D-29　XML文書の事前形式変換によるデータ処理性能改善の検討

Improving Performance of Processing XML Documets by Format Transformation in Advance

吉田 茂†　Shigeru Yoshida
矢作裕紀†　Hironori Yahagi
井谷宣子†　Noriko Itani

1．まえがき

近年，伝票や顧客名簿等，レコード構成のデータへのXML適用がBtoB用途で広がっている。XMLはデータ表現力，柔軟性，拡張性が高い反面，データ処理時にはその冗長性が負担になる。現状は，数千レコード(数MB)以上のXML文書は，通常，RDBに格納してデータ処理する。XML文書をランダムに操作する場合，W3C標準のAPIであるDOM [1]を用いるが，DOMによるデータ処理は，CPU負荷が高く，XMLファイル容量の5～10倍もの大量のメモリを消費し，メモリ上で大容量XML文書を扱うには課題があった。本稿では，レコード構成のXML文書に対してDOM応用ソフトでのデータ処理性能を改善するため，XML文書を別な形式のXML文書に事前に変換する方法を検討・評価したので，結果を報告する。

2．従来技術と狙い

負荷を軽減するXML形式として，携帯機器のWAP [2]で使われているバイナリXMLが知られている。バイナリXMLは特定用途には有効だが，全ての用途向けに一般化したバイナリ符号を作るのは困難であり[3]，新たに専用DOMソフトも必要であり，W3C XML規格と整合が難しい。本検討では，実用性を重視し，応用ソフトにわずかな修正で適用できて，変換後に基本的に元のXML文書と同様(トランスペアレント)に扱える形式を目指すことにした。

XMLの導入現場では，既存データをXML化することで処理能力不足が生じる場合も多く，これを補うため，ユーザーによって個々にXML文書の形式を修正する幾つかの方策が採られている[4]。基本的な方策は，(a) XMLの定義情報の長さを短くする，(b) 冗長な情報を一つにまとめる，(c) 業務に不要なXMLデータを取り除く，である。

XML文書の要素名を省いてデータ量を圧縮するため，Trotter [5]は，レコード毎に全要素内容をCSV(Comma Separated Values)形式で繋いで1個の要素にまとめたXML文書に変換する方法を「ZXML」と称して提案した。ZXMLでは，XML文書の変換はXSL変換[6]として実行し，XML文書がレコード内1階層の場合の簡単な例を示した。変換XML文書のデータ処理には専用ソフトを用いることを想定した。

以降では，これらの従来技術をデータ処理性能改善のツールとして使える汎用手法に高め，XML文書の事前変換後の形式と，その変換の実行法について述べる。

3．技術開発

3.1 XML文書の事前変換形式

DOM処理はXML文書の全要素をメモリ上に展開するため負荷が重くなるが，その応用ソフトでは全種の要素をレコード横断のデータ処理の対象にすることはまれである。冗長な情報を一つにまとめて要素数を減らすことが有効なので，レコード毎に，応用ソフトで処理対象とする要素はそのままにして，処理対象としない複数の要素の要素内容をCSV形式で一つの要素にまとめたXML文書に変換する(図1 (a), (b))。これにより，冗長な要素名が省かれ，要素数が削減されて，DOM展開時の負荷が軽減される。本方式を「CSV圧縮変換」と呼ぶ。

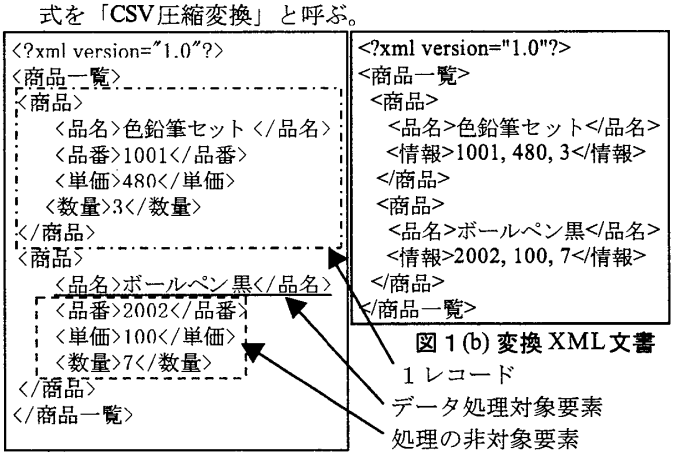


図1 (b) 変換XML文書

図1 (a) 変換前XML文書の例

3.2 CSV圧縮変換の実行

XML→XML文書形式変換であるCSV圧縮変換を，「XSL変換」として実行する(図2)。特別なソフトを作らず変換が実行できるように，処理対象要素名と非対象要素名を列挙した「変換仕様XML文書」(図3)をユーザが作成すれば，XSL変換/逆変換に用いるXSLシートを自動的に生成できるようにする。この自動生成も，XSLシートを用いて，XSL変換として実行する。

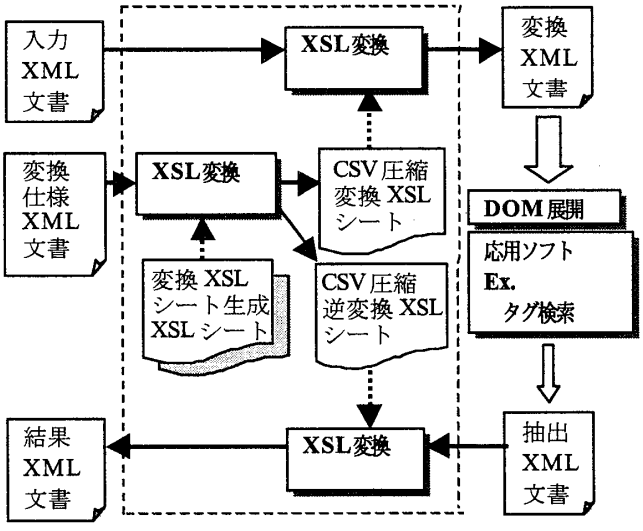


図2．変換XSLシート生成，CSV圧縮変換，及び，応用ソフトへの適用の全体フロー

† (株) 富士通研究所 ペリフェラルシステム研究所

```
<?xml version="1.0"?>
<transform>
 <structure>
     <root>商品一覧</root>
     <record>商品</record>
</structure>
 <items>
   <merging_tag>情報</ merging_tag >
     <key>品名</key>
     <nonkey>品番</nonkey>
     <nonkey>単価</nonkey>
     <nonkey>数量</nonkey>
 </items>
</transform>
```

各タグによる指定

<root>：
ルート要素名

<record>：
レコード要素名

<merging_tag>：
変換先CSV形式の
要素名

<key>：
処理対象要素名

<nonkey>：
処理非対象要素名

図３．　図１(a)から図１(b)への変換仕様ＸＭＬ文書の例

図１(b) のような表形式データでは，処理非対象要素名は変換/逆変換XSLシートの中に吸収され，変換XML文書中から省かれる。この作用によって，変換 XML 文書のデータ量は，レコード毎に全要素をCSV形式で１要素にまとめた場合，元の最大約1/3に圧縮される。

3.3 複雑なXML文書の扱い

XML 文書の特徴は，非表形式や，深い階層構造のデータが表現できることである。複雑な XML 文書は次のようにして扱うことができる。これらは変換仕様 XML 文書で指定して図２と同様に変換XSLシートを生成すればいい。

(1)非表形式のデータの変換形式

レコード内に出現する要素が不定であるため，変換XML文書上に，処理非対象の要素名を保持して，要素内容と対応付ける必要がある。CSV形式の要素内容と同じ並び順で，処理非対象要素名を CSV 形式で繋いで，変換先CSV形式の要素の属性として置く。

CSV 圧縮変換によって非表形式データの要素数を最大限に減らしたいときは，レコード内の要素を，固定して現れる要素と，不定の要素との２グループに分け，それぞれを変換先の表形式の CSV 要素と， CSV 形式の要素名を属性に置いた非表形式のCSV要素，の２個に格納すればいい。

(2)レコード内２階層以上，属性を持つ場合の変換形式

表形式データの場合は前述と同様に，処理非対象要素の要素名，属性名は変換XSLシートに吸収される。非表形式データの場合は(1)で述べた変換先 CSV 形式の要素の属性に，処理非対象要素や属性のレコード内の階層位置をXPath で表して，CSV 形式で繋いだものを置く。そのXPath の表現が長くなる場合は，XPath を短縮名に置換えれば，データ量を増やさずに一意に識別することができる。

4．入出力性能の改善効果

レコード内２階層の XML文書が扱える変換/逆変換 XSLシート生成XSLシートを作成して，方法論の確認と，変換XML文書の性能評価を行なった。CSV 形式にまとめて減らした要素数の割合にほぼ比例して，DOM のメモリ使用量，メモリへの展開時間，DOM 木を辿る時間が削減された（図４(a),(b),(c)）。CSV 圧縮変換は，同じ評価環境で，レコード毎に全要素を 1 個の CSV 要素にまとめたとき10MBのXML文書の変換が20秒，逆変換が８秒であった。

5．むすび

本技術の特長は，応用ソフトでデータ処理の対象にする要素の種類が予め分かれば XML 文書を必要最小限のDOM 木で扱えること，XSL 変換として実行するため種々の言語で利用できること，特別なソフトを書かずに変換が実行できること，である。種々の実際の応用で本技術の効果を確認することが今後の課題である。XML 技術は普及途上であり，様々な便利ツールが必要とされるが，本技術もその一端を担うものと考える。

参考文献

[1] DOM http://www.w3.org/DOM

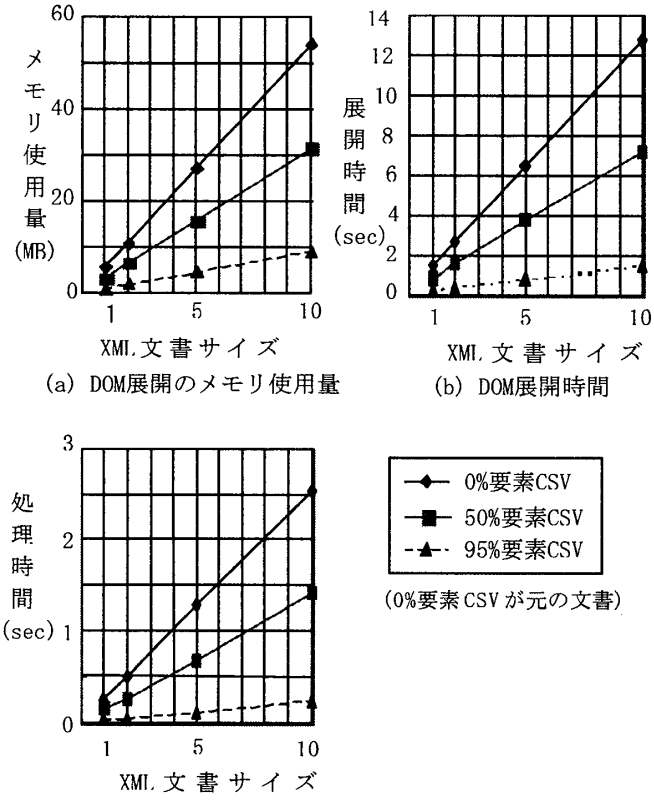
[2] WAP(The Wireless Application Protocol), http://www.atmarkit.co.jp/aig/01xml/wap.html

[3] "Intuition and Binary XML", http://www.xml.com/pub/a/2001/04/18/binaryXML.html, 2001.04.16

[4] 「見えてきた万能幻想の真実　XMLの”常識”を覆す」pp. 52-71, 日経コンピュータ 2001.2.26 号

[5] Alain Trotter,"Building an XML Bloat Buster using ZXML -- XML Compression Mehod", 有料情報配信 Web サイト http://www.asptoday.com 2001.1.3 の記事概要は http://www.xml.com/pub/r/904

[6] XSLT http://www.w3.org/TR/xslt



(a) DOM展開のメモリ使用量　(b) DOM展開時間

(c) DOM木全体を辿る時間

(0%要素CSVが元の文書)

図４．CSV圧縮変換の性能評価

評価環境：PentiumII 450MHz Windows2000 マシン
JDK1.3, DOM ソフト Crimson 1.1 で測定

試料：ﾚｺｰﾄﾞ内１階層の XML 文書，50 要素(2KB)/ﾚｺｰﾄﾞ