

オブジェクト指向データベースにおけるビュー機能の試作

1E-8

福本 徹 中西 通雄 橋本 昭洋

fukumoto@ics.es.osaka-u.ac.jp

大阪大学基礎工学部情報工学科

1 はじめに

オブジェクト指向データベース（OODB）では、スキーマ構造はクラス階層に基づいている。従ってデータベース設計者がデータの意味をより忠実にスキーマに反映させようとする、そのスキーマの自由度は低くなる。またデータベース設計者の見方と利用者の見方とは異なるのが通常である。そこで関係データベース（RDB）におけるビューの概念をOODBに導入する。

しかしながら、OODBにおけるビュー機能は明確な定義がなく、現在のところビュー機能を実装した商用OODB管理システム（OODBMS）は存在しない。研究段階では、GemStoneを用いてビュー機能を実装している例がある[2]。しかしこの例は、ビューを用いた検索質問については言及していない。

本稿では、商用OODBMSであるVersantとVersantに付属したオブジェクトSQL（OSQL）とを用いて、ビューを定義する機能とビューを用いた検索を処理する系とについて述べる。

2 ビュー機能のモデル

2.1 仮想クラス

Abiteboulらは実クラスに対して特化または汎化を行ない、仮想クラスを定義することを提案している[1]。特化の場合は仮想クラスは実クラスの直接の子クラスとなり、汎化の場合は直接の親クラスとなるので、仮想クラスと実クラスとの間のクラスの親子関係は一意に定まる。

OODBにおいて仮想クラスを定義するには最低限の機構として、クラスの特化・汎化・射影を行なうことが必要であると考え。このうち射影については次の議論から、属性に対するアクセス権を用いることとした。

仮想クラスを定義する操作としては、以下のことを考えて設計を行なう。新しいオブジェクトを生成しないことと、かつ、データベースの実クラス階層に収ま

ることである。OODBではスキーマ構造は、属性やメソッドの継承によってクラス階層を構成しているからである。またオブジェクトの管理は、VersantDBMSに任せることとした。

SQLのSelect文とWhere文を併用することを考える。Selectの意味では属性やメソッドの数が減少している、仮想クラスは実クラスの親クラスである。Whereの意味では特化なので、仮想クラスは実クラスの子クラスである。以上のことから、仮想クラスがクラス階層には収まらなくなる。セキュリティ上の問題でSelectを使うのであれば、属性やメソッドに対するアクセス権を認証する機能をビュー機能とは別に実装して制限をかけるべきであると考え。

新たに属性を追加することを考える。OODBのスキーマ設計と同じように、属性を親クラスから継承し子クラスで新たに付け加える。この場合、新しいオブジェクトを生成することになる。つまり仮想クラスの各インスタンスごとに新しい属性に対応するオブジェクトの管理を、ビュー機能のシステム側で行なう必要がある。従ってOODBMSを利用する意味がなくなってしまう。

結局本システムで定義できる仮想クラスとしては、クラスの特化と汎化を採用した。

2.2 仮想スキーマ

データベース上に存在する実スキーマと仮想クラスとを統合したスキーマを仮想スキーマという。図1に、特化または汎化によって仮想スキーマを構築する例を示す。実線はデータベースに実際のスキーマを、点線は仮想クラスと仮想クラスに関するクラスの親子関係を表す。図1に、特化または汎化によって仮想スキーマを構築する例を示す。実線はデータベースに実際のスキーマを、点線は仮想クラスと仮想クラスに関するクラスの親子関係を表す。属性やメソッドはクラス間で継承する。特に汎化による仮想クラスの型は、仮想クラスの導出元である実クラスの直上の親クラスと同じ型とする。実クラスの型の共通集合としないのは、属性の名前が同じでも型が異なることがあり得るからである。

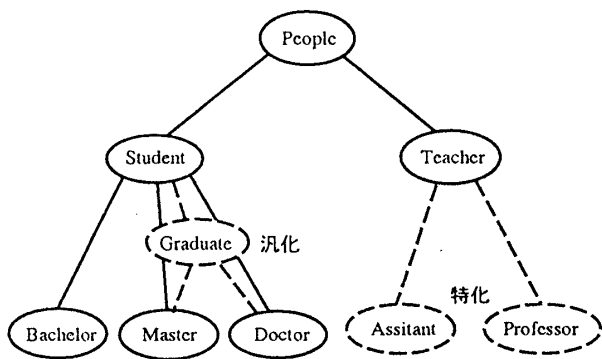


図 1: 特化と汎化による仮想スキーマ

3 Versant を用いた実装

以下に述べるシステムは、現在 C 言語を用いて作成中である。

3.1 仮想クラスの定義

仮想クラスは我々が設計中である OSQL-like な言語で定義する。仮想クラスに関するスキーマ定義ファイル(仮想クラスファイル)を出力する。ファイルの中身は、仮想クラスの名前、仮想クラスの定義となる質問、仮想クラスの親・子クラスである。このファイルを仮想クラスについてのデータベースとして用いる。以下に仮想クラスを定義する例を示す。

```
/* 実クラス Master, Doctor を汎化した */
/* 仮想クラス Graduate を定義する。*/
create view Graduate from Master, Doctor;
```

```
/* 実クラス Teacher を属性 Jobtitle の値によって、*/
/* 特化した仮想クラス Professor を定義する。*/
create view Professor from Teacher
where Jobtitle = "Professor";
```

なお属性やメソッドの射影は、アクセス権を認証する機構として実現できると考えるが、今回は実装しない。

3.2 仮想スキーマの構築

仮想クラスの定義は、実クラスあるいは既に存在する仮想クラスに対する特化あるいは汎化だけなので、仮想クラスの親/子クラスは一意に決定することができる。Versant の DBMS が利用するスキーマ定義ファイル(スキーマ名プラス拡張子.sch)には、すべての実

クラスについて、すべての親クラスが記述されている[3]。このファイルを実データベースと常に同期がとれているものと仮定して参照し、仮想スキーマを構築する。作成した仮想スキーマはファイルとして管理する。

3.3 検索処理

ユーザは仮想クラスを含んだ OSQL を検索文として入力する。システムは仮想スキーマファイルを参照し、仮想クラスを記述した部分を仮想クラスを含まない OSQL による検索質問に展開する。これを Versant への入力とする。仮想クラスが汎化により作成されている場合は、実クラスの検索結果の論理和をとる。特化による場合は、Where 文として展開して結果の論理積をとる。検索した結果を利用者に返す。図 2 に本システムの概念を示す。

Versant から得られる検索結果を利用者に返すには、本システムで変換する必要がある。Versant の OSQL には、Select 文に最大値や平均値を計算するものもある[4]。例えば、属性の平均値をとる Select Avg 文がユーザからの質問として与えられたときは、条件に合うインスタンスの総数とその属性の値とを数え上げて、本システム側で計算を行い、その結果を利用者に返す。

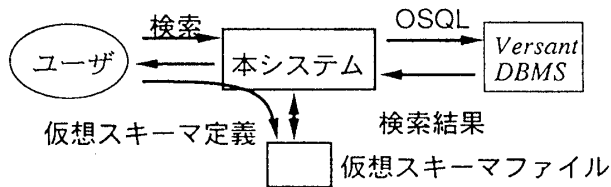


図 2: 本システムの概念図

参考文献

- [1] S.Abiteboul and A.Bonner, "Object and Views", ACM SIGMOD, pp.238-247, June 1991.
- [2] H.Kuno and E.Rundensteiner, "Developing an Object-Oriented View Management System", IBM CASCON, July 1993.
- [3] Versant Object Technology Corporation, Versant System Reference Manual Version 1.6, June 1991.
- [4] Versant Object Technology Corporation, Versant ObjectSQL Reference Version 2.0, June 1992.