2E-6

携帯電話での更新を考慮したXML文書符号化方式

村松茂樹　小林亜令　太田慎司　西山智

株式会社KDDI研究所

1. はじめに

XMLは拡張性・可読性の長所を持つ反面，テキストデータのため冗長性が高い．そのため，特に計算機リソースや伝送路に制約のある携帯電話環境を中心にXML文書を圧縮して伝送符号量を低減する試みが行われている．XML文書の圧縮には，単純にgzip[1]等の汎用のツールを利用する方法，XML文書の特性を利用して圧縮効率を高める方法[2]，XML文書を解析しイベント列として符号化する方法[3][4][5]等が存在する．これらの技術を利用することで，文書取得時の伝送符号量の低減が可能となっている．一方で，データ取得後に行われた操作内容を元のデータに反映することを考えると，符号化に必要な計算機資源が大きいためこれらの符号化技術を利用して伝送符号量を低減することは難しい．また，取得した符号化データを更新するためのAPIを提供する試みもある[6][7]が，伝送路を介して元のデータに反映するためのものではない．本稿では，携帯電話上でXML文書データを更新し，その結果を元のデータに反映することを想定した符号化方式について述べる．

2. 提案方式

XML文書の更新を考えた場合，更新対象となる要素等の位置を指定できることが必要であり，更新は指定された位置において追加/削除等の操作を実施することで実現することができる．このことから，移動による現在位置の変更と現在位置での追加/削除等の操作からなる組み合わせで，XML文書を符号化することとした．操作を順次符号化していくため，符号化されたデータはログ構造化ファイルシステム[8]のような追記型のデータとなる．

符号化には以下の操作を利用した．

新規文書の作成
空の文書を生成する．符号化文書の先頭はこの操作となる．

現在位置の指定
指定された位置に現在位置を移動する．位置の指定には，編集対象の符号化文書を復号する際の要素/属性/値等の出現順に基づいて生成するIDを利用することとした．

要素の追加
要素を追加し，追加した要素の位置に現在位置を移動する．XML文書は，深さ優先で木構造をたどるようにその内容が記述されており，ある要素の次に記述されている要素は，その要素の子要素，弟にあたる要素，何代か親にあたる要素の弟要素の可能性がある．符号化効率を考慮し，要素の追加に関する操作としては，現在位置の要素の子の位置に追加，弟の位置に追加，親要素の弟の位置に追加，n代親要素の弟の位置に追加を別々に用意した．

属性の追加
現在位置の要素に属性を追加し，追加した属性の位置に現在位置を移動する．

要素値の追加
空の要素値を追加し，追加した要素値の位置に現在位置を移動する．要素の追加と同様に，現在位置の要素の子の位置に追加，弟の位置に追加，親要素の弟の位置に追加，n代親要素の弟の位置に追加を別々に用意した．

値の追加
現在位置の要素値，属性に値を追加する．

値の置き換え
現在位置の要素値，属性値を置き換える．

削除
現在位置の要素，要素値，属性を削除する．削除後は，現在位置が未定義の状態になるので，続けて操作を行う場合は一度現在位置を指定する．

符号化時は，要素や属性，また操作を示すのに1Byteを用いることとした．したがって，符号化側と復号側は操作に関する共通の符号化テーブルを保持する必要がある．また，要素名や属性名が必要であれば，同様にテーブルを保持する必要がある．

例えば，<elem attr=“abcd”>XYZ</elem>か

Edit-friendly XML Encoding Scheme for Cellular Phones
Shigeki Muramatsu, Arei Kobayashi, Shinji Ota and Satoshi Nishiyama
KDDI R&D Laboratories Inc.

らなる文書片を符号化し，その後<elem attr="efgh">XYZ</elem>に更新した場合の符号化データの構造は図1のようになる．

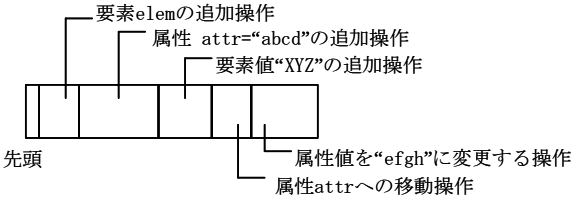

図1: 符号化データの構造

3. 評価

提案方式について以下の評価を行った．

3.1 更新処理

更新処理のパフォーマンスを評価するために，更新対象の文書を格納するサーバと更新操作を発行するクライアントからなる環境を構築した．サーバはCentOS4.2上，クライアントはW32TのBREW3.1環境上のソフトウェアとしてそれぞれ実装した．クライアント内のデータの保持に関しては，例えば[7]に提案があるが，ここでは階層構造を持つリスト構造を用いた．評価環境での処理のフローを図2に示す．

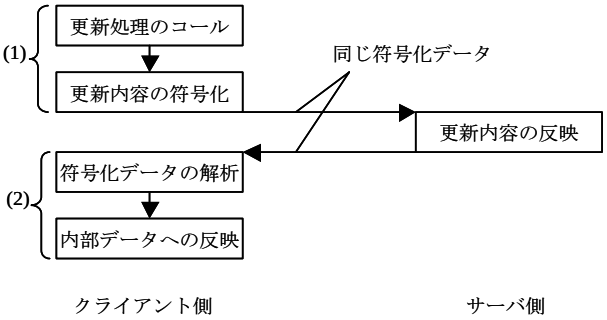

図2: 評価環境での処理のフロー

各操作個別の処理時間を測定するために，5,608Byte(要素数 215)の XML データを作成し，以下の操作について要する時間を測定した．

1. 子要素の追加
2. 子の位置への要素値の追加
3. 属性の追加および属性値の設定
4. 属性値を文字列'setvalue_char'に変更
5. 要素の削除

測定したのは，(1)更新処理のコールから更新内容の符号化までの時間，(2)符号化データの解析から内部データへの反映までの時間である．結果を表 1 に示す．時間は処理を 100 回繰り返した平均である．

(1)については，1操作あたり0.01ms程度で完了している．また，(2)については，遅いもので0.7ms程度要しており，内部データ構造について検討を要する．

表1: 更新処理時間

処理	時間[ms]	
	(1)	(2)
1.	0.002	0.22
2.	0.016	0.722
3.	0.038	0.462
4.	0.008	0.672
5.	測定不能	0.106

3.2 符号化効率

提案方式を用いて符号化を行った場合の符号化効率を評価するため SAX イベントを入力とするエンコーダを実装した．圧縮率(符号化後のデータサイズ/符号化前のデータサイズ)を表 2 に示す．評価コンテンツは XMark[9]を利用した．また，SAXパーサはQt4.0.1を利用した．

表2: 圧縮率

データサイズ	提案方式		XEUS	xmlppm
	単体	gzip併用		
10KB	0.87	0.45	0.46	0.36
100KB	0.89	0.39	0.39	0.28

提案方式は更新に適するよう符号化しているが，符号化後のサイズは悪化していない．また，例えばgzipを適用することで(gzipの列)他の符号化方式と比較し，同等の圧縮率となる．

4. おわりに

本稿では，携帯電話上で XML 文書データを更新し，その結果を元のデータに反映することを想定した符号化方式について報告し，評価によりその有効性を示した．

参考文献

[1]http://www.gzip.org/
[2]H.Liefke et al., Xmill: an efficient compressor for XML data, Proc. 2000 ACM SIGMOD, 2000.
[3]J.Cheney, Compressing XML with Multiplexed Hierarchical PPM Models, Proc. DCC 2001, 2001.
[4]小林他，XML 文書符号化方式「XEUS」，信学技法 DE2001-9, 2001.
[5]小林他，携帯電話のためのXML文書符号化方式"XEUS"の高圧縮化に関する一考察，第124回DPS研究会，2005.
[6]M.Girardot et al., Millau: an encoding format for efficient representation and exchange of XML over the Web, Proc. WWW9, 2000.
[7]小林他，携帯電話を想定したXML文書の高速DOM操作方式，第13回DPSワークショップ，2005.
[8]M.Rosnblum et al., The Design and Implementation of a Log-Structured File System, ACM Trans. Computer Systems, Vol.10, No.1, 1992.
[9]http://monetdb.cwi.nl/xml/index.html