

異種データベースの仮想化技術 -クエリ変換方式-

菖蒲 佳右[†] 渡辺 裕太[†] 和田 雄次^{††} 澤本 潤^{†††} 加藤 貴司^{†††}東京電機大学大学院 情報環境学研究所[†]東京電機大学 情報環境学部^{††}岩手県立大学 ソフトウェア情報学部^{†††}

1. はじめに

ユビキタスコンピューティング環境において利用されるデータベース (DB) には, RDB, XMLDB など, 用途に応じて多種多様な DB (マルチデータベース) が存在する. それにより, 収集された情報に適した形で情報を整理することができるようになってきている. しかし, それらを複合的に利用しようとする際, DB の差異がユーザにとって障害となることが考えられる. 今回は, そういった差異を埋めることを目的とした仮想DBの開発を行った.

2. 既存のDBの問題点

RDBやXMLDBを同時に利用する際, RDBにはSQL, XMLDBにはXQueryといった各DBに用いる問い合わせ言語や, データモデルなどの違いがある. 例えば, 図1のようなデータがRDBに, 図2のようなデータがXMLDBに格納されている場合を考える.

テーブル名: 社員情報			
社員ID	氏名	生年月日	性別
1	山田太郎	1970-03-04	男
2	金子信二	1975-04-02	男
3	太田麻子	1974-01-09	女
4	安田伸佳	1980-06-25	女
5	森島勲	1948-10-16	男
6	大西功輝	1959-09-02	男

図1. 社員情報テーブル

```

組織構造.xml
<組織構造>
  <開発部署>
    <第一開発部>
      <部長>3</部長>
      <課長>2</課長>
      <係長>7</係長>
    </第一開発部>
    <第二開発部>
      <部長>5</部長>
    </第二開発部>
  </開発部署>
</組織構造>

```

図2. XMLDB データ

このデータを基に第一開発部の部長を知ろうとする場合, 次のようなクエリを送信する.

○第一開発部の部長の名前を知りたい

●RDB から社員 ID と氏名の一覧を得る

```
SELECT 社員ID, 氏名 FROM 社員情報
```

●XMLDB から対象の社員 ID を得る

```
let $a := document("組織構造.xml")/組織構造/開発部署/第一開発部/部長
return $a
```

このクエリの実行結果は図3, 図4のようになる.

●RDB

社員ID	氏名
1	山田太郎
2	金子信二
3	太田麻子
4	安田伸佳
5	森島勲
6	大西功輝

●XMLDB

```
<部長>5</部長>
```

図4. 第一開発部部長の社員 ID

図3. 社員 ID, 氏名一覧

以上のような結果がそれぞれのクエリに対して出力されるが, RDB と XMLDB はそのお互いの結果を用いて最終的にユーザが知りたい結果を出力するといったことを行うことができず, ユーザ自身がその結果を比較しなければならない. また, 問い合わせ言語がそれぞれ違うため, 使用するDBに応じた複数のクエリ (この場合はSQLとXQuery) について理解していなければならない. さらに, どのデータがどのDBに格納されているかということも把握しておく必要がある. これらはユーザにとっても負担となり, 作業により多くの時間をかけることにもなる.

3. 仮想DB

3.1 概要

今回開発した仮想DBは, 2章で述べた問題を解決するために, 複数のDBがユーザからはあたかも一つのDBMSで管理されているかのように見えるようにする. 例えばRDB, XMLDBなどの異種データモデルを扱う際, ユーザに仮想DBのDBMS (仮想DBMS) を介してアクセスさせることで仮想DBのみを意識すれば済むことになる.

DBの仮想化は, 仮想DBMSが扱う実際のDB (実DB)

Virtualization technique for heterogeneous database - query conversion method -
[†] Keisuke Shobu ^{††} Yuta Watanabe ^{†††} Yuji Wada ^{†††} Jun Sawamoto ^{†††} Takashi Kato
[†] Graduate School of Information Environment, Tokyo Denki University
^{††} School of Information Environment, Tokyo Denki University
^{†††} Faculty of Software and Information Science, Iwate Prefectural University

の構造を統合化したものを XML で表現することで行った。この XML で表現したものを共通スキーマと呼ぶ。図 1, 図 2 のデータを共通スキーマで表した場合, 図 5 のような構造を持つ XML で表現される。ユーザはマルチデータベースの構造を統合化して表現されたこの共通スキーマを基にクエリを発行し, 仮想 DBMS がそのクエリに応じて各実 DB へのアクセスを行う。

3.2 仮想 DB 内におけるデータ構造

仮想 DB 内において, RDB のデータと XMLDB のデータは同じ構造を持つ Java オブジェクトの形に変換される。その Java オブジェクトの構造は図 6 に示す通りである。RDB のデータと XMLDB のデータをこのようなオブジェクトに変換して形を統一することで“仮想 DB のデータ”として扱い, 双方の差異を埋めている。その上でユーザが発行したクエリに応じて処理を行い, 各実 DB を複合的に利用した値をユーザに返却する。

3.3 仮想 DB に用いるクエリ

ユーザが入力するクエリは各実 DB に対するクエリではなく, 仮想 DB に対するクエリとなる。仮想 DB に対してユーザが発行するクエリとして考えられるものとしては, SQL, XQuery, 仮想 DB オリジナルのクエリなどと候補がある。今回は仮想 DB の全体の構造を表す共通スキーマが XML の形で記述され, それが仮想 DB 全体の構造としてユーザに提示されることから, XQuery を仮想 DB に対してユーザが発行するクエリとした。

3.4 仮想 DB に対する問い合わせ

2 章で挙げた例を仮想 DB を用いて行う場合, ユーザは次のようなクエリを発行することになる。

```
○第一開発部の部長の名前を知りたい
for $employee in common-schema()/RDB/sample_rdb/社員情報/row
for $manager in common-schema()/XMLDB/sample_xmldb/組織構造.xml
where $employee/社員ID = $manager/組織構造/開発部署/第一開発部部長
return $employee/氏名
```

このクエリの実行結果は次のようになる。

```
○実行結果
<氏名>霧島勲</氏名>
```

クエリ中にある common-schema() は, 仮想 DB において新規に定義した関数であり, 共通スキーマの root を指し示

す際に使用する。

このように, 仮想 DB を用いることにより, DB の違いを意識することなく, RDB と XMLDB の値を取得, 利用することができるようになった。

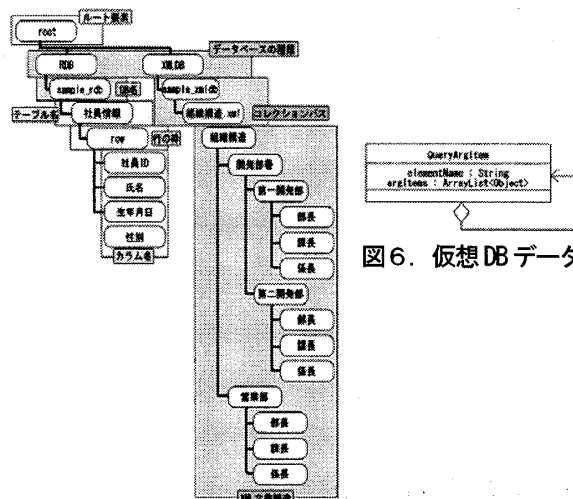


図 5. 共通スキーマ例

図 6. 仮想 DB データ

4. まとめ

通常, 複数の種類の DB からデータを取得する際, その DBMS に合わせたクエリを発行し, その結果を用いてユーザ自身で欲しい情報を導く必要があった。今回開発した仮想 DB を用いることにより, ユーザはそういった違いを意識することなく, あたかも一つの DB を扱うかのように情報を取得することができるようになった。

現在は述部の記述や order by 句, SQL の参照制約などといったものに対応していないので, 今後はそれらに対応させていく必要がある。

謝辞

本研究は, 科学研究費補助金 (基盤研究(C) 課題番号 20500095 「ユビキタスデータベース仮想化技術によるデータ利用の効率化に関する研究」) の支援による。

参考文献

- [1]. W3C, XQuery 1.0 and XPath 2.0 Functions and Operators
- [2]. W3C, Building a Tokenizer for XPath or XQuery
- [3]. W3C, XQuery 1.0: An XML Query Language
- [4]. W3C, XML Path Language (XPath) 2.0
- [5]. 1keydata.com, SQL 入門, <http://sql.1keydata.com/jp/>
- [6]. Jeff Ullman, and Jennifer Widom., "A First Course in Database Systems", Pearson Education Internatiola, 2008. pp.517-554