

# オブジェクト指向DBMS - Odin

## - 導出クラス機構 -

2H-6

木村 裕 鶴岡 邦敏

日本電気(株) C &amp; C システム研究所

### 1. はじめに

従来のオブジェクト指向DBMSでは、オブジェクト指向言語におけるクラス(または、型)や抽象化表現を導入し、さらにデータベースの分野での意味データモデルの研究成果を取り入れることによって、データベースの表現能力や拡張性を向上させた。一方、オブジェクト指向を強調する反面、従来のデータベースのいくつかの利点が失われてしまっている。たとえば、3層スキーマに見られるようなデータ独立性[TsKi89]、結合操作のような動的な関連付け機能[KiTs89b]、ビュー、等の有益な機能に欠けている。現在我々が開発中のオブジェクト指向DBMS-Odinでは、導出クラスという概念を導入しビュー機能を実現している。本稿では、この導出クラスの実現方式について述べる。

以下では、2節で導出クラスを定義し、3節で導出クラスにおけるメソッド検索の特徴について記述する。

### 2. 導出クラス

オブジェクトは、ただ1つの概念クラスに属する。また、オブジェクトは複数の導出クラスに属することができる。すなわち、導出クラスのインスタンスは、概念クラスのインスタンスを直接参照する。この点は、DBからのコピーによって実現される関係DBと異なる。

導出クラスとインスタンスとの関連付けは、選択・結合・グループ化操作がインスタンスに適用されたときに行われる。関係DBでは、複数の集合演算の結果としてできる表がビューであるが、Odinでは、集合演算の度に導出クラス(ビュー)が生成される。また、関係DBとは異なり、そのトランザクションが終了するまで、導出クラスは存在する。

集合演算の結果の導出クラスは、すべてのクラスのスーパークラスの頂点に位置するクラス `standar-object` のサブクラスとして生成される。

### 3. 導出クラスのメソッド

概念クラスでは、メソッド(以下、基本メソッドと呼ぶ)とインスタンス変数が定義できた。一方、導出クラスはメソッドのみが定義(付加メソッドと呼ぶ)できる。さらに、元となったクラスのすべてのメソッドは、導出クラスに伝搬され利用することができる。これは、選択演算や結合演算の結果のインスタンスは、元の概念クラスのメソッドがそのまま使用できることを意味する。ただし、演算で可視条件が指定されている場合、指定された名前だけが伝搬され、メソッドとして利用できる。これは、関係DBにおける2つの表の結合で、射影条件を満足する2つの表の属性が参照可能になることと類似している。

伝搬されるメソッドは、クラス  $S_i$  のメソッド集合を  $m(S_i)$ 、DB操作用メソッドを提供した結果の導出クラスを  $D$  とする。このとき、伝搬されるメソッドは、

$$m(D) = \bigcup_{i=1}^n m(S_i)$$

となる。ただし、 $i=1$  のとき `select` における伝搬、それ以外るとき `join` による伝搬を表わす。Odinでは結合・グループ化演算により構造化のためのオブジェクトが作られるが、この伝搬機構により既に定義されたメソッドは変更することなく使用することができる。しかし、関係DBとは異なり、新しく定義された付加メソッドはこの構造を意識しなくてはならない。

### 4. 実現方式

#### 構造

図1にクラスの基本構造を示す。クラス・オブジェクトの保持する重要な属性は、`instances`, `visible-methods`, `derived-classes` である。`instances` は当該クラスに属するインスタンスを持つ集合オブジェクト、`visible-methods` は参照可能なメソッド名のリスト、`derived-classes` は当該クラスから生成された導出クラスのリストを保持する。

検索

図2は、選択演算によって生成された導出クラスとインスタンスの関係を表わす図である。このインスタンスにメソッドが送られた場合、メソッドを次のように探索する：

- (1)当該インスタンスの属する親クラスを見つける。
- (2)親クラスのderived-classesにメッセージを送信したトランザクションから参照されている導出クラスがあるか調べる。
- (3)このような導出クラスが複数ある場合、最新のを選び、その導出クラスに付加メソッドが存在するか調べる。
- (4)付加メソッドがあれば、それを起動する。付加メソッドがない場合、対応するメソッド名が参照可能リスト中にあるか調べる。参照可能リストになければ、エラーを出力する。
- (5)参照可能リスト中に存在するとき、親クラスが概念クラスの場合、概念クラスのインスタンス・メソッドが起動される。

図3は、結合演算によって生成された導出クラスとインスタンスの関係を表わす図である。結合導出クラスの生成時に、元のクラスのメソッドと同名のメソッドが自動的に定義され、これらは単に元のクラスのメソッドを呼び出す。結合インスタンスにメソッドが送られた場合、メソッドを次のように探索する：

- (1)~(5)を実行する。
- (6)当該インスタンスの親クラスが結合導出クラスの場合、メッセージに対応する付加メソッドが存在するか調べる。
- (7)付加メソッドが存在すれば、そのメソッドを起動する。ないときは、基本メソッドが起動され、さらにそのメソッドから当該インスタンスの指す

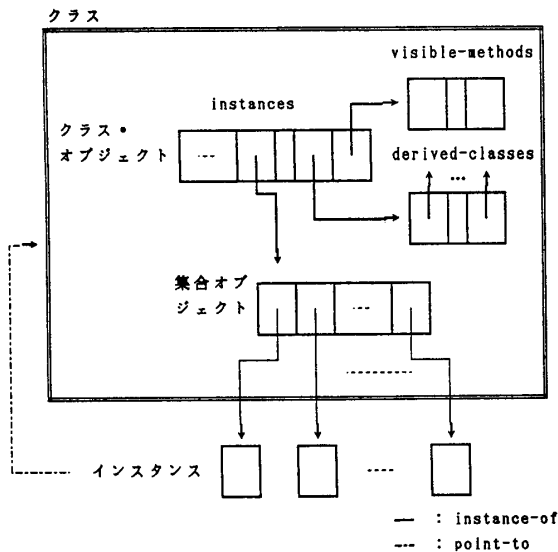


図1 クラスの基本構造

インスタンスへ同一名のメッセージを送信する。(8)ステップ(1)へ戻る。

5. 永久導出クラス

導出クラスを複数の利用者で共有するために、トランザクションが終了しても存在する導出クラス—永久導出クラスを提供する。このクラスの宣言は、derived属性を持つクラス定義と、導出クラス生成規則を持つderiveメソッドからなる。この導出クラスのインスタンスは、クラスに対してderiveメッセージを送信することにより生成される。

例：

```
(define-class J-Car () :category 'derived ())
(define-method derive J-Car ()
  (select 'Car '($1)
    '(= (made-in $1) "Japan")))
(derive J-Car)
```

6. おわりに

本稿では、オブジェクト指向DBMS—Odinにおけるビュー機構—導出クラスの定義と実現方式について述べた。関係DBにおけるビューとオブジェクト指向DBの導出クラスとを比較すると次のようなことが分かる：関係DBのビューはビューの定義式に名前を付けてを共有し、この名前を参照したときに式を実行し値をコピーしたものであるのに対し、導出クラスは選択・結合演算の結果生成され、そのインスタンスは元のクラスのそれを直接参照する。さらに式の最終結果の導出クラスに名前を付けて共有する。

参考文献

[BaKK88] J.Banerjee, et.al. "Queries in an Object-Oriented Databases," DE'88.  
 [KiTs89a] 木村, 鶴岡 "オブジェクト指向データベース管理システムの柔軟性について," 第38回全国大会 1989/3.  
 [KiTs89b] 木村, 鶴岡 "オブジェクト指向データベース管理システム—Odin オブジェクト操作言語," 第39回全国大会 1989/10.  
 [TsKi89] 鶴岡, 木村 "オブジェクト指向データベース管理システムの物理構造仮想化方式" データベースシステム研究会, 1989/9.

C:概念クラス  
 S:選択導出クラス  
 J:結合導出クラス

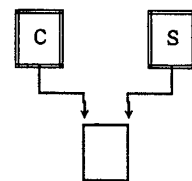


図2 選択導出クラス

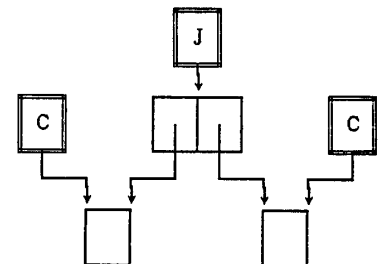


図3 結合導出クラス