

Web サービス連携のための XML マージ処理方式の実験と評価

XML Merging Method for Web Service Integration

西田 晋平[†]中所 武司[†]

Shinpei Nishida

Takeshi Chusho

1 はじめに

近年, 既存の Web サービスを連携させて, 迅速にアプリケーションを構築する技法が注目されており, このようなソフトウェアのサービス化の傾向は今後加速されると思われる. エンドユーザが考案したビジネスモデルを実現するアプリケーションをサービスの組み合わせで短期間に開発し, ビジネスモデルの変更に対してもタイムリーに保守を行っていくという方法には, エンドユーザ主導開発 [1] が適している.

われわれはエンドユーザ主導の Web サービス連携の研究の一環として, これまでに, 学生向け個人別試験時間割表システムの開発を通して, 複数の XML 文書のマージ処理を XSLT で記述する方式を実現した.

本稿では, より複雑なマージ処理を伴う例題として学生向け卒業条件判定システムを考え, この開発を通して XML マージ処理方式の実験と評価を行った.

2 基本的なマージ処理方式

最初の例題としてとりあげた個人別試験時間割表システムは, 試験の実施予定が記述された試験時間割表 XML と, 学内システム Oh-o!Meiji[2] で提供される個人時間割表 HTML から, 各個人に特化した個人別試験時間割表 HTML を作成し, 学生に提供するシステムである. その XML マージ処理は以下のような処理になる.

1. 個人別時間割表 HTML から得られた履修状況 XML と試験時間割表 XML から比較対象となる要素(この例では科目要素)を一つずつ取り出す.
2. その要素内の比較したい要素(この例では科目名と教員名)の持つデータを比較し, 同じ場合は, 指定した形式に従ってデータを出力する.
3. 処理 1,2 を比較対象要素の全ての組み合わせについて行う.

この処理を XSLT によって記述するために, 外部の XML 文書へのアクセスを可能にする XPath の document 関数を用いて, ひとつの XSLT で複数の入力を取り扱えるようにした. また, ある親要素の直下の子である同じ要素の中でのノードの位置を返す position 関数を用いて, 現在比較している要素の位置を変数に保持し, 出力データの指定に利用するようにした.

図 1 に示すように, この XSLT を入力 XML へ適用することで XML マージ処理を行う.

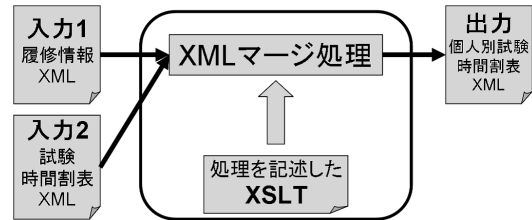


図 1: 個人別試験時間割表システムの XML マージ処理

3 複雑なビジネスロジックのマージ処理方式

3.1 システム概要

本稿では, より複雑なマージ処理を必要とする卒業条件判定システムを考えた. これは, 明治大学の学内システム Oh-o!Meiji[2] から得られる, 修得済み科目の名前と単位数と成績が記載された個人別成績表と, 卒業までに必要な単位数や必修科目, 選択必修科目等の条件が記載された卒業条件表の情報をマージし, 各学生に卒業するために修得すべき単位の情報を提供するシステムである.

3.2 卒業条件判定システムの特徴

卒業条件判定システムは, 入力 XML から直接得られる一次情報だけでなく, 入力 XML の情報を組み合わせて加工した二次以上の情報も出力情報に含むという点で複雑である. 例えば, 必修科目の総修得単位数は, 履修済み必修科目の単位数の合計なので, 個人別成績表 XML から得られる科目の履修済み情報や単位数と, 卒業条件 XML から得られる科目が必修科目か否かという一次情報の組み合わせから得られる二次情報である.

3.3 XSLT による記述

この特徴を考慮した XML マージ処理を XSLT で記述する. 前述の必修科目の総修得単位数のような二次の情報が含まれる場合を考える. C や Java などを用いたプログラムであれば, 総修得単位数をあらわす変数を用意し, 各科目を調べて, 科目が必修かつ履修済みであれば, その単位数を変数に足しこむという方法が考えられる. しかし, XSLT では更新可能な変数が無いため, このような方法が取れない.

そこで, 回数毎に処理を区切ることを考える. すなわち, まず一次情報を取得して一次情報を含んだ中間出力 XML を生成し, 中間出力 XML を用いて二次情報を得るようにする.

一次情報を取得する XSLT は以下のような処理手順となる.

1. 入力となる 2 つの XML ファイルのノード集合を変数へ格納する.
2. 一次情報を取得し, 出力 XML に埋めこむ.

[†]明治大学 大学院 理工学研究科 情報科学系 ソフトウェア工学研究室

この一次の中間出力 XML には、2 つの入力ファイルに含まれる情報に加えて、例えば、個人別成績表 XML に記載されている科目の履修済み情報などが新たに埋め込まれる。

二次以上の情報を取得する XSLT は以下のような処理手順となる。

1. 前の次数の中間出力 XML を読み込む。
2. 現在の次数の情報を取得して、出力 XML に埋め込む。

例えば二次の場合、一次の中間出力 XML を読み込み、必修科目の総修得単位数を計算して二次の中間出力 XML に埋め込む。

このように、前の次数の情報を埋め込んだ中間出力 XML を作成することで、XSLT を用いて二次以上の情報を得ることができる。

上記の処理手順を記述した XSLT 用意し、図 2 のように適用することで XML マージ処理を行う。

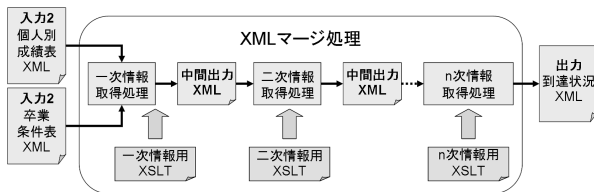


図 2: 卒業条件判定システムの XML マージ処理

試作した XSLT は、入力 XML のルート要素、科目区分、必修科目、選択必修科目の各要素に対応する、4 つのテンプレートを含む。マージ処理で入力要素を読み込むたびに呼び出されるテンプレートが呼ばれ、情報の取得と出力作業を行う。例えば、二次情報用 XSLT の必修科目要素に対応したテンプレートは以下ようになる。

```
<?xml version="1.0"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:output encoding="UTF-8" indent="yes" method="xml" version="1.0"/>

<xsl:template match="必修科目">
<xsl:element name="必修科目">
<xsl:attribute name="条件単位数">
<xsl:value-of select="current()/@条件単位数"/>
</xsl:attribute>
<xsl:attribute name="大小関係種別">
<xsl:value-of select="current()/@大小関係種別"/>
</xsl:attribute>
<xsl:attribute name="総修得単位数">
<xsl:value-of select="sum(current()/科目 [@済 = &quot;true&quot;] /@単位数)"/>
</xsl:attribute>
<xsl:attribute name="完全達成">
<xsl:value-of select="not(current()/科目 [@済 = &quot;false&quot;])"/>
</xsl:attribute>
<xsl:for-each select="current()/科目">
<xsl:element name="科目">
<xsl:attribute name="id"><xsl:value-of select="current()/@id"/></xsl:attribute>
<xsl:attribute name="単位数">
<xsl:value-of select="current()/@単位数"/>
</xsl:attribute>
<xsl:attribute name="済"><xsl:value-of select="current()/@済"/></xsl:attribute>
</xsl:element>
</xsl:for-each>
</xsl:element>
</xsl:template>

</xsl:stylesheet>
```

必修科目は、条件単位数と、条件単位数に対する総修得単位数の関係を指定する大小関係種別と、総修得単位数と、完全達成の 4 つの属性を持ち、子に科目を持つ。科目は、科目名と、単位数と、履修済みか否かを表す済の 3 つの属性を持つ。

まず一次情報の条件単位数と大小関係種別をそのまま出力している。

次に総修得単位数は、必修科目に含まれる、済属性が true である科目の単位数の総和として得られ、以下のような XPath 式で表される。

`sum(current()/科目 [@済 = "true"] /@単位数)`
 式中の sum 関数は引数の総和を計算する関数で、current 関数は現在の要素（ここでは必修科目）を得る関数である。また“科目 [@済 = "true"]”が、済属性が true である科目を表している。この式によって表される値を xsl:value-of を用いて出力している。

次に完全達成は、必修科目に含まれる、済属性が false の科目要素が存在しなければ true、存在すれば false とし得られ、以下のような XPath 式で表される。この値を xsl:value-of で出力している。

`not(current()/科目 [@済 = "false"])`
 式中の not 関数は引数の論理否定を返す関数である。

次に xsl:for-each によって必修科目に含まれるすべての科目要素の情報を取得して出力する。情報はいずれも一次なので、そのまま出力している。

4 おわりに

本稿では XSLT を用いた XML マージ処理のスク립ト化により、プログラミング言語を用いるよりも容易に Web サービスを連携させることができることを確認した。しかし、XSLT では、XPath を用いてデータの場所を指し示したり、テンプレートや変数の概念が必要であるなど、エンドユーザにとって困難な部分が残る。このため、ビジュアルツールによる XSLT の記述支援などがエンドユーザ主導開発にとって必要である。

参考文献

- [1] Takeshi Chusho, Katsuya Fujiwara, Hisashi Ishigure and Kei Shimada : A Form-based Approach for Web Services by Enduser-Initiative Application Development, SAINT2002 Workshop (Web Service Engineering), IEEE Computer Society, pp.196-203 (Feb. 2002).
- [2] 明治大学 : Oh-o!Meiji
<http://www.oh-o.meiji.ac.jp/>
- [3] 湯浅亮祐, 中所武司 : Web サービス連携のための Web ページラッピング技術の提案と評価, 情報処理学会 ソフトウェア工学研究会資料, 2004-SE-144, pp.115-122 (Mar. 2004).
- [4] 西田晋平, 湯浅亮祐, 中所武司 : エンドユーザ主導開発を志向した Web サービス連携方式の実験と評価, 情報処理学会 ソフトウェアジャパン 2004, pp.55-62 (Oct. 2004).
- [5] Michael Kay : XSLT パイブル, インプレス (2002).
- [6] Microsoft : Microsoft XML Parser (MSXML) 3.0 Service Pack 4,
<http://www.microsoft.com/japan/内ダウンロードセンターより> (Aug. 2003).