

オブジェクト指向データベースの開発

1E-4

- システム構成と実現方式 -

土屋 武彦 脇園 竜次 川村 敏和 田中 立二

(株) 東芝 重電技術研究所

1 はじめに

プラント監視制御等のオンライン・リアルタイム・システムにおいてはその高速応答性を要求される事から、既存のデータベース管理システムを使わず専用のファイル管理システムを用いる事が多かった。

これに対してオブジェクト指向データベース(OODB)の出現により以下の機能を実現する事が可能になった。

- (1) 複雑、不規則な構造を持つデータベースの構築
- (2) データに付随した操作・処理・アルゴリズムのデータベース化
- (3) データベース言語とプログラミング言語の統一

これにより従来は適用の難しかったプラント監視制御などのオンライン・リアルタイム・システムへのデータベースの適応を考慮する事が可能になった。

そこで我々は、オンライン・リアルタイムの用途を含めた一般の応用システムに組み込んで使う事を想定し、「言語透過性、高速性、コンパクト」を実現目標とするオブジェクト指向データベース管理システム Odb を開発した。

Odb は C++ を基本言語とする OODBMS であるがその主な特徴は以下に示すとおりである [1]。

- (1) 言語透過性および高速性
- (2) 集合管理機能を一般化したデータモデル

本稿では、これらの特徴に対応したデータベース管理システムの実現方式について紹介する。

2 言語透過性および高速性

言語透過性および高速性は、永続性を記憶域クラスの属性と考え、オブジェクト相互の關係に C++ のポインタを用いて表現する事により可能となり、さらに永続/一時オブジェクトの統一的な扱いが可能となる。これらを以下に示すメモリ管理方式を導入する事により達成した。

2.1 永続オブジェクトのメモリ管理方式

永続オブジェクトバッファとして特定のメモリ領域を使用し、このメモリ領域内のオブジェクトをユーザ

に直接アクセスさせる事で、永続性を記憶域クラス属性として管理する体系を実現できる。

また、オブジェクト識別子としてこのメモリアドレスを用いる事で、通常の C++ プログラムと同様にオブジェクト相互の關係は C++ のポインタを用いて表現する事が可能になる。

この方式の場合、オブジェクトの再配置に伴うポインタ値の保守が必要になるが、これはページ読み込み時に一括して行えばオーバーヘッドを少なくする事ができる。再配置に伴うポインタ値の書き換えは Swizzling at Page Fault Time 方式を用いた [2]。即ち、「永続オブジェクトのポインタは永続オブジェクトを指す」という制約を導入し、オブジェクトを 2 次記憶へ書き出す時にポインタをチェックし、後の再配置のために必要な情報を保存する様にする。

2.2 トランザクションのためのメモリ管理方式

並行処理および障害管理のために計算機システムの仮想記憶管理機能を用いて、メモリ領域をマッピングしたキャッシュファイルをログファイルとして使用した。即ち、トランザクションの開始、終了、廃棄、ネスト処理などをキャッシュファイルとデータベースファイル間のページ転送により管理しデータの一貫性を保証している。ロック等の管理単位は高速性を考慮してオブジェクトではなく物理アクセス単位であるページとしている。

図 1 に以上のメモリ管理を中核とした、Odb のシステム構成を示す。この図に示す様に、作成・参照・更新・関係付けなどの永続オブジェクトに対する操作および管理はすべてメモリ上のオブジェクトの処理と同様に行う事ができる。また更にオブジェクトだけでなくスキーマやデータベース管理情報も同様にメモリ上に割り付けて操作する事により高速に処理する事が可能となる。

3 集合操作機能の一般化

Odb では、クラスに属するオブジェクトの集合としての外延とユーザ定義集合の概念の統合化をはかり、データベースにおけるデータモデルとオブジェクト指向におけるデータモデルを統一した仕様とした。

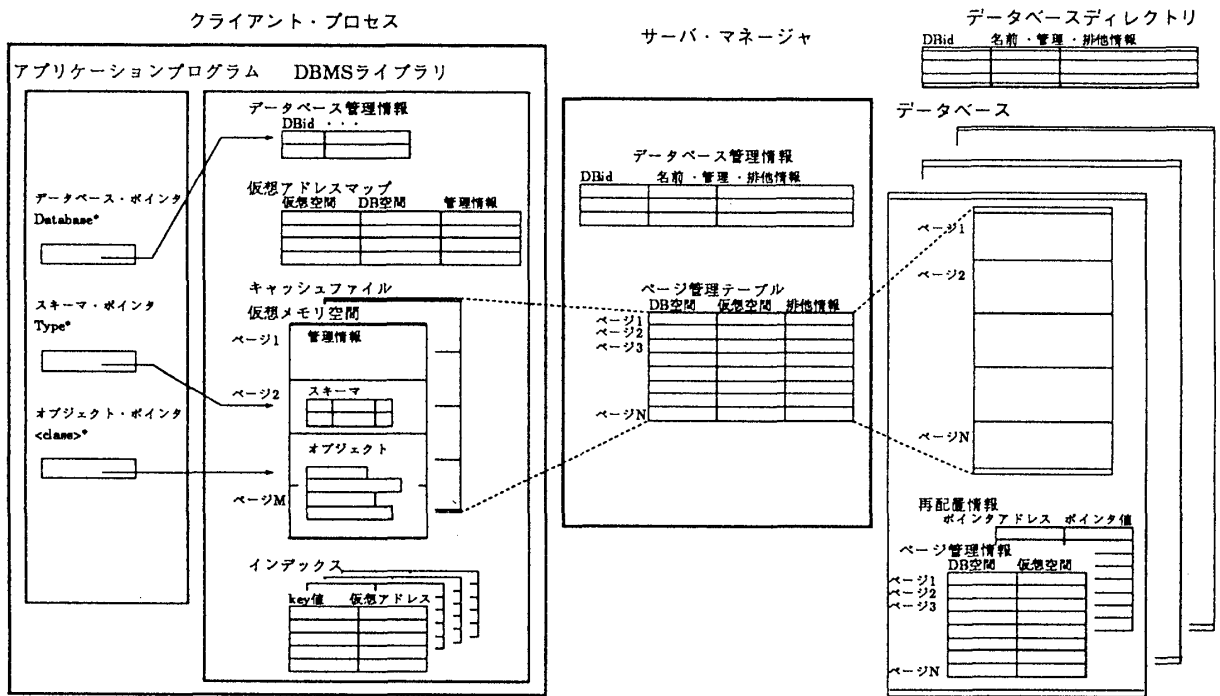


図1: Odbのシステム構成

図2に外延および集合の論理的な構造を示す。索引部は高速検索が必要な場合にユーザの指示により作成され、カーソルも必要に応じて複数個持つ事が可能である。集合演算および集合操作は全てこの構造をもとに実現されている。

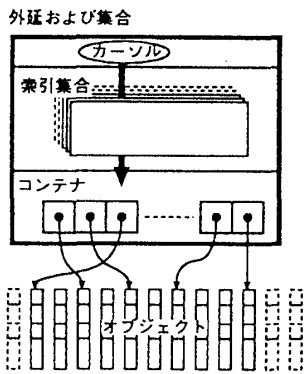


図2: 外延と集合の概念構成

現在のところ外延のコンテナ管理は2次記憶の物理アドレス (database, page, address) で管理しており、集合のコンテナ部はユーザオブジェクトと同様にポインタを用いて管理している。従って、外延 (Type クラス) と集合 (Bag, Set クラス) は同じユーザインタフェースを提供しているがコンテナに関する部分はそれぞれ別の処理体系となっている。

外延を管理するスキーマ管理 (Type) を集合管理 (Bag, Set) と同じ方式で実現する事ができれば、スキーマ管理も集合管理体系の中に組み入れられ、集合クラス体系と実装方式が一致する。これに関しては今後の研究課題である。

4 まとめ

計算機システムの提供するメモリ管理機構を利用し、永続オブジェクトのメモリと2次記憶上での表現を一致させた高速で言語透過性の高いOODBMSの実現方式を示した。本方式はメモリ保護機構を持つ計算機システムであれば容易に実現可能である。

現在、本方式に基づくOODBMSをUNIX上に開発した所である。今後は性能の評価および改良、他の計算機システムへの移植の検討等を行なう予定である。

参考文献

- [1] 脇園竜次、土屋武彦他：
オブジェクト指向データベースの開発 -機能仕様-
情報処理学会第48回全国大会、1994.
- [2] 鈴木慎司、喜連川優、高木幹夫：
永続的プログラミング言語P3Lの実装方式
情報処理学会、データベース研究会資料 89-9、1992.
- [3] W.Kim:
Introduction to Object-Oriented Databases
MIT Press, 1990.