

Web2.0 におけるマッシュアップ標準化手法の検討

秋永 智[†] 木村 昌臣[†]芝浦工業大学工学部情報工学科[†]

1. はじめに

近年インターネット上では Web2.0 と呼ばれる新しい概念が生まれた。これにより新しいタイプのサービスや技術を構築することが可能となり、そのひとつとして企業や個人が WebAPI と呼ばれるコンテンツやサービスを複数組み合わせる新たなサービスを作り出すことが可能となった。この技術をマッシュアップと呼び、Web2.0 の技術の一つとなっている。しかし、このマッシュアップを構成する WebAPI は提供元によって仕様が異なっている。つまり新たな WebAPI を利用したサービスを実現するには最初からプログラムを作り直す必要があるため、WebAPI ごとに別々に扱わなければならないという問題がある。そこで本研究では WebAPI でやりとりするデータのフォーマットを自動的に変換し構造を同一にすることでどの WebAPI でも統一的に利用できる環境を構築し、サービスの作成を容易に実現できる手法を検討・開発する。

2. WebAPI の問題点と提案手法

WebAPI を利用したアプリケーション・サービスを構築する際にはそこから返される XML 形式のデータを利用していく。XML はファイルの製作者が自由にフォーマットを決められるため、WebAPI の提供元によって XML の構造や構造を規定するタグの名前が異なっている。そのため互換性を持っていない。しかし類似した情報を扱っている XML ならば構造などは違っていても同じ情報を持っている部分があることが期待される。そこで似たような情報を持つ複数の XML を比較し、そのどちらにもあるデータだけを抜き出して同一構造を持つ XML に変換する。XML の変換にはファイルの構造変換をすることにより異なるフォーマットに変換する XSL と呼ばれる技術を用いる。

3. システムの実装

前処理として、比較するそれぞれの XML 文書中で同じ情報を持っているタグをテキストファイルに記述してデータ同士の対応付けを行う。

図 1 にその対応付けを記述したものを示す。図 1

An inquest of standardization method for mashups on Web2.0

[†] Satoshi Akinaga

[†] Masaomi Kimura

[†] Shibaura institute of technology

では同一行にあるタグ名が同じ情報を保有していることを示している。

response	Results
total_hit_count	NumberOfResults
hit_per_page	DisplayPerPage
rest	Shop
id	ShopIdFront
name	ShopName
name_kana	ShopNameKana
address	ShopAddress
station	StationName
latitude	Latitude
longitude	Longitude

図 1 同一内容を保持するタグの対応付け

ここで Web サービスから得られる XML を扱うための前提として、①一つの XML 中に同名のタグが二つ以上存在しない、②葉ノード同士で対応がつくものがある、③葉ノードのみにデータを持つ、とする。また今回の研究ではスキーマ、名前空間や、タグに付加している属性値は扱わないとする。まずそれぞれの XML から DOM ツリーを構築し、全ノードを走査して葉ノードまでの XPath を全て求め保存する。XML を比較するため葉ノードの要素を各 XML から一つずつ取り出して組み合わせを作り、さらに XML から構築した DOM ツリーで見たときにその葉ノードの組み合わせが作成した表のそれと同じになるような組み合わせを探索していく。同じ組み合わせのものを発見したらその部分を格納しておき、その要素を起点としてそれぞれの DOM ツリーを上位へと辿っていく。ツリーを辿っていきながら先ほどと同様に上位のタグ中に同じ組み合わせがあるかどうかを調べていく。起点としたノードの上位で対応する組み合わせを発見したらその部分にまた格納し、さらにそこを起点としてまた上位へと辿っていく。この処理を繰り返してルートノードまで到達したら今までに格納しておいたタグを「/」区切りで繋ぎ合わせて Xpath を作る。この手法を利用して対応付けがついている葉ノードまでの Xpath を全て作っていく。抜き出したタグで作った Xpath の構造は同一のものになっている。

次に構造を同一にした Xpath を利用し、同じ構造を持つ XML に変換するための XSL ファイルを生成する。XML の構造を作るためにあらかじめリスト構造を用意しておき、XML の構造が先ほど作った Xpath になるように各タグの開始タグ、終了タグをリスト構造に挿入、配置し XML の入れ子構造を表現していく。また各タグの挿入後にテキストデータを反映させるための XSL 要素をそれぞれの開始タグと終了タグの間に挿入し、XSL を XML に適用したときに自動的にテキストデータがタグの間に記述されるようにする。そしてリスト構造に格納されているデータを XSL ファイルとして生成する。最後に先ほど生成された XSL ファイルをそれぞれの XML に適用し、同じ構造を持つ XML 文書を生成する。

4. システムの評価

インターネット上で利用できる WebAPI を任意に選び、取り扱う情報が同じ XML 二つを一組として上記の提案手法で変換した結果、生成した XML が正しく変換できているか、同時に生成した XML を比較して同一の構造になっているかを検証する。また、XML を変換する実行時間を測定する。なお対象とする WebAPI は表 1 の通りである。

表 1 対象とする WebAPI

Web サービス(WebAPI)名	取り扱う情報
ホットペッパーWeb サービス	レストラン
ぐるなび Web サービス	レストラン
食べログ.com Web サービス	レストラン
楽天 Web Service	商品
価格.com Web サービス	商品
Youtube API	動画
AmebaVision API	動画
Yahoo!動画検索 Web サービス	動画
Sagool API	ウェブ検索
Yahoo!ウェブ検索 Web サービス	ウェブ検索

5. 評価結果

変換の結果、レストラン情報を扱うものは 3 件中 3 件、商品情報を扱うものは 1 件中 1 件、動画情報を扱うものは 3 件中 1 件、ウェブ検索情報を扱うものは 1 件中 0 件が正常に変換し、成功した割合は 62.5%となった。しかし名前空間が設定されている Yahoo!が提供する各 WebAPI、及び一つの XML 文書中に同名のタグが複数存在している AmebaVision が先に提示した前提に当てはまらないためそれらを除外した結果、成功した割合は 100%となり、生成した XML を比較したところ同一の構造が得られ、対応付けがしてある要素同士の各テキストデータは同じ内容で

あることが確認できた。変換にかかった時間は 500ms 程度のものが一番多く、変換にかかった時間は 300~700ms の間だった。また変換にかかる時間を測定した結果のグラフを図 2 に示す。なお実行環境は Pentium4 2.66Ghz, メモリ 1GB, WindowsXP SP2 である。

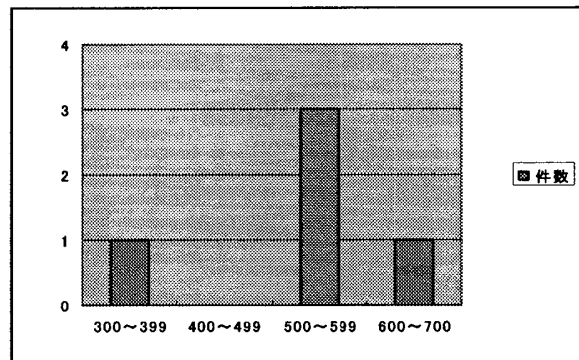


図 2 実行結果グラフ

実行時間は XML を読み込み、XSL を生成してそれを適用し新たな XML が生成されるまでの時間を計測した。実行時間の平均値は 505.6ms となり変換元 XML の文書量や生成するファイルに多少依存するものの極端な時間差は出ず、およそ 500ms 前後で変換できることが分かった。

6. おわりに

本研究では WebAPI から提供される XML の構造を統一して互換性を持たせ、仕様に関わらず任意に利用できるような手法を検討した。現在ではいくつか前提があり、それらに当てはまらないものがあるため今後前提に依存せずに動作するものを作る必要がある。また対応付けを自動で行い、Web 上で動作させることによって実用的なシステムが可能になると考えている。またどちらかにしかないデータをどのように反映させるかを考える。

参考文献

- 1) 横山昌平, 的野晃整, サイドミルザパレビ, 小島功: Web2.0 における JavaScript コードのモジュール化とマッシュアップの枠組み, 日本データベース学会 Letters Vol. 5, No. 3, pp. 1-4 (2006)
- 2) Tim O'Reilly, Web 2.0: 次世代ソフトウェアのデザインパターンとビジネスモデル
<http://www.oreillynet.com/lpt/a/6228>
- 3) Web サービス・API ディレクトリ - モジュールスイート,
<http://www.modulesuite.com/jp/wsd/>