

RDB上のXMLビューに対するDOM操作実現機構

小島 章 森嶋 厚行
芝浦工業大学工学部情報工学科

1. はじめに

XMLはインターネットにおけるデータ交換フォーマットの標準となりつつある。しかし現実のシステムでは、データはリレーショナルデータベース(RDB)に格納されているのが一般的である。これまで、RDBに格納されたデータ上にXMLビューを構築する研究は数多く行なわれてきたが、それらのほとんどは、XPathやXQuery、拡張SQL等を通じたアクセスを提供するものであった。本稿では、RDB上のXMLビューに対してDOM API準拠の操作を提供する機構の設計について説明する。DOM APIは、XMLに対する手続き的操作を可能とするAPIとして広く使われている。本システムでは、XMLビューに対するDOM APIに準じた操作(以下DOM操作)をSQLに変換し実行する。アプリケーション側ではその変換を意識する必要はないため、XMLデータを想定して作成されたアプリケーションを容易にRDB環境に適用できる。

2. システム概要

本システムの全体像を図1に示す。システムはまず、XMLビュー生成ルールを入力として受け取る。本システムはビュー生成ルールとしてViewtree[1]を用いる。アプリケーションは、通常のDOM準拠APIを用いてXMLビューの操作を行なう。DOM-SQL変換モジュールは、XMLビュー生成ルールとDOM操作からSQL問合せを生成し、RDBに投入する。また、問合せ結果からDOM操作の結果を作成し、アプリケーションに返す。

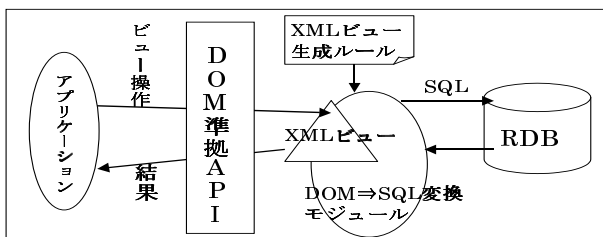


図1 システム全体像

3. Viewtree

Viewtreeは、XMLビューの構造と、各XML要素を計算するためのSQL問合せを組み合わせたものである。例として、図2のリレーションから図3のXMLビューを生成することを考える。この変換を表すViewtreeは図4になる。直感的には、Viewtreeの構造はXMLビューの構造を

要約したものである。各ノードは要素名と、その要素を計算するためのSQL文から構成される。また、SQLの実行結果リレーションのキー情報、およびXMLビュー中に入力される値を格納している属性の情報を持つ。例えば、図4のViewtreeでは、ルートノードN0の子ノードN1は、要素名として「個人情報」を持ち、その要素を計算するためのSQLとして「SELECT * FROM 学生データ」が格納されている。N0のSQL問合せは空になっているが、これはXMLビューのルートノードとしてただ一つのノードを出力することを示す。詳細な定義は[1]にある。

学生データ		電話番号	
学籍番号	名前	番号	学籍番号
1	伊藤	111-111	1
2	小島	222-222	2
3	油井	333-333	3

図2 リレーション例

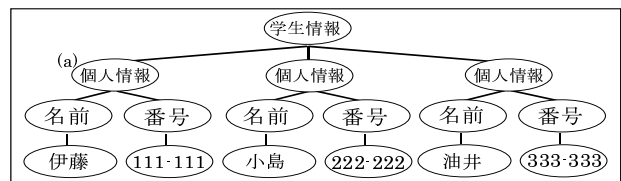


図3 RDB上のXMLビュー

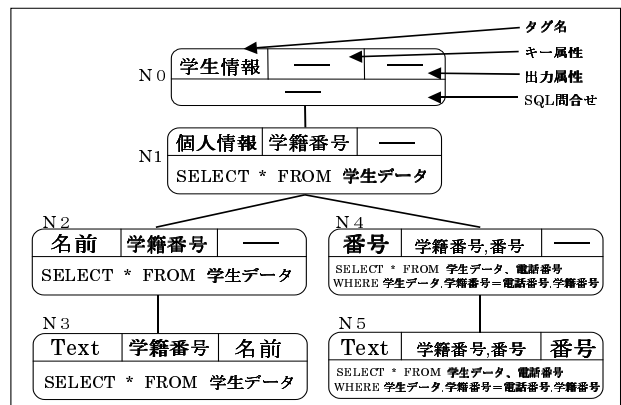


図4 Viewtreeの例

4. Viewtreeを用いたDOM準拠APIの実装

本システムでは、DOMオブジェクトへの参照として、組 (n, K) を用いる。ここで、 n はViewtreeノードのIDであり、 $K = [a_1 : x_1, \dots, a_n : x_n]$ はキー値を表す(a_i は n で指定されているキー属性、 x_i は各キー属性の値である。)例えば、図3のXMLビュー上のオブジェクト(a)の参照は、 $(N1, [学籍番号:1])$ である。本システムで実装する関数の仕事は、このようなDOMオブジェクト参照を計算することである。ここでは、子ノードの参照を返す`getFirstChild()`

A DOM-compliant Manipulation Framework for XML views over RDBs
Akira Kozima, Atsuyuki Morishima
Dept. of Info. Sci. and Eng., Shibaura Inst. of Tech.

関数の実装について説明する。他の関数についても類似した仕組みで実装することができる。図5に `getFirstChild()` のアルゴリズムを示す。

基本的な流れは次のとおりである:(1)Viewtree における子ノード $n2$ を取得。(2) $n2$ の要素を計算する SQL 問合せを取得。(3)SQL 問合せ結果のうち, r のキー属性と値が一致するものだけを選択。(4) その結果の先頭のタプルから新しいキー属性値を取得。

1. Let r be $(n1, [a_1:x_1, \dots, a_n:x_n])$
2. $n2 = n1.getFirstChild()$
3. $q = n2.query()$
4. $q.where().append("a_1=x_1 \&\& \dots \&\& a_n=x_n")$
5. $ResultSet\ rs = q.executeQuery()$
6. $Tuple\ t = rs.next()$
7. $(a_1, \dots, a_m) = n2.KeyAttributes()$
8. return $(n2, [a_1:t.get(a_1), \dots, a_m:t.get(a_m)])$

図5 `r.getFirstChild()` を計算するアルゴリズム

5. 問合せ最適化

4章の手法では, DOM 関数が呼び出されるたびに SQL 問合せの作成・実行を行なうため, DOM 関数を多数呼び出すようなアプリケーションにおいてパフォーマンスに問題がある。本システムでは, 可能な場合には複数の SQL 問合せ呼出しを一つにまとめることにより, 処理の効率化を計る。本最適化の流れを図6に示す。本手法のポイントは, アプリケーション実行時の DOM 操作の実行履歴を基にアプリケーションの実行パターンを作成し, 2回目以降の問合せ実行時には, そのパターンを利用して, SQL の生成および実行を行なうことである。実行パターンは基本的に状態遷移図であり, データフローに関する情報が付加されている。例えば, あるアプリケーションプログラムが図4の Viewtree で定義される XML ビューを深さ優先走査する場合を想定する。その場合の実行パターンは図7のようになる。この「深さ優先走査」パターンを発見すると, システムはこれらの結果を一度に生成する SQL 問合せを生成する。SQL 問合せは, sorted outer-union プラン [2] を用いる。この場合は次のようになる。

```
SELECT 学籍番号, 名前, NULL as 番号, NULL as 学籍番号
FROM 学生データ
UNION
SELECT *
FROM 学生データ, 電話番号
WHERE 学生データ.学籍番号 = 電話番号.学籍番号
ORDER BY 学生データ.学籍番号
```

この問合せを実行し, 先頭のタプルから順に読み出していくことにより, 上記のパターンと矛盾しないデータの読み込みができる。図1の DOM-SQL 変換モジュールは, このようなパターンに基づく SQL 問合せを実行した後は DOM 操作関数の実行毎に SQL 問合せを作成することはせず, 順

次タブルの内容から DOM 操作関数の結果を計算し, アプリケーションに返す。

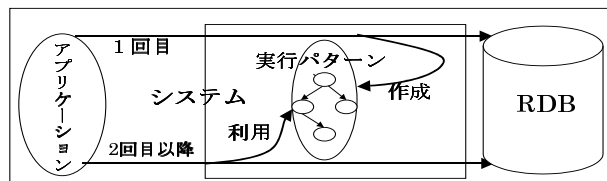


図6 本システムによる最適化

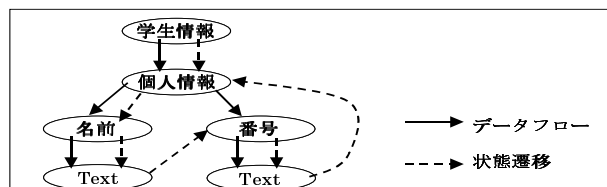


図7 実行パターンの例

実行パターンは, DOM 操作関数の実行時における (1) メソッド適用対象の DOM オブジェクトの参照。および, (2) 結果の DOM オブジェクトの参照を基に計算される。

6. 関連研究

ROLEX[3] は, 本システムと同様に RDB 上の XML ビューに対して DOM 操作を実現するシステムである。ROLEX は, 特定のメインメモリデータベースを想定したシステムであり, 任意の DBMS と接続可能なミドルウェア環境を想定した本システムとは異なる。ROLEX の最適化機構は, あらかじめ Profile として与えられた条件付確率に基づいている。我々のアプローチは, アプリケーション実行時の情報をより有効活用しようとしている。

7. おわりに

本稿では, RDB 上の XML ビューに対して DOM API に準じた操作を実現するシステムについて説明した。本システムでは, 動的に計算する実行パターンを用いて SQL 問合せを作成することにより処理の効率化を計る。今後の課題としては, より洗練された問合せパターンの発見・SQL 問合せ生成アルゴリズムの開発がある。

参考文献

- 1) M. Fernandez, Y. Kadiyska, A. Morishima, D. Suciu, W. Tan. SilkRoute : A Framework for Publishing Relational Data in XML. ACM TODS 27(4):438-493.
- 2) Jayavel Shanmugasundaram, Jerry Kiernan, Eugene J. Shekita, Catalina Fan, John Funderburk: Querying XML Views of Relational Data. VLDB 2001: 261-270.
- 3) Philip Bohannon, S. Ganguly, Henry F. Korth, P. P. S. Narayan, Pradeep Shenoy: Optimizing View Queries in ROLEX to Support Navigable Result Trees. VLDB 2002.