

関係データベースに基づく XPath2.0 処理器の構築 — 問合せ結果再構築について —

佐用 健[†] 田中勇也[†] 天笠俊之[†] 田中二郎[†]

[†]筑波大学 大学院システム情報工学研究科 コンピュータサイエンス専攻

1. はじめに

XML (Extensible Markup Language) は柔軟なデータ構造を文字列で簡単に表現できることから、広く利用されるようになった。今後、XML 文書が増加することが予想されるため、それを管理するための XML データベースが重要である。そのため、XML 文書のデータベースへの格納、格納された XML 文書の処理アルゴリズムの研究が盛んに行われている [1]。

XML データベースの研究において、実験のためのシステム開発に要するコストは無視できないほど大きい。しかしながら、既存の XML データベース処理系は規模が大きく、利用しづらい。一方、独自の実装は実装に時間がかかる上、コードの品質を保証しづらいという問題がある。

そこで我々は、小規模な XPath2.0 [2] 処理器を開発した。本稿では開発した XPath2.0 処理器の概要および著者の開発部分である問合せ結果再構築の部分について述べる。

2. XPathProcceingPlatform について

2.1 システムの概要

我々が開発した XPath2.0 処理器である XPathProcceingPlatform の概要を図 1 に示す。

利用者は、CUI (コマンドラインインタフェース) を用いて、登録したい XML データをシステムに格納したり、XPath2.0 による問合せを行うことができる。

一方、システム内部では、データの格納のために関係データベースを用いる。このため以下の機能を実装している。

- ・ 利用者から与えられた XML データを変換し、関係データベースに格納する機能
- ・ 入力された XPath 式に対応する SQL 問合せに変換し、データベースに問合せを行う機能
- ・ SQL 問合せの結果を XML データに再構築して利用者に返却する機能

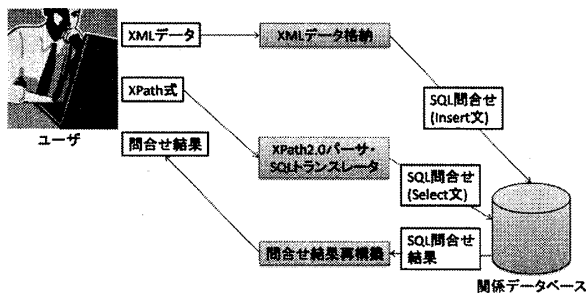


図 1 システム概要図

2.2 サポートする XPath2.0 サブセット

本システムは実用性と実装の容易性を考慮した結果 XPath2.0 のサブセットを実現することとした。その定義を以下に述べる。

- ・ 軸として child, descendant, namespace, attribute を解釈できること
- ・ 述語において論理演算子 (and, or) を利用できること
- ・ 述語において算術演算子 (<, <=, =, !=, =>, >) を利用できること
- ・ 述語において関数 fn:compare, fn:contains, fn:not を利用できること

サブセットは XPath2.0 の基礎であるパス式と述語が解釈できることに重点を置いている。

2.3 XML データの関係表への格納

XML データを関係表へどのように格納するかを説明する。

2.3.1 関係スキーマ

XML データを関係データベースに格納する際には、木構造から平坦な関係表への構造変換を行う必要がある。本システムではその変換方法にモデル写像アプローチを用いた。モデル写像アプローチは XML データモデルに基づいて関係スキーマを設計する方法である。

2.3.2 ノードラベリング

ノードラベリングとは XML データを関係表に格納する際、XML ノードにラベルを付与することである。このラベルを用いて XML ノード間の親子関係を関係表の中でも保持している。

The development of XPath2.0 Processing Platform based on a relational database management system.

[†] Takeshi Sayo, [†] Yuuya Tanaka, [†] Toshiyuki Amagasa, [†] Jiro Tanaka, Department of Computer Science, Graduate School of Systems and Information Engineering, University of Tsukuba

図 2 の XML データが関係表に格納された際の具体例を表 1 に示す。格納の際にノードラベリング方法として DeweyOrder を用いた例である。

```
<book>
    <word>XPath2.0</word>
</book>
```

図 2 サンプル XML データ

表 1 XML データが格納された際の関係表

name	value	type	label	path
book	null	1e	1	/book
null	(改行,空白)	4t	1.1	/book
word	null	1e	1.1	/book/ word
null	XPath2.0	4t	1.1.1	/book/ word
null	(改行)	4t	1.1.1	/book

3. 問合せ結果 XML の再構築について

3.1 概要

著者の開発担当部分である「問合せ結果再構築」の概要を説明する。

この部分は「SQL トランスレータ」部分より SQL 問合せ結果を受け取る。この問合せ結果は特定要素 (XPath 式により選択された部分のルート要素に対応するレコード) のみを含む。開発部分は、受け取った問合せ結果に対応する部分 XML データを再構築する機能である。このため、関係データベースより部分 XML データに含まれるレコードを取得し、XML データに変換し、ユーザに提供するという処理を行う。

3.2 特定要素を頂点とする部分木の取得方法

特定要素を頂点とする部分木を取得するには特定要素の label カラムの値を用いて条件問合せを行う。表 1 の要素名 book が特定要素である場合を例に説明する。表 1 は DeweyOrder でノードラベリングされている。DeweyOrder ではノードの先祖子孫および親子関係は label の値が部分一致するかどうかで判定できる。よって子要素以下を取得する SQL 問合せは

```
select * from テーブル名 where label like '1%' order by label, type;
```

となる。この SQL 問合せを行うことで子要素以下を取得している。なお、XML ノード間の親子関係の保った順序になるよう、label カラム type カラムを用いてソートしている。

3.3 関係表の XML データへの変換方法

特定要素を頂点とする部分木のレコードを XML データに変換する方法を説明する。簡単に説明するため、関係表に要素とテキストのみが格納されている処理の流れを説明する。

特定要素とその子要素以下において処理が異なる。特定要素を出力する処理についてまず説明する。

- ① 関係表の先頭行を読み込み、name カラムの値を用いて開始タグ形成し、出力する
- ② path カラムより階層を求める
- ③ name カラムの値を用いて終了タグ形成し、一時的に保存する
- ④ 関係表の次の行を読み込む
次いで部分木を出力する処理方法を説明する。

- ① path カラムより階層を求める。求めた階層と前回読み込んだ階層を比べ、終了タグを出力するかどうか判断する
- ② type カラムの値が element であれば開始タグ形成し、出力する、処理③へ。値が text の場合はテキストを出力する、処理④へ
- ③ name カラムの値を用いて終了タグを形成し、一時的に保存する
- ④ 関係表の次の行があれば次の行を読み込み処理①へ。次の行がなければ保存している終了タグを全て出力し処理を終える

開始タグと終了タグの形成は SAX(The Simple API for XML) フィルターである XMLWriter クラスを用いた。階層は path カラムの '/' の個数を計算した値である。

7. まとめ

本稿では関係データベースに基づく XPath2.0 処理器の開発について紹介した。実用性を考慮した XPath2.0 サブセットを実現しており、今後 XML データベースの研究に役立つことを期待する。

謝辞

著者ととともに本システムの開発に携わったメンバーである上田保祐氏、羽鳥貴之氏に深く感謝いたします。

参考文献

- [1] 清水 敏之, 鬼塚 真, 江田 毅晴, 吉川 正俊, “XML データの管理とストリーム処理に関する技術”, 電子情報通信学会論文誌
- [2] W3C: ” XML Path Language (XPath) 2.0”, <http://www.w3.org/TR/xpath20/>, W3C Recommendation 23 January 2007