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.

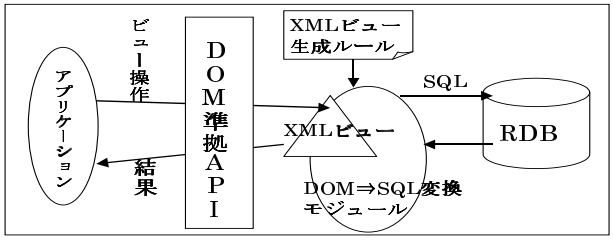

図 1 システム全体像

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 ビュー中に出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

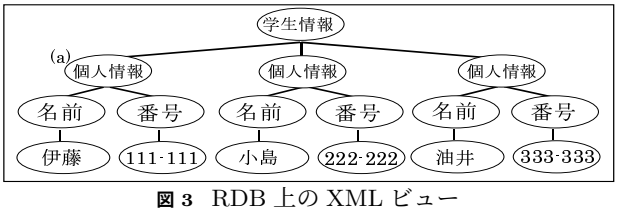

図 3 RDB 上の XML ビュー

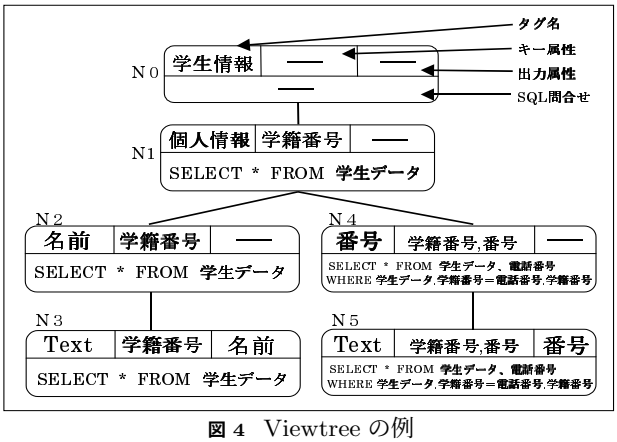

図 4 Viewtree の例

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，$K = [a_1 : x_1, ..., a_n : x_n]$ 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,...,a_n:x_n])
2. $n2$=$n1$.getFirstChild()
3. q=$n2$.query()
4. q.where().append("a_1=x_1 &&...&& a_n=x_n")
5. ResultSet rs=q.executeQuery()
6. Tuple t=rs.next()
7. (a_1,...,a_m)=$n2$.KeyAttrbutes()
8. return ($n2$,[a_1:t.get(a_1),...,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，順次タプルの内容から DOM 操作関数の結果を計算し，アプリケーションに返す．

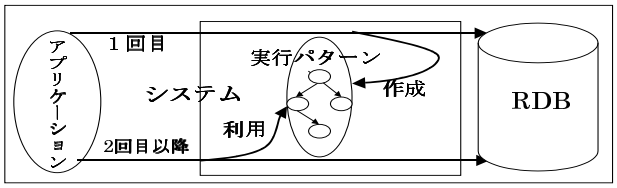

図 6 本システムによる最適化

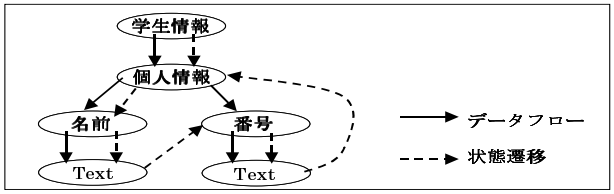

図 7 実行パターンの例

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.