

異種時制データベースシステムの統合的利用方式の設計

1 Q-8

柳川 史彦[†]

北川 博之^{††}

鈴木 孝幸[‡]

[†] 筑波大学 理工学研究科

^{††} 筑波大学 電子・情報工学系

[‡] 筑波大学 工学研究科

1 はじめに

近年、データベース利用の高度化に伴い、時間情報を管理することが可能な時制データベースシステムが登場している。一方、近年のネットワーク技術の発展により、各種のデータベースや情報資源をネットワークを介して統合的に利用する技術の重要性が認識されている[1]。時間情報を扱うデータモデルの中にもリレーショナルデータモデルやオブジェクトデータベースなど、多様なものが存在する。時制データベースシステム間の統合利用に関する研究として[2]が挙げられるが、統合のための理論的枠組を示したに過ぎない。

そこで、本研究では時制オブジェクトデータベースシステムと時制リレーショナルデータベースシステムを対象としてプログラミング言語レベルでの統合を試みる。

2 対象とする時制データベースシステム

我々研究グループの開発した時制オブジェクトデータベースシステム POST/C++[3] 上から、時制リレーショナルデータベース POSTGRES[4] 上のデータを利用する事を考える(図1)。

2.1 POST/C++

POST/C++はC++言語上のオブジェクトに永続性を持たせるとともに、その履歴を管理する事が可能なメモリマップ方式に基づくオブジェクトデータベースシステムである。各オブジェクトはトランザクションタイムによる時間情報を持ち、以下の操作が可能である。

- snapshot(db, ptr, time)
ptrで指されるオブジェクトの時刻 time の状態を返す。
- set_iterator(db, ptr, time)
ptrでさされるオブジェクトのイテレータクラスを作る。イテレータには next, previous といったメンバ関数が用意されており、直前、直後の状態を求める等の事ができる。

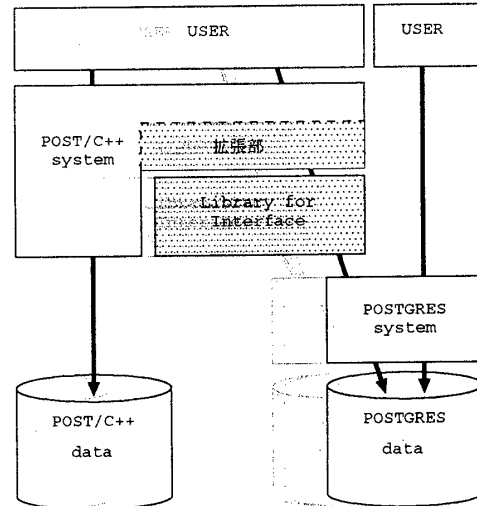


図1: 異種時制DBMSの統合利用

2.2 POSTGRES

POSTGRESは拡張型リレーショナルデータベースシステムであり、プログラミング言語からはSQLを拡張した言語を用いてアクセスできる。

POSTGRESは標準のリレーショナルモデルに比べて様々な面で拡張されているが、今回利用するのはその内の時間情報管理の機能のみで、その他の部分は標準リレーショナルモデルとして扱う。

3 リレーションのマッピング

POST/C++はC++言語上のオブジェクトを扱うためのシステムであるため、POSTGRESのリレーショナルモデルを扱うためには、リレーショナルモデルをC++言語上のオブジェクトへと変換する必要がある。ここでは構造体にマッピングする事を考える。

SQLによる問合せの発行や結果を構造体に代入する作業はユーザが行う必要はなく、問合せを発行する際に必要となる情報を関数で与えるだけでよい。

POST/C++からPOSTGRESのデータを読み出す手順は次のようになり、図示すると図2のようになる。

1. ユーザはリレーションのマッピング先の構造体を定義しておく
2. ユーザがライブラリにリレーション名と主キー値を与える
3. ライブラリはリレーションのどの属性が主キーであるかを調べ、問合せを発行する

Design of an integration scheme of heterogeneous temporal DBMS's

Fumihiko YANAGAWA[†], Master's Degree Program in Science and Engineering, Univ. of Tsukuba

Hiroyuki KITAGAWA^{††}, Institute of Information Sciences and Electronics, Univ. of Tsukuba

Takayuki SUZUKI[‡], Doctoral Degree Program in Engineering, Univ. of Tsukuba

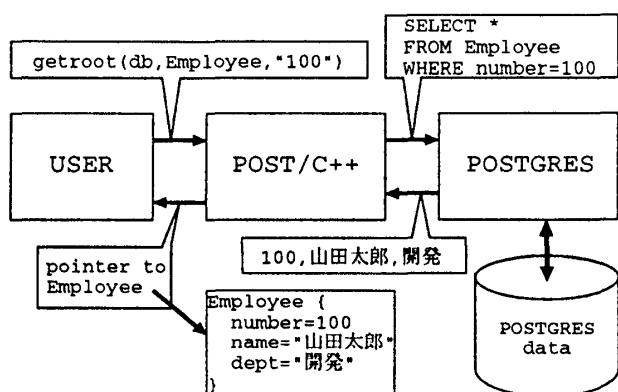


図 2: getroot 関数の動作

4. ライブラリは問い合わせの結果を解析し、構造体のフォーマットでメモリにマップする
5. ライブラリはポインタとリレーション名、主キー値の関係をメモリ上の対応表に格納した後、ポインタをユーザに返す
6. ユーザはポインタを受けとり、マップされた構造体にアクセスが可能となる

ポインタと、そのポインタが指すオブジェクト情報との対応は、履歴管理時と更新時に必要となる。

ユーザは以下の関数を用いてマッピングに必要な情報をライブラリに与える。

- `get_root(db,type,key)`

データベース `db` 中の `type` で表されるリレーションから主キー値が `key` であるようなリレーションを構造体にマッピングし、そのポインタを返す。条件に合うリレーションが見つからない場合は NULL ポインタを返す。

4 履歴管理

POST/C++ から POSTGRES 上のデータの過去の状態を透視的に読みだしたい。そのために、次の関数を用意する。これらの関数が呼び出された時には、ポインタで指される構造体が以前に検索に使われた条件を用いて、それに更に時間情報を付加した問合せを発行する。

- `snapshot(db, ptr, time)`

`ptr` で指される構造体に対応するリレーションの `time` での状態を検索して構造体にマップし、そのポインタを返す。例えば、`Employee.number=100` であるようなリレーションが `getroot` 関数によって、`XXXX` 番地にマッピングされていた時、`snapshot(db,XXXX,"Jan 1 1996")` が実行されると、ライブラリは以下のような問い合わせを発行する。

```
SELECT *
FROM Employee['Jan 1 1996']
WHERE number='100'
```

- `set_iterator(db, ptr, time)`

`ptr` で指される構造体のイテレータクラスを作りそのポインタを返す。その時点では時刻 `time` の時に有効だったリレーションを指しており、メンバ関数 `next`, `previous` などを使って時間軸をたどることができる。ここで `set_iterator(db,XXXX,"Jan 1 1996")` と実行された時、以下のような問い合わせが発行される。

```
SELECT *
FROM Employee['epoch', 'now']
WHERE number='100'
```

問い合わせ中の `epoch` はシステムクロックの始まりを表し、`now` は現在を表している。この問い合わせによって、`number='100'` である `Employee` のリレーションが全て求められる。リレーションはメモリにマップされ、時間順にイテレータクラスに格納される。最後に、与えられた時刻にカレントであったリレーションに内部カーソルをセットし、イテレータクラスへのポインタが返される。ユーザはイテレータクラスのメンバ関数を使うことによって、目的のオブジェクトの履歴をたどる事が出来る。

5 問題点及び今後の課題

今回、リレーションのマッピング先である構造体はユーザが定義する事になっているが、POSTGRES のカタログ情報を調べる事によって自動定義が可能ではないかと考えられる。

異種 DBMS 間では時間の概念さえ異なっている場合もある。例えば異なる時間単位(日、月など)での運用されている場合や、トランザクションタイムとバリッドタイムの問題についても考える必要がある。

参考文献

- [1] W.Kim, I.Choi, S.Gala and M.Scheevel, "On Resolving Scheme Heterogeneity in Multidatabase Systems," *Distributed and Parallel Databases 1*, pp. 251-279, 1993.
- [2] X.S.Wang, S.Jajodia and V.S.Subrahmanian, "Temporal Modules: An Approach Toward Federated Temporal Databases," *Proc. of ACM SIGMOD Conf.*, pp. 227-236, 1993.
- [3] 鈴木, 北川, 林, 柳川, "時制永続オブジェクト管理システム POST/C++ の構築と基本性能評価," 日本ソフトウェア科学会第 12 回大会論文集, pp. 113-116, 1995.
- [4] "POSTGRES95 manual"