージの処理の高速化を提案した．今後は，本提案方式によるパース処理の実装を行い，従来手法との比較評価を行う．

参考文献

[1] piccolo Parser:http://piccolo.sourceforge.net/
[2] 白浜哲他：Web サービス技術を基盤とする GridRPC システムの評価,情報処理学会研究報告,2002-HPC-91,pp.197-202