

## 拡張可能DBMS MODUSのデータ管理機構

5H-2

石川 佳治\* 古瀬 一隆\* 于 旭\*  
 山口 和紀\*\* 北川 博之\*\* 大保 信夫\*\*  
 (\*筑波大学工学研究科 \*\*筑波大学電子情報工学系)

### 1. はじめに

今日では、CAD/CAM、CASE、AIなどのさまざまな応用分野においてデータベースの利用が進められている。しかし、応用分野によってデータベースに要求されるデータ型や機能は異なり、従来のデータベース管理システム(database management system: DBMS)では対応が不十分であった。

この問題に対するアプローチのひとつとして、拡張可能データベース管理システム(ExtensibleDBMS)の研究が行なわれている[1][2]。このアプローチでは、ユーザの使用目的に応じ、DBI(Database Implementer)が核となるDBMSをカスタマイズすることで応用目的に適したDBMSを作り出す。そのため、核となるDBMSはモジュール化が進められていて拡張性が高くなっている。

現在、筑波大学データベース研究室では、拡張可能データベース管理システムMODUSの研究を行なっている[3][4]。本稿ではMODUSのデータマネージャについて述べ、次いで、新たなデータ型およびアクセスメソッドの追加を支援するための機構について述べる。

### 2. MODUSのデータマネージャについて

MODUSのデータマネージャの機能は、主に以下の4つである。

- 1) スキーマ情報の管理
- 2) レコードへの高速なアクセスの支援
- 3) 一時ファイルの管理
- 4) 新たなデータ型とそれに伴う演算、アクセスメソッドの追加の支援

MODUSのデータマネージャの構成を図1に示す。データマネージャの構成要素はモジュール化され、互いに独立している。

カーネルインターフェイスは、問い合わせ処理系に対しオブジェクトの集合上での検索・更新・絞り込みなどの機能を提供する。必要に応じオブジェクトの集合を保持する一時ファイルが作られ、カーソルなどの抽象的な制御機構が使用できる。これらの機能は特定のデータモデルには依存しない比較的低レベルのものである。

スキーママネージャは、データ辞書から取り出した

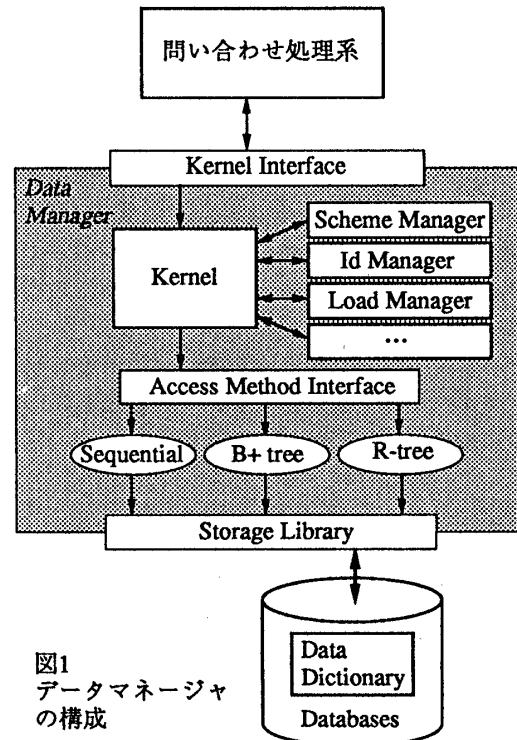


図1  
データマネージャ  
の構成

スキーマ情報をまとめて管理し、スキーマ情報への高速アクセスをはかる。ロードマネージャは、データ型やアクセスメソッドに付随するコードが必要とされたときに動的ローディングを行なう。Idマネージャは、データマネージャ内で管理される一時ファイルなどに識別子を与える働きを持つ。

アクセスメソッドインターフェイスは、個々のアクセスメソッドに依存しない抽象的なレコードアクセス手段を提供する。ここでは1つのレコードまたは複数のレコードに対する検索、更新などのオペレーションが用いられる。

ストレージライブラリは仮想記憶管理、LRU管理、バッファ管理のための基本ルーチンを含んでいて、アクセスメソッドの追加の際に利用できる。

カーネルは物理ファイルまたは一時ファイルにストアされている物理レコードへのアクセスを受け持つ。また、モジュール間の通信もつかさどる。

### 3. データ型の追加

新たなデータ型を追加可能とするため、MODUSではデータ型とそれに付随するオペレーションの情報をデータ辞書で一括して管理する。データ型の追加のためには、DBIはデータ型とそのオペレーションについて必要な手続きをコーディングし、関連する情報をデータ辞書へ追加する。データ辞書ではこのために、

- 1)Typeリレーション：タイプ番号とタイプ名の対応付け、関連するモジュール名の格納を行なう
- 2)TypeOperatorリレーション：タイプに伴うオペレータに関する情報を格納する
- 3)TypeLibリレーション：データマネージャ内部と外部の間のデータフォーマットの変換手続きの名前を格納する

などのリレーションを用いる(図2)。これにより、データマネージャ内部で新たなデータ型とそれに付随する演算を用いることができる。

### 4. アクセスメソッドの追加

新たに導入したデータ型を効率良く扱うためには、データ型や応用目的にあったアクセスメソッドが利用できることが必要である。特に最近ではR-tree[5]、Grid file[6]など、特定の対象を指向したインデックス機構が研究されていて、これらが追加可能であることが求められる。

アクセスメソッドインターフェイスが上位に提供するレコードアクセスのためのオペレーションは、個々のアクセスメソッドに依存しない抽象的なオペレーションである(open, close, search, insertなど)。このため、DBIによるアクセスメソッドのコーディングはアクセスメソッドインターフェイスの個々のオペレーションに関して、対応する手続きをコード化することになる。アクセスメソッドに関する情報を管理するためにデータ辞書が用いるのは次のリレーションである(図3)。

- 1)AccessMethodリレーション：アクセスメソッド番号とアクセスメソッド名の対応付け、モジュール名の格納などを行なう
- 2)AccessOperationリレーション：アクセスメソッドインターフェイスのオペレーション名と各アクセスメソッドのコード中の手続き名の対応づけ、およびモジュール名の格納を行なう

### 5. おわりに

本稿では、拡張可能DBMS MODUSのデータ管理機構について述べた。特に、新たなデータ型とそれに付随するオペレータおよびアクセスメソッドの

Typeリレーション

Id	Name	Module	Mode
#1	int	int.o	permanent
#2	RECT	RECT.o	temporal

TypeOperatorリレーション

Id	Parent	Name	unary_or_binary
#1	int	GE	binary
#2	int	LE	binary
#3	RECT	OVLP	binary

TypeLibリレーション

Id	Parent	Encode	Decode
#1	int	intE	intD
#2	RECT	RECTE	RECTD

図2 タイプ管理のためのリレーション

AccessMethodリレーション

Id	Name	Type[]	Module	Mode
#1	B+ tree	int, float, ...	Btree.o	permanent
#2	R-tree	RECT, CIRCLE	Rtree.o	temporal

AccessOperationリレーション

Id	Parent	Name	Module	Open	Search	...
#1	B+ tree	int	BopL.o	Bopen	Bsearch	...
#2	R- tree	RECT	RopR.o	Bclose	Rsearch	...

図3 アクセスメソッド管理のためのリレーション

DBMSへの追加を支援するための機構に焦点を当てた。

現在、UNIX上で稼働する商用の関係データベースシステムG-BASEをもとにして、シングルユーザに限定したMODUSのプロトタイプバージョンの詳細な仕様を検討している段階である。

### 6. 参考文献

- [1] M. J. Carey et al., "The Architecture of the EXODUS Extensible DBMS," in *Proc. of the Intl. Workshop on Object-Oriented Database Systems*, Sep., 1986.
- [2] D. S. Batroy et al., "GENESIS: An Extensible Database Management System," *IEEE Trans. on Software Engineering*, Vol.14, No.11, Nov. 1988.
- [3] 古瀬 et al., "拡張可能DBMS MODUSのアーキテクチャ," 情報処理学会第39回全国大会予稿集, Oct., 1989.
- [4] 陳 et al., "ルールベースに基づく最適化処理の設計," 情報処理学会第39回全国大会予稿集, Oct., 1989
- [5] A. Gutman, "R-trees: A Dynamic Index Structure for Spatial Searching," in *Proc. 1984 ACM-SIGMOD Conf.*, Boston, Mass.
- [6] J. Nievergelt et al., "The Grid File: An Adaptable, Symmetric Multikey File Structure," *ACM Transactions on Database Systems* Vol.9, No.1, March 1984.