

## データベース管理システム「沙羅」の実現方式

9G-1

石丸 知之 三堀 佐知子 山川 利治 植村 俊亮

東京農工大学 工学部 電子情報工学科

1. まえがき

われわれは、マルチメディア処理を主な目標としたデータベース管理システム「沙羅」を開発中である[1]。

沙羅は基本のデータモデルとして、オブジェクト指向データモデル[2]を採用し、「マルチクラスインスタンス」[3]と呼ぶ一つのオブジェクトに複数の表現を結びつける機構を導入した。

オブジェクト指向データモデルは、データ型の拡張性、カプセル化の機能、複合オブジェクトの表現といった点で、マルチメディア情報処理に適している。

沙羅ではデータベースを可搬にすることと、カプセル化の機能をデータ独立に達成することもめざしている。

2. 手続きとデータ独立

データ独立性とは、データベースの内部データ構造や論理的なデータ構造が変化しても応用プログラムを変更しなくても済む性質である。

オブジェクト指向の機能の一つである抽象データ型はデータ型を拡張する機構であり、抽象データ型で定義されたデータは型名と操作によってだけ参照できる。オブジェクト指向データベースでは、抽象データ型の機能を、データ独立性を満足しながら取り入れなければならない。また、データベースがCD-ROMなどの媒体によって配布されることも多くなるため、可搬であることも重要である。

オブジェクト指向データベースの一つの方式として、プログラミング言語にデータベース機能を強化するデータベースプログラミング言語がある。この方式では、データに関する手続きを従来のプログラミング言語と同じ環境に格納し、データベースのオブジェクトと関係づけているだけである。つまり、データベース内にデータとともにカプセル化される手続きを格納していないので、可搬なデータ独立性を達成していない。

もう一つのオブジェクト指向データベースの方式は、手続きをデータベースに格納する方法である。これは、POSTGRES、GemStone、ORION そして O<sub>2</sub> などで採用されている。これらの方法ではコンパイル後のプログラムをデータベースに取り込むためにデータベースの保護が十分でなかったり、問合せ言語をデータベースに

埋め込むために計算完備でなかったり、基本命令の実行単位が小さいため効率が犠牲となっていた。

これらの欠点を解消する方式として、沙羅ではデータベースに適した実行単位の大きい命令を含む仮想機械を設計し、データに結びついた手続きをその基本命令で表現して、データベースに格納する方式を採用した。

3. 「沙羅」アーキテクチャ

オブジェクト指向データベース管理システム「沙羅」を試作している。沙羅ではデータベースに手続きを、オブジェクトのメソッドとして格納する方法を採用した。格納する手続きは、たとえば検索命令などのデータベースを意識した実行単位の大きい命令を含む仮想機械の基本命令からなる。

3.1 「沙羅」のソフトウェア構成

沙羅のソフトウェア構成は、大きく、言語処理部、実行部、メモリ管理部に分割できる。

「沙羅双樹Ⅱ」は、応用とのインタフェース言語である。沙羅双樹Ⅱで、オブジェクトの定義、メソッドの記述、問合せの記述をすることができる。応用プログラムでは、応用に依存する記述だけをオブジェクトのメソッドの呼出しとして記述する。

実行部は仮想機械として実現されている。仮想機械はハードウェアから独立したデータ操作の命令集合をもつ。命令をハードウェアから独立させることでデータベース内の手続きを可搬にしている。これは異なった動作環境でも仮想機械を実装することでその環境にデータベースを実現することが可能であることを意味する。

メモリ管理部は、オブジェクトを単なるバイトの列として扱い、データ記憶領域を管理する。1次記憶と2次記憶の管理は、メモリ管理部で実現している。

3.2 言語処理部

利用者とのインタフェースは、データベース言語「沙羅双樹Ⅱ」である。実行部の仮想機械の命令は、機械寄りであるため、データベース操作を高レベルで記述できる言語も必要である。

沙羅双樹Ⅱの特徴は、

- (1) オブジェクト指向言語
- (2) 完結した表現力
- (3) 宣言的問合せ

である。沙羅双樹Ⅱは、データ定義、データ操作、問合せが記述でき、利用者が要求するデータベース応用プログラムがすべて記述できる。

沙羅のデータモデルの特徴であるマルチクラスインスタンスを扱うためにこれまでの言語仕様[4]を見直し、拡張している。

拡張の一つであるマルチクラスインスタンスの作成は、沙羅双樹Ⅱで、

```
multiclass name1, name2
```

のように記述する。このように記述すると、一つのオブジェクトにインスタンス名name1とname2で示される表現を結び付けることができる。

### 3.3 実行部

実行部は、データベース操作のマクロな命令としてプログラムコードを実行する仮想機械である。実行部は、仮想機械命令に従い、メモリ管理部を利用してデータベースを操作する。メモリ管理部とのインタフェースは操作関数である。

この実行部は次の特徴をもつ。

- (1) スタックマシン
- (2) オブジェクト指向
- (3) インタープリタ

沙羅の実行部は、PostScriptの実行モデル[3]に、オブジェクト指向の機能を追加してデータベース向きに拡張している。拡張は、PostScriptのオペレータを追加することでなされている。PostScript言語のグラフィクス操作の機能はない。

PostScriptの実行モデルはForthを元にしたもので、小さい処理系で高速な実行速度が実現でき、拡張性に富むという特徴がある。

仮想機械方式は、データベース管理システムを機械独立にするために採用した。つまり、データベースに登録したメソッドは仮想機械の命令であり、そのメソッドを含めたデータベースを可搬にするためである。

実行部は、複数のプロセスをライトウェイトプロセスとして同時実行できる。このプロセスは、プロセスの状態を記述するデータの集合であるコンテキストで表現される。コンテキストは仮想機械の実行命令数とI/O待ちで切り替えられる。

実行部の基本命令の集合の選択は実行速度に関わる重要な問題であるが、最適な命令の集合はわかっていない。

### 3.4 メモリ管理部

メモリ管理部は、データベース内のオブジェクトの1次記憶と2次記憶の配置を管理し、実行部からの呼出しで、オブジェクトの生成、取出し、更新、削除を行う。トランザクション処理に関するオブジェクトの処理もこの部分の機能である。

実行部がオブジェクトの意味を扱うのに対し、メモリ管理部はオブジェクトをバイトの列として扱い、意味を取り扱わない。

メモリ管理部は、すべてのオブジェクトを生成するときに、オブジェクト識別子を発行する。以後、実行部はオブジェクト識別子でオブジェクトを参照する。このオブジェクト識別子は、利用者から見えるオブジェクトだけでなく、すべての管理部の内部オブジェクトにも生成している。

オブジェクトの領域は参照されなくなった時点で、回収している。

沙羅ではオブジェクト識別子をキーとした、インデクスとB-treeによる管理方法を採用した。インデクスは、オブジェクトの大きさ、存在位置、被参照数、クラス名などを内容にもつ構造体である。

沙羅では、インデクスとオブジェクトを別のファイルに分割して格納している。B-treeは、動的に生成している。

1次記憶と2次記憶の転送は、オブジェクトを一つの単位としている。転送するオブジェクトは、LRUアルゴリズムを使用して決定している。

LRUに使用する参照の情報は、クラスとオブジェクトのふたつの階層で記録している。情報は、まず、クラスの記録を使用して、もっとも最近使用していないクラスを選択し、そのクラスの中で、もっとも古く読み込まれたものを2次記憶に返している。

クラスに関する記録を優先しているのは、沙羅の検索はクラスを対象にしているためである。

### 4. むすび

オブジェクト指向データベースシステム「沙羅」では手続きを可搬な仮想機械命令としてデータベース内に格納することができ、これによって高度なデータ独立を実現している。このような沙羅、データベース言語、仮想機械、メモリ管理部の特徴となる実現手法について述べた。

仮想機械の命令は、まだ最終的なものではない。ベンチマークテストなどの実行により評価し、調整しなければならない。

### 参考文献

- [1]石丸他:データベース管理システム「沙羅」の設計と試作、信学研報 DE92-1, pp.1-7, 1992.5.11
- [2]T. Ishimaru et al: Sarah: An Object-Oriented Database System Capturing Multiple Meanings、アドバンスドデータベースシステムシンポジウム予稿集、IPSJ SIGDBS, 1992.12.8
- [3]Adobe: PostScript Language Reference Manual SECOND EDITION, Addison-Wesley, 1990
- [4]久保田他:オブジェクト指向データベース「沙羅」のデータベース言語、情処全大第44回 2H-9, 1992