

分散情報統合システム

6K-4

野崎康行 古舘丈裕 小野山隆
日立ソフトウェアエンジニアリング（株）

1. はじめに

インターネット・イントラネットの発達とWWWの普及により、ネットワークを通して様々な種類のデータを簡単に入手できる環境が整ってきた。しかし、現状では入手したそれぞれのデータの個別的な閲覧が主たる利用法であり、それらのデータを有機的に組み合わせて利用するには、表記の統一や重複の除去などの編集作業の大部分を手作業で行う必要があった。これらを自動化し分散したデータを統合的に利用できるアプリケーションを実現するためには、それぞれのデータのスキーマ情報を保ちつつ、データ間の異種性を解決する必要がある。

我々は利用しやすい単位でデータを抽象化したビューオブジェクトを構築することで統合的なインタフェースを提供するというアプローチを採っている。本稿では、RDBや表計算ソフト等のテーブルデータを対象として、アプリケーション間でのデータの交換利用をサポートするミドルウェアとして試作した分散情報統合システムについて述べる。

2. 統合ビューインタフェース

複数のサーバ上に分散したデータをクライアントから統合的に利用するには、プラットフォームやデータの格納方法などの物理的・形式的な異種性ととともに、意味的な異種性も解決する必要がある[1]。意味的な異種性とは、ここでは内容的に同じ意味のデータにそれぞれ異なる名称がつけられていたり、データの値の表現が異なっていることを指す。しかし意味的な異種性の解決は非常に困難な問題である。我々はまずアプリケーションデータを構成する要素をテ-

ブル、図、テキストといった形式で分類し、それぞれをCORBA[2]のオブジェクトモデルを用いて抽象化することで物理的・形式的な異種性を吸収する。その上でデータの名称や値の変換を行うオペレーションを追加していくことで意味的な異種性の解決を目指す。図1にテーブルデータを対象として試作した分散情報統合システムの構成を示す。

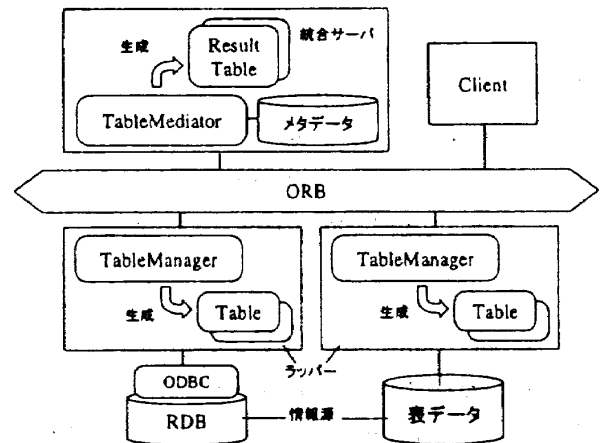


図1 分散情報統合システムの構成

本システムでは情報源に含まれる一つ一つのテーブルに対応したオブジェクトを作成する。このオブジェクトは、実際のテーブルデータに対するビューの役割を果たす。他のアプリケーションはテーブルオブジェクトが提供するインタフェースを通して実際のテーブルにアクセスする。これらのオブジェクトは、IDL[2]で記述するインタフェースのみが統一されており、実装は対象の情報源毎に異なっている。例えばテーブルの各セルの内容を実際に持つような実装や、実際のデータはRDB等に格納されていて要求に応じてそれらを引き出すような実装が考えられる。

3. 分散情報統合システムのアーキテクチャ

本システムは、クライアント、統合サーバ、情報源の三階層アーキテクチャであり、情報源

にはラッピングモジュール（以下、ラッパー[3]）が付加される。ラッパーはデータの形式的な異種性を吸収するモジュールであり、情報源の種類毎に作成する必要がある。

本システムにおけるラッパーはテーブルオブジェクトの生成や管理を司る TableManager と、それが生成する Table オブジェクトから構成される。Table はテーブルのデータにアクセスするための操作を提供し、TableManager はテーブルの作成や削除、Join 等のテーブル間の演算を行う操作を提供する。図 2 に、IDL による TableManager 及び Table のインタフェース定義の一部を示す。

```
interface Table {
    attribute string name;
    attribute long row;
    attribute long column;
    any value(in long row, in long col);
};

typedef sequence<Table> TableList;
interface TableManager {
    boolean removeTable(in string name);
    Table findTable(in string name);
    Table join(in Table r, in Table s);
    TableList search(in Condition query);
};
```

図 2 テーブルのインタフェース定義

情報統合サーバでは、TableMediator が複数の情報源からデータを検索し、得られた結果を合成して Table オブジェクトを生成する。このオブジェクトを参照することでクライアントからは、実際には散在しているデータがあたかも一個所にまとめられているように見える。情報統合サーバは問合せ条件の指定を受け取り、条件を満たしたテーブルを返す操作を提供する。

4. データの異種性の解決手段

多くの場合それぞれの情報源は自律的であり、所有する部署や組織によって独自に管理される。そのためスキーマは独自に設計され、その結果意味的な異種性が生じる。

例えば、図 3 に示す二つの表において、左の表の「定価」と右の表の「単価」は共に価格という概念を表しているが属性名は異なっている。また、値の表現の仕方も異なる。このような異種性を解決する為に、情報統合サーバではドメインを表すオブジェクトを定義する。

製品情報			発注		
形式	定価	在庫	型番	数量	単価
AX-18	38000	12	AX-18	1	38,000円
AX-20	25000	18	AX-20	2	25,000円
BX-22	18000	4	BX-22	4	18,000円
...			

図 3 異種性を含むテーブルの例

テーブルの属性はドメインと表現形式から成る。ドメインは型と制約から成り、制約はある型の値がそのドメインに含まれるかどうか判定する関数で表す。また表現形式は、あるドメインに含まれる値の、記述する際のルールを定める関数で表す。価格という概念を表すドメインを Price とすると、Price は型 Integer とブール関数 $f(x)$ から構成される。ここでは f は x が 0 以上ならば真を返すような関数とする。表現形式としては、IntValue(x)、StrValueYen(x) などがあり、前者は数値として、後者は“円”を末尾に付した文字列として値を表現する。

これらのドメインおよび関数群をオブジェクトとして情報統合サーバで管理し、これらを用いてラッパーを介して取得したテーブルに対して属性名変換やデータ値の変換を行う。

5. まとめ

テーブルデータを抽象化し複数の異種情報源上のテーブルを統合利用するための、CORBA を利用して試作したシステムを紹介した。今後はドメインオブジェクトの充実と、テーブル以外のアプリケーションデータの抽象化を行う。

参考文献

- [1] 上林弥彦, “マルチデータベースの研究開発動向”, 情報処理, 35(2), 1994
- [2] Object Management Group, “The Common Object Request Broker: Architecture and Specification Rev.2.0”, July 1995
- [3] M. T. Roth, P. M. Schwarz, “A Wrapper Architecture for Legacy Data Sources”, IBM Technical Report RJ10077, 1997