

# 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

2R-5

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

近年、オブジェクト指向データベース管理システムが、数多く提案されている[1,3]。オブジェクト指向データベース管理システムで、他のモデルを提供するシステムと異なる最大の特徴は、利用者プログラムだけでなくシステム自身の拡張性と柔軟性にある。これまでのオブジェクト指向DBMSでは、従来のオブジェクト指向システムに、永続的なオブジェクト、トランザクション、版管理を追加することによって、オブジェクト指向のモデルとDBMSの機能を実現している。

しかし、今までのオブジェクト指向DBMSでは、物理ファイルとそのシステムが提供しているメソッドを使用してアクセスするので、新しい構造の物理ファイルや従来から存在するOSの管理下にあるファイルや関係データベース・ファイルをアクセスするためには、自分でそのアクセス・メソッドを記述することになる。従って、利用者は、自分のアプリケーション以外に、システムの物理的な所まで立ち入って、プログラムをつくらなければならない。

本稿では、ファイル構造やプログラミング環境の変更に柔軟に対処できるオブジェクト指向データベース管理システムの構成方法について考察する。

## 2. 従来の問題点

従来のオブジェクト指向DBMSで、新しいオ

ブジェクトをアクセスするためには、アクセスのためのメソッド中に、open,close,get-next-block,等の従来DBMSでも使用されていたファイル操作コマンド(i.e.メソッド)を記述しなくてはならない[2](図1)。プログラマは、対象オブジェクトのファイルブロックをメモリに取り出し、さらにそのブロック内のレコードの格納構造を意識して、目的とする項目を取り出す必要がある。また、メモリ上でのデータは、プログラミング言語によって異なるので、同じオブジェクトでも各言語用にメモリ・データをアクセスするメソッドを記述しなくてはならない。

このようなオブジェクト指向DBMSのアーキテクチャでは、従来のDBMSと比べて、プログラマの生産性は向上しない。プログラマにとっては、使用しているプログラミング言語の記憶データ参照方法だけで、データベースを参照できることが望ましい。

## 3. 格納形式の変換

上記の問題点を解決するために、図2のようにDBMSのアーキテクチャを3つのクラスの階層で構成する。

まず、1つの物理ファイルはファイルの性質を記述するクラス(以降ファイル・クラスと呼ぶ)のインスタンスである。このクラスは、ファイル領域およびファイル属性(i.e.ファイル制御プロ

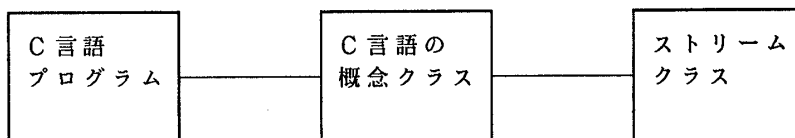


図1 従来のDBMSの構成

ック)や、ファイル・ブロックのアクセス方法、対応する概念クラス、そのクラスのインスタンス・アクセス方法、そのインスタンスのインスタンス変数のアクセス方法、等のメソッドを持つ。

2番目の層は、概念クラスの層である。この層のオブジェクトは、DBMSが扱う共通の形式を持つ。

3番目の層として、プログラム言語のデータ格納形式に関する知識データを持つクラス(以降、プログラム・メモリ・クラスと呼ぶ)を提供する。ここには、当該言語のメモリ表現形式で格納されたインスタンスと2番目の層の概念クラスのオブジェクトをプログラム・メモリ・クラスの形式に変換するメソッドを保持する。

プログラムの中でオブジェクトの参照があると、制御がDBMSへ移り、該当オブジェクトがメモリに存在しなければ、ファイル・クラスを経由して物理ファイルから概念クラスの形式にデータ変換して取り出される。さらに、この概念クラスのオブジェクトは、このデータをプログラム・メモリ・クラスのメソッドで、利用者が使用しているプログラミング言語環境に合ったメモリ格納形式へ変換する。この変換後、制御はDBMSから利用者プログラムへ戻り、そのオブジェクトを従来のプログラミング・システムの枠組みで参照できる。

4. 柔軟性

このようなDBMSは、次の2つの理由により従来のオブジェクト指向DBMSに比べ柔軟である:

(1)多くのプログラミング言語とDBMSとのインタフェースを取ることができる。新しい言語でデータベースをアクセスする要求があっても、プログラム・メモリ・クラスを1つ追加すればよい。

(2)既存ファイルの格納方式およびアクセス法の変更、ファイルの合成、ファイルの分割が、アプリケーション・プログラムを全く変更せずに可能となる。新しいファイル構造に対してもファイル・クラスを1つ追加するだけですむ。

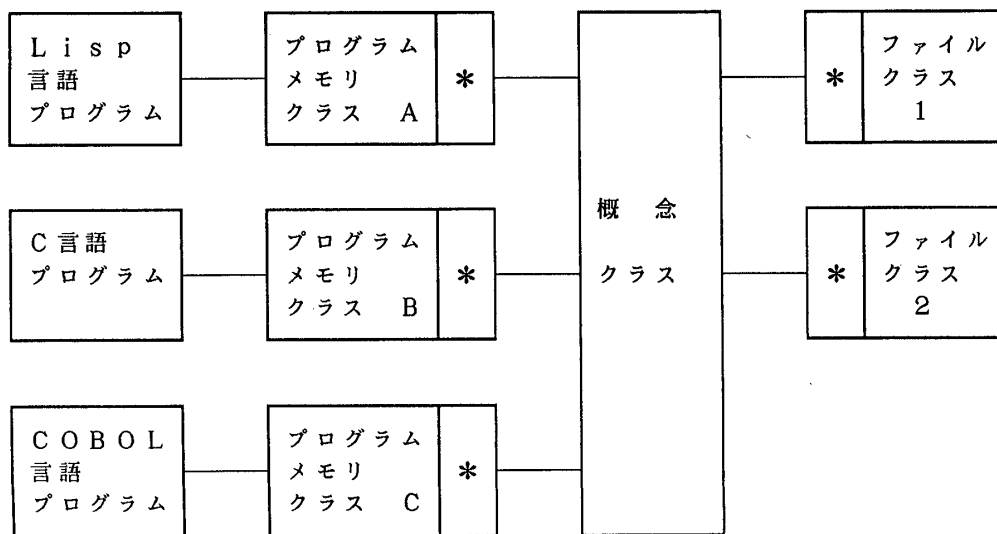
5. おわりに

本稿では、プログラムの負荷をできるだけ軽減するために、プログラム・メモリ・クラス、概念クラス、ファイル・クラスの3つの層によりオブジェクトを表現するオブジェクト指向DBMS構築方法を示した。

本稿の方式を支援するオブジェクト指向DBMSを作成し評価することが今後の課題である。

参考文献

[1]W.Kim,et.al."Integrating an Object-Oriented Programming System with a Database System", OOPSLA'88.  
 [2]D.Woelk,W.kim."Multimedia Information Management in an Object-Oriented Database System", VLDB'87.  
 [3]M.J.Carey,et.al."A Data Model and Query Language for EXODUS", SIGMOD'88.



\*はメソッド。

図2 提案方式のDBMS構成