

分散するXML文書間の関係を効率的に保持する機構*

2J-05

辻山昭彦[†] 斎藤信男[†]
 慶應義塾大学政策・メディア研究科[‡]

1. 導入

現在、インターネットの普及と利用の増加が著しい。適用分野・規模も拡大している。併行してそのインフラとなる技術の標準に対する需要もますます強くなっている。多くの標準技術の基盤としてXMLとその関連技術の利用が拡大している。

XMLからXMLへの型変換については木構造の変換関係を簡単に記述できるXSLTが早くから提案されており、利用が進んでいる。

このXSLTによってXML間に確立される変換関係を変換元XMLの更新に対して維持する機構とそれによって得られる利点について以下に述べる。

2. 処理方式

XSLT文書はXML文書間の変換関係を記述している。以下では、変換されるXML文書を文書A、変換されて記述されたXML文書を文書Bとする。文書A、文書Bが別の計算機ノードに存在しネットワークすることを特に想定する。

文書AとXSLTによって文書Bが記述された後、文書Aに何らかの更新があるとき、文書Bの実態を再度文書Aから変換を行って記述しなおすことが現在一般的には行われる。この方式は更新分量に対して文書が大きいとき無駄な処理が多い。

これに対して文書Aの更新から文書Bの更新を算出する手法を提案する。

文書A・B・XSLT文書・文書Aの更新部分から

- ・ 適用されるXSLTインストラクションインスタンスの変更
- ・ インスタンスの出力の変更

の計算によって、文書Bに対する更新を算出する。

この更新を文書Bの実体を処理する計算機ノードに送る。

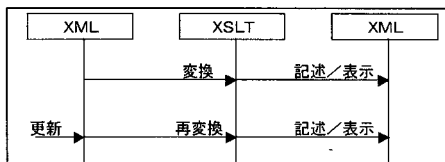


図1：現行のXSLT再処理

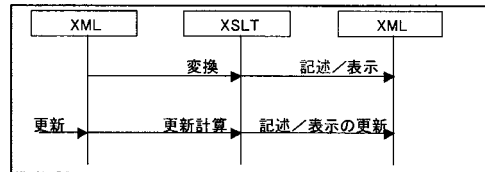


図2：本研究での手法

3. 試行システム

XMLの更新差分だけを記述する方法は提案されているが標準的となっているものは現時点では存在しない。しかしXMLを標準的なモデルに基づいて操作するAPIはDocument Object Modelとして標準化されている。本研究の時点においてはこのAPIによるXMLの更新操作を送信することでプロトタイプを作成を行った。DOM操作を送信するには分散システム、例えばCORBA等によることも可能であるが、今回はJavaScriptで対応する操作を行う記述をテキストで出力・送信する手法をとった。

このようなプロトタイプによって本手法の処理を行ったときの単純な例を以下に示す。

```
<!DOCTYPE doc SYSTEM "doc.dtd">
<section>
  <title>Section Title</title>
  <para>This is a test.</para>
  <para>This is second paragraph.</para>
</section>
```

図3：変換元文書とその更新部分

```
<xsl:transform version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns="http://www.w3.org/TR/xhtml1/strict">
  <xsl:strip-space elements="doc chapter section"/>
  <xsl:output method="xml" indent="yes"
    encoding="iso-8859-1"/>
  <xsl:template match="section">
    <html>
    <head></head>
    <body>
    <h3>
      <xsl:apply-templates/>
    </h3>
    </body>
    </html>
  </xsl:template>
  <xsl:template match="para">
    <p>
      <xsl:apply-templates/>
    </p>
  </xsl:template>
</xsl:transform>
```

図4：XSLT文書

* Mechanism that sustain relations between distributed XML Documents

[†] TSUJIYAMA Akihiko, SAITO Nobuo

[‡] Keio University Graduate School of Media and Governance

```
<?xml version="1.0" encoding="iso-8859-1"?>
<html xmlns="http://www.w3.org/TR/xhtml1/strict">
<head></head>
<body>
<h3>Section Title</h3>
<p>This is a test.</p>
<p>This is second paragraph.</p>
</body>
</html>
```

図5：変換先文書とその更新部分

```
<script>!--
var parent = document.childNodes[1].childNodes[1];
var node_1 = document.createElement("p");
var node_1_1 = document.createTextNode("This is second paragraph.");
node_1.appendChild(node_1_1);
parent.appendChild(node_1);
--></script>
```

図6：JavaScript, DOMでの更新操作

図3のXML文書中、下線部分が追加更新されるとする。この文書が図4のXSLT文書によって変換されている場合、図5のXML文書内、下線部分の追加更新を行う。この操作はDOMとJavaScriptテキストで表すと図6のようになる。

4. 本手法によって得られる利点と定量評価

XML文書実体内の全てのノードを記述しなおさなくてすむ点が本手法の第一の利点である。これは例えば、2つのXML間の関係以外のインタラクションを妨げない利点や、XMLのノードをDBに保持している場合に更新に関係ない無駄な書き込みをしないですむといった利点である。

また同様に、ネットワークする際の通信量が減らせるという利点が存在する。

以下にこの手法を用いて、インターネット掲示板を作成した例を示す。

掲示板の書き込みデータをXMLで保持、その表示をXML(XHTML)で行い、前者の更新から後者の更新を算出してその関係の維持を行う。これを現在一般的な再計算・再ロードを行う方法と比較する。

図7は実在の掲示板サイトのデータ(<http://www.2ch.net/>)をXML化、表示用のXSLTを作成してその通信量を評価したものの一部である。本手法を適用しない場合/する場合とそれぞれの転送量の累積の推移を示したものである。

ある時間中にデータの更新に表示の更新を迫らせる場合、現在一般的な再計算による全体の出力では更新されていない部分の再ロードを含むため時間内の通信量の総計は時間の長さに対して二次関数的に増加する。一方、本手法によれば、その時間に対して更新だけの通信量の総計は一次正比例の増加である。

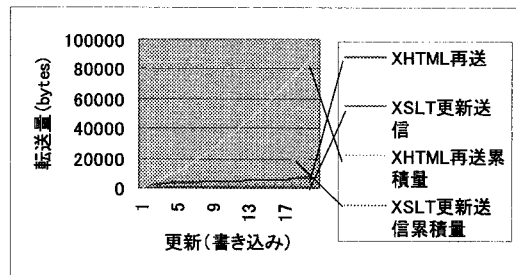


図7：通信負荷の評価抜粋

また本手法は、ユーザ・エージェントの種類を増やしても同じだけXML間の変換関係を増やすだけですむというXSLTを使用する際の利点を維持したまま、同時に、更新の共有を可能にする。ユーザによる記述の必要があるのは通常のXSLTによる文書間変換関係の記述だけで、変換先文書の更新は自動化される。これにより更新を伴う共有文書をインターネット上に簡単に記述可能にする。

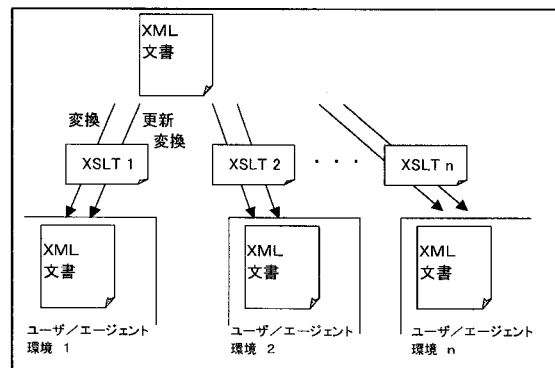


図7：更新を伴う文書の共有

5. この手法の制限と今後の課題

本手法においては文書間関係の記述にXSLTの使用を強いる。XSLTで記述できるような関係にのみ適用可能である。

また、プロトタイプではXML処理系にDOM-APIの存在を仮定していた。その通信トランスポートもスクリプトをストリーム送信するだけの単純なものとした。しかし、更新の記述、配信トランスポートについては更に有用かつ一般化可能なものの実現について考慮する余地があるかもしれない。

また、本手法で使用できるXML間関係は一対一かつ一方のみである。これがどのように拡張可能であるかについての議論は今後の課題である。