

広域分散環境における個人間連携データベース提供環境の提案と試作

奥田 琢馬[†] 川越 恭二^{††}

[†]立命館大学大学院理工学研究科 ^{††}立命館大学理工学部

525-8577 滋賀県草津市野路東 1-1-1
{okuda,kawagoe}@ims.cs.ritsumeikan.ac.jp

概要

広域ネットワーク上に多数の利用価値の高いデータベース群が接続されている環境において、散在する情報源の共有化および有効性のための個人異種データ群の連携利用が重要な課題となっている。個人作成者が固有の目的や編集利用するデータ群では、データ間の結合に伴う異種性の解決は特に困難な問題となる。本稿では、広域データベース環境上において個人管理者や利用者作業の軽減と情報検索の効率化を図るシステム支援方法を提案する。本システムでは個人データベースに焦点をおき、そのマルチデータベース環境において、個人利用者でも相互接続を容易に実現するためのシステム間を統合するインターフェースを設定した。

Information Platform for Database Services using Mutual Relationships between Personal DBs under Global Network Environment

Takuma Okuda[†] Kyoji Kawagoe^{††}

[†]Graduate School of Science and Engineering, Ritsumeikan University ^{††}School of Science and Engineering, Ritsumeikan University

1-1-1, Noji-Higashi, Kusatsu-city, Shiga 520-8577, JAPAN
{okuda, kawagoe}@ims.cs.ritsumeikan.ac.jp

Abstract

In this paper, we propose an information platform for database services using mutual relationships between personal DBs under global network environment. There exist some problems on combining existing personal databases mutually for sharing and utilizing these databases more effectively. One of the important problems is related with the difference in structures and interfaces on personal database environment. In order to solve the problem, we propose some common interfaces for combining personal databases mutually. Especially, the interfaces and platform with the interfaces enable personal users to generate their own personal databases using others existing databases and to publish their generated personal databases later. In the paper, the concept, the structure and the prototype system of the proposed platform are briefly described.

1 はじめに

WWW 技術の普及により、ネットワークを通して様々な情報を即座にしかも大量に入手できるような環境が存在する。特にデータベースを利用したシステムは様々なデータの収集、分析を行う上で重要なものとして位置づけられている。データベース群が広域ネットワークに接続されうようになり、散在する情報源の共有化および有効利用が求められている。このような環境下では、個人データベース(ローカルデータベース)の自律性および拡張性を維持しつつ、それらの連携

利用を可能とする相互運用の実現が重要であると考えられる。

本稿では、広域データベース環境上において個人管理者や利用者作業の軽減と情報検索の効率化を図るシステム支援方法を提案する。本システムでは個人データベースに焦点をおき、そのマルチデータベース環境において、個人利用者でも相互接続を容易に実現するためのシステム間を統合するインターフェースを設定した。このインターフェースおよび構成について説明するとともに、試作したプロトタイプを紹介する。

2 研究背景

ネットワークの普及に伴い、インターネットにおける WWW のように、不特定多数に情報提供を行うデータベースが世界規模で分散して存在している。このインターネットで多数のデータベースが散在する環境では、WWW の様なリンクによるアクセスだけでなく、散在するデータベースに対する統合的な条件検索が可能になればネットワーク利用の可能性は飛躍的に高まると考えられる。

しかし、現在、大多数のデータベースは閉ざされた空間内に存在し、データベース自身は公開されていない。個人あるいは企業が独自のデータベースを作成し、独自の Web 上から利用するという形態で利用されている。このため、一般個人利用者にとっては直接にデータベースにアクセスすることができず、また、未知なデータベースも散在している。また、個人利用者がデータベースを構築しようとしても、HP 作成のように容易に構築する環境は提供されていない。特に、データベース以外に Web サーバなどのソフトウェアやプログラミング技術などの技術を用いてのみ公開することができ、平易に利用できるまで普及しているとは言いがたい。

データベースは、セキュアな情報源を除いて、特定の利用者に限定するのではなく、多くの利用者へ解放することで利用価値を高めることができる。また、特定のウェブサイトあるいはデータベースには、テキストデータ、画像データ、記録など種々のメディアの情報が混在し、情報の細分化し、内容に応じて集約化、再編成を行うことで、それぞれの情報は何倍もの価値を持つようにできる。

例えば、現在利用されているオンラインデータベースには、a)地図、画像、音など多様なデータを連携される電子図書館、b)オンラインショッピングなど電子モール、c)医療関連の情報提供データベース、d)企業および団体などの営利目的とする市場データベース群は、情報源の集合体に過ぎず、複数の情報源を活用することはできない。また、独自の Web ページで作成されている為、統一的なアクセスを行うのは困難である。一方、これらのデータベースを公開データベースとして公開しさらに管理することで、データベースがどこに存在し、また、設定情報なども利用することができる。さらに、個々の公開データベースの集合をグループ化、ジャンル別に細分化し、集約化に再編成することで大きく利用者の目的に応じて情報へ効率よくアクセスを行うことが可能となる。そこで、上記のように、既存の公開データベースを細分化し、集約化、再構成が簡単

に行うことのできる情報基盤環境システムを提案する。

以降、まず 3 章で本システムの構成およびインタフェースに関する方式の基本的方針を述べ、4 章では本方式を実装したマルチデータベースシステムの構成を示す。最後に実現したプロトタイプシステムを作成し、その実用性・有効性を述べる。

3 基本的方針

まず、管理者・利用者の観点から利用問題点をあげ改善するポイント、その解決に向けての基本的な考え方について述べる。図 1 に利用概念図を示す。

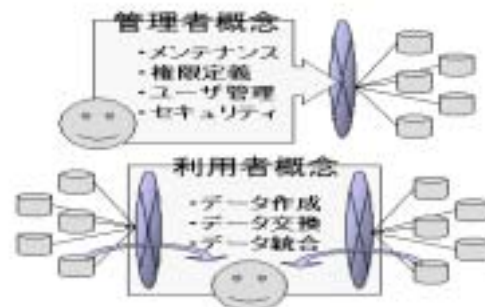


図1 利用概念図

3.1 公開DBの管理者

管理者はDBソフトだけ使用し、データベースの維持管理からデータ操作まで簡単に行いたい。また、公開してもよいDB一覧情報にアクセスし、取得したいDBのアクセス情報を取得する。それぞれのDBにアクセスし取得したい情報だけユーザによって取捨選択し情報を統合した上で既存DBに追加したい。以下に管理者項目を示す。

- 管理者はDBテーブル構造などのデータベース管理を編集可能とする(メンテナンス)
- 管理者は他利用者が作成した(DBアプリケーションプログラム)を利用可能とする
- 利用者は上記を用いてプログラムでき、自らの画面出力を作成可能とする
- 利用者・管理者が公開DBを他のDBと組み合わせてwebブラウザに出力する

3.2 公開DBの利用者

利用者は情報を入手したいが実際にDBがどこに存在するかわからない。そこで公開してもよいDB一覧情報にアクセスし、取得したいDBのアクセス情報を取得する。それぞれのDBにアクセスし取得したい情報だけユーザによって取捨選択し情報を統合した上で入手することができる。以下に利用者項目を示す。

- 利用者は公開DB情報からアクセス情報取得しアクセス

を行う

- 利用者は他利用者が接続している DB と情報共有すると共に個人間接続を可能とする
- 利用者は複数の異種 DB を組み合わせて Web 上に出力を可能とする。

実際の使用者（利用者・管理者）が、単にデータベース内を検索し閲覧するのではなく、たとえばデータの操作あるいは検索したデータを表やグラフに加工しレポートとして仕上げる事や分析のための資料とする事も可能となるような柔軟性を持たせることが重要となる。

そこで、本稿で提案するシステムでは管理者・利用者における次の3点を目的として設定する。

(1) 利用者および管理者作業の軽減支援

公開DBのDB利用者、DB管理者のそれぞれに対してデータベース操作を柔軟に行うためのインターフェイスを設け、作業の軽減を図ることが重要である。

データベース利用や管理を行うことはDB操作に関する経験・知識が必要であり個人にとって大きな負担となる。また、管理者は、Web上からデータベースを構築・運営管理することが難しい。また、利用者は、散在したデータベースから希望する情報源を探すには多大な時間と労力を必要とする。

(2) 既存DB / 異種DBへのアクセス支援

提案する方式では異種機間のデータベース群を統一的に扱う枠組みを提供する。

既存データベース・異種データベース間での機能の異種性と問い合わせ言語間の異種性の問題が考えられる。そこで、次の2種類の異種性を抽象化することで実現する。

- 問い合わせ言語の異種性
- データベースシステム機能の異種性

データベース群において既存・異種データベースが同種の問い合わせ言語から構成されている場合は、双方間で方言のないSQLで問い合わせが可能である。また、データベース集合において既存・異種データベースが異種の機能から構成されている場合は、双方間で機能が統一されていないため、機能の抽象化を行う。多くの場合、データベースシステムのインタフェース(API)はベンダごとに異なるものであり、その異種性を吸収する必要がある。このAPIの異種性を吸収するものに、JDBC^[11]やODBC等が存在する。

(3) 問い合わせ処理におけるデータ統合支援

提案する方法では、全体を仮想的なデータベースとし各サイトはその一部を格納した個人データベースと考

えることにより、広域問合せを実現する。

散在する公開データベースにまたがる情報の検索では種々の不整合や重複、欠落などの問題があり単純な問合せでも処理する事が困難である。これまで、異種データベースを統一的に扱う問題はマルチデータベースで扱われているが、インターネットのように不特定多数のデータベースが散在する環境では、WWWの様なりんくによるアクセスだけでなく、散在するデータベースに対する統合的な条件検索が必要である。これまでに、BrokerやMediator^[5]、Wrapper^{[2][4]}等による実現が図られている。

4 システムアーキテクチャ

提案する本システムでは、3章に述べたような個人データベース連携支援システムの問題点を回避するシステムの実現を目指す。

本稿では、図2に示すシステム構成の個人データベース連携支援システムを提案する。なお、図2の個々の機能については5章で説明する。

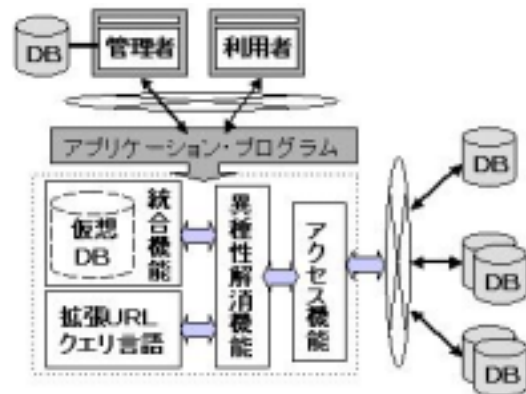


図2 システム構成図

本システムは以下の4つのモジュールから構成される

- アクセス機能方式
- 異種性解消方式
- 拡張（URL）クエリ方式
- データ統合方式

上記の4つの機能を実現することにより、個人データベース連携支援システムは、以下の特徴をもつ。

システム開発期間の短縮

DB設計と独立してAP開発が可能となるため並行して開発が可能となる。

システム運用の容易性

新たな情報源の追加 / 編集 / 削除がWeb上で実施でき指定フォームに従いDB構造の変更にAPが影響を受けない。

システム間の異種性吸収

プラットフォームやDBソフトに依存せずDB設計を行うことが可能である。

5 システム内部構成

5.1 アクセス機能

実際に個々の情報源に対してアクセスを行い、検索要求の送信、検索結果の受信等を行う機能である。

データベース接続に対してアクセスを行うため以下のような処理形式を可能とする。アクセス時にデータベース固有情報を取得しデータベース内容を把握すると共にアクセスを行う。また、DB のデータ構造、データの表現形式、列の取り得る値の範囲等も管理する。これらのアクセスを容易にするためにメタデータ管理機構を構成しデータのやり取りをスムーズに行う。

5.2 異種性解消方式

多種多様なデータソースに散在したデータの多くの問い合わせに答えることができる環境が必要である。しかし、異種性という問題点が挙げられる。実際データベースにアクセスするためには以下の問題点が存在する。データベース毎に提供されるインターフェースが異なるため、アクセスするデータベース毎にインターフェースやアプリケーションを作成しなければならないという問題である。そこで、ユーザが指定した検索要求に対し異種 DB クラスを用いて、その異種性を解消し情報源にアクセス機能を介して複数の情報源を把握する。各データベースに対する接続ドライバをクラス化し接続時にマッチングされた DB ドライバを呼び出すことにより、異種性を解消する。

なお、試作システムでは、異種性解消機能の実装として、拡張 JDBC を定義する (図 3)。

これは、通常の JDBC よりも一段抽象度の高いクラスライブラリで JDBC を補完するフレームワークである。これにより、既存のデータベース間や異種間のデータベースに直接問い合わせることができ、固有機能の異種性を吸収している。

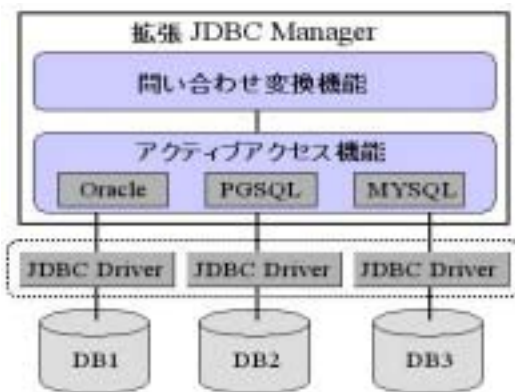


図 3 拡張 JDBC 構成図

次に拡張 JDBC の実現する機能を示すために、標準的

な JDBC との差異について述べる。標準的な JDBC は、主に次のような機能を実現している。

- システムレベルでのデータ交換
- 多種多様なデータベースシステムを対象として、統一的なプログラムコードで利用することが可能。

拡張 JDBC は上記に加えて、次のような機能を実現する。

- メタデータによるデータベース全体的な情報を取得し、アクティブに利用可能とする。
- 多種多様なデータベースシステムを対象として、統一的な問い合わせ言語で利用することができる。

このように、拡張データベースは、標準的な JDBC では実現されていない機能を、JDBC よりも一段抽象度の高いクラスライブラリで実現する。これにより、既存の JDBC に一切変更を加えずに機能の追加や問い合わせ言語の統一が可能となる。

また、異種データベースへのアクセスを柔軟なもとするために以下の機能を付加している。

- DB 接続設定をプロパティファイル管理
- 複数種類の DB 接続設定を登録可能
- JDBC API の利用により、独自拡張された複数の DB 接続方法を提供

DB 接続はプロパティ・ファイル管理を行うことで容易に設定を行うことができる。また、独自に DB 登録を行うことができ、複数の接続が容易となる。

5.3 拡張 (URL) クエリ方式

異機種間 DB デバイスの通信方式として SQL 言語があるが、標準化仕様はあるものの DBMS ごとの方言、機能制限が存在し、標準的 SQL の使用では十分ではない。そこで、データベース固有の問い合わせ言語を URL 埋め込み形式で抽象化する。このことで、異種機問い合わせ言語間を統一的に利用でき異種性を吸収することができる。

さらに、多種多様なデータソースに散在したデータの多くの問い合わせに答えることができる。また、様々な形の問い合わせを処理できる形に補正することも必要となる。そこで、既存データベースと異種データベース間の異種問い合わせを抽象化する。

拡張項目を以下に示す。

- 1) 複数の DB にまたがって問い合わせ処理を行う
- 2) 標準 SQL およびベンダ SQL を統一
- 3) マルチ同種/異種 DB へ並列クエリ
- 4) 独自クエリステートメントを定義および生成
- 5) DB 設計・システム設計処理を可能とする
- 6) URL 上で直接 SQL を記述し DB にアクセス

本拡張 (URL) クエリを利用することでデータ統合や検索など柔軟性をもたせることで、SQL の方言や異種性を解消する。

以下に、拡張(URL)クエリを説明する。

データベース TEST_DB01 のテーブル T01 とデータベース TEST_DB02 のテーブル T02 へ接続し HTTP アクセスを行うものとする。データの抽出として SELECT ステートメント以外にデータの挿入 (INSERT)、データの削除 (DELETE)、データの編集更新 (UPDATE) 等問い合わせを可能とする。

問合せ例 1 シングルDB_シングルクエリ

```
http://localhost/test01DB.DBw?SQL=Select+ * +  
from + 't01'+WHERE . . .
```

あるいは

```
http://localhost/DBWeb?DB=test01_DB,SQL=Select  
+ . . .
```

問合せ例 2 シングルDB_マルチクエリ

```
http://localhost/DBWeb?DB=test01_DB,SQL1=show  
+ tables,SQL2=selct+ * +from+'t01'
```

問合せ例 3 マルチDB_シングルクエリ

```
http://localhost/DBWeb?DB=test01_DB+test02_DB,  
SQL=show+tables
```

問合せ例 4 マルチDB_マルチクエリ

```
http://localhost/DBWeb?DB=test01_DB+test02_DB,  
SQL1=select+ * +from+'test01_DB  
. . .  
t01'+','+'test02_DB.t02'+where+ . . . ,SQL2= . . .
```

上記で URL 上に直接 SQL の SELECT 文を記述してテーブルにアクセスを行う。SQL 引数には SELECT ステートメントをしているが URL なので、スペースの代わりに「+」を利用している。また、URL から、スキーマ、テンプレートを指定して検索することが可能である。

5.4 データ統合方式

多種多様なデータベースを個別に検索するだけでなく、取得した情報を統合して提供を行う上でデータの共有化が重要である。特定の情報源を他から利用するためには、利用される (公開される) 情報源を、ある共通の交換モデルに変換しなければならない。そこで、多種多様に散在する各異機種間の情報源のデータを選別し、形式を統一した上でデータ統合を図る。その際、フォーマット変換を行うことで異種性を解消し、

統一されたデータの集合となり統合あるいは検索を行うことを可能とする。

変換モジュールにはラッパーを用いてデータ変換を行う。特定のアプリケーションやデータに依存した部分を包み込むことによってインタフェースを共通化し、データの互換性を実現する。クライアントが SQL の問い合わせを出すことを通じて、ラッパーは JDBC API 経由でアクセスされる。ラッパーを実現するために、Web ページから抽出した属性を処理する複雑化された言語に適応する抽出ルールを用いる。さらに、スキーマに適応した初期化を変換するためにデータ変換ルールを用いる。また、完全なデータを保証するためにデータルールを用いる。

多種多様なデータソースと相互作用する軽い Java アプリケーションはラッパーへダイレクトに接続できる。アプリケーションは独自のスキーマを持ち、離された JDBC ソースとして互いのデータソースを見ることができ。さらに、それらをつなげ、取得したいデータを結合することができる。DB ラッパーの実際の機能としてはデータベースに対する (共通形式の) 問合せを、その DB 特有の問合せに変換。問合せの結果を共通形式のオブジェクトに変換である。

以下に、ラッパーを用いる例を説明する。

集合演算定義 (データ集合例)

データベース TEST_DB01 のテーブル T01 とデータベース TEST_DB02 のテーブル T02 の2種類のデータベースを和集合するものとする。

```
http://localhost/DBWeb?DB=test01_DB+test02_DB,  
SQL=select+ * +from+'t01'+union+  
select+ * +from+'t02'
```

関係演算定義 (データ統合例)

データベース TEST_DB01 のテーブル T01 とデータベース TEST_DB02 のテーブル T02 とデータベース TEST_DB03 のテーブル T03 の3種類のデータベースを結合するものとする。

```
http://localhost/DBWeb?DB  
=test01_DB+test02_DB+test03_DB,SQL=select+ * +fr  
om+'t01'+','+'t02'+','+'t03'+where+ . . . +AND+ . . .
```

6 試作システム

6.1 公開データベース群の適用

ここでは、公開データベース群を対象とした本システムの適用について述べる。カテゴリー別に分類し公開されるデータベース群の適用データを下記の表 1 に示す。公開情報データを元にアクセス管理を行う。

ID	公開名	カテゴリ	位置情報	内容メタデータ
1	共有ブックマーク	一般	127.0.0.1	"bookmark", "favorite"...
2	PCコスト	個人	133.10.60.30	"personal computer", "cost", "cpu"...
3	ミュージック	音楽	61.122.35.61	"music", "artist", "title"...
14	ファンクラブ	団体	133.10.60.29	"group", "fanclub"...
16	書籍データ	所蔵	202.152.21.2	"book", "library"...

表1 公開データベースの適用データ

6.2 試作

上記の公開データベースデータを用いプロトタイプを試作した。プロトタイプで使用したデータベース群は、フリーウェアであるMySQL, PostgreSQL, MS-ACCESSである。システムは、Javaにより実装され、また、JavaによるAPIが実現されている。

Webブラウザ上でURLから直接SQLによる問合せを行い、動的にビューを生成・結果表示する。選択されたビューを実行するアプリケーションにServlet・JSPを用いて実現している。プロトタイプアプリケーションの画面を図4に示す。

図4に示す画面例では、では公開データベース群の一部を表示する。メタデータを元にカテゴリ別に分類されて表示されている。利用者は選択的に情報源にアクセスを行うことができる。では、URL上あるいは入力フォームから直接SQLによる問合せを行う。動的にビューを生成・結果を表示する。個別あるいは並列クエリも可能である。では、それぞれの問合せに対しての結果を表示する。結果欄のボタンでデータ内容を動的に追加・削除・変更あるいは他データに引用することも可能である。また、複数のテーブルにアクセスを行う場合、データ変換・統合方式により結果表示される。



図4 プロトタイプ利用例

まず、管理者は管理者モードを選択し公開DB群への登録を行う。登録内容はXMLで記述されている。このため、図4の画面によりWeb上で登録を行うことができる。そののち、管理者は登録したDBを選択しアクセ

ス権などの詳細設定を行う。

一方、利用者は利用者モードを選択し図4のに表示されている公開DB群から取捨選択する。公開DB群から取得したい情報を含んだDB検索はに表示されているメタデータを基にで検索要求を行い、に表示される検索結果で確認を行う。また、データ統合を行うには、公開DB群から複数のDBを選択しテーブルのスキーマ属性などの統一をで行う。統合後は他の公開DBと同様にで検索要求を行う。さらに、統合したデータをDBへの登録・交換も可能である。

6.3 考察

本プロトタイプは、ネットワークに散在するデータベース群を一つの枠組みとし、自在に協調動作させることが可能である。これまで、広域分散環境に対応したミドルウェアや標準インタフェースは既に存在するが、多くのアプリケーションは独自仕様であり、利用範囲が狭く応用性に欠けるとい問題がある。一方、本稿で提案した方法は、既存のインタフェースを基本に個人利用者でも公開されたDBを活用して、新たなDBを公開できる仕掛けを実現する点に特徴がある。プロトタイプの試作および利用により、個人利用者でも広域分散環境を想定した個人DBを連携した個人データベースを容易に構築できることが確認できた。

7 おわりに

本稿では、個人DBの相互連携を実現するためにシステム間を統合するインタフェースを設け、データベースの集合処理やHTTPアクセスなど自由度のあるデータベースシステムを利用することにより、管理者や利用者の作業を軽減と情報検索の効率化を図る個人DB提供利用支援方法を提案した。既存データベースと異種データベース間での異種性を吸収することにより固有の機能や問い合わせ言語を統一的に利用することが可能となった。

参考文献

- [1] JDBCMTtechnology
<http://java.sun.com/products/jdbc/>
- [2] M.W.Bright, A.R.Hurson and S.H.Pakzad: A Taxonomy and Current Issues in Multidatabase System, IEEE Computer vol. 25 No.3, pp.50-59 (1992)
- [3] 星野 隆 他: 「マルチデータベース環境における情報資源管理と検索方式」, 第114回DBS研究会, 1998
- [4] 北川 博之 他: 「異種情報源統合利用環境 Infowaver」, 第59回情報処理学会, 全国大会 No. 3-243 1998
- [5] Bojiang Liu, Kazumasa Yokota, and Nobutaka Ogata, Specific Features of the QUIK Mediator System, IEICE Transactions on Information and Systems, vol.82, no.1, pp.180-188, Jan., 1999.