

拡張可能 DBMS MODUS のアーキテクチャ

2M-3

古瀬 一隆† 于 旭† 山口 和紀†

北川 博之† 大保 信夫† 藤原 譲†

(†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

CAD/CAM、画像処理、知識処理など大量データを扱うさまざまな応用分野において、そのデータの管理のためにデータベースを利用する試みは、広く行なわれている。しかし、既存のデータベース管理システム (database management system, DBMS) ではこれらのデータを個々の応用分野に適した形で扱うことが必ずしもできないというのが現状である。この問題点を解決する策として、現在大きく分けて2種類の方法が研究されている。ひとつは、ADTなどの機能を付加することによってDBMSそのものの機能を強化しようとする方法 (complete DBMS) [3]であり、もうひとつはDBMSを構築する個々の構成要素のモジュラリティを高めることによってDBMS自体の拡張性を上げ、これをカスタマイズすることでそれぞれの応用分野に適応したDBMSを作り上げようとするもの (extensible DBMS) [1][2]である。

筑波大学データベース研究室では、これまでの研究[4][5]をもとに、後者の流れをくむ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。本稿ではMODUSの構成について述べ、このシステムによるDBMS構築のデータモデル実現の仕組みについて述べる。

2. MODUSの構成

図1にMODUSの構成を示す。本システムは、いくつかの構成要素の組合せによってDBMSの機能を構築している。従来のDBMSとは異なり、各構成要素の依存関係が明確になっているためそのモジュラリティは高く、これによって拡張性を実現している。このため、他に大きな影響を与えることなく各構成要素に手を加えることができ、比較的少ない能力で個々の応用分野に要求に添った(カスタムメイドの)DBMSを実現することが可能となっている。

DDLは、特定のデータ定義言語による記述を受け取り、各データ定義がどのようにDBMS内で扱われるかを示すマッピング情報を生成する。生成されたマッピング情報はデータタイプマネージャによって管理され、後に個々のDML文の処理を行なう際にDMLから参照、使用される。また、データ定義の内容はデータ辞書に登録される。

DMLはデータ操作言語の処理を行なう。オブティマイザジェネレータはデータ操作言語の最適化規則を受け

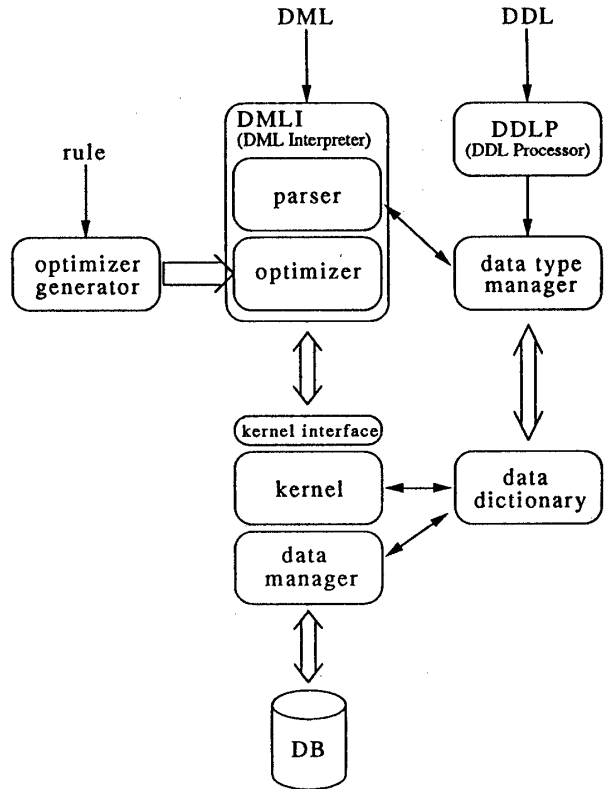


図1. MODUS のアーキテクチャ

取り、その規則に基づいたオブティマイザを生成する。生成されたオブティマイザはDMLIに組み込まれる。

カーネルは、データマネージャを通してデータベース上のデータの管理を司り、実際のデータ操作を行なう。MODUSではこのカーネル自体も拡張可能であり、新しいデータタイプ(本システムでは、カーネルの扱うデータタイプをドメインという)を導入したり、そのデータタイプに対するデータ操作(オペレータメソッド)を組み込むことができるようになっている。

カーネルインターフェイスはカーネルの機能をDMLIから使用するための環境を実現する。このインターフェイスを記述することによって、DMLIとカーネルとのやりとりが確立する。

The Architecture of the MODUS extensible DBMS
 Kazutaka Furuse †, Xu Yu †, Kazunori Yamaguchi †,
 Hiroyuki Kitagawa †, Nobuo Ohbo †, and Yuzuru Fujiwara †
 †Program in Engineering Sciences, Univ. of Tsukuba
 †Institute of Information Sciences and Electronics, Univ. of Tsukuba

3. データモデルの実現

MODUS による個々の (応用分野に適した) データモデルの実現方法について述べる。特定のデータモデルを実現するためには、そのデータモデルに即した DDL および DML を設計し、これらの言語を処理する DDLP、DMLI を作成すればよい。また、カーネルを拡張して、データタイプの導入やオペレータメソッドの追加を行なう必要がある場合もある。

(1). DDLP の動作

DDL P は、個々の DDL 文を受け取り、その文の内容に添ったマッピング情報を生成してそれをデータタイプマネージャに渡す。このマッピング情報には、DDL 文で定義されたデータが DBMS カーネル内でどのように扱われるかが記述されている。データタイプマネージャは、マッピング情報を受け取ると、それに即したデータ定義をデータ辞書に登録する。

(2). DMLI の動作

DMLI の動作の例を図 2. に示す。DMLI は、DML 文を受け取ると、まずデータタイプマネージャから必要なマッピング情報を取りだし、それをもとにデータ操作の解析木を作る。次にこれを最適化して最終的に DBMS カーネルに対するコマンド列を作り出す。これらのコマンドはカーネルインターフェイスによって提供される。したがって DML の実行の際にカーネル内で行なわれるデータ操作は、すべてカーネルインターフェイスの記述によって DML から命令することができるようになっていなければならない。

(3). 新しいドメインの導入

応用分野の要求を満足させるような DBMS を作り上げるためには、その分野に即したデータ表現やデータ操作を実現しなければならない。MODUS では、カーネルを拡張して新しいデータタイプ (ドメイン) を導入することによってこれを可能としている。導入されたドメインに対するデータ操作は、そのドメインに対するオペレータメソッドを定義することによって行なわれる。定義したオペレータメソッドを DMLI が使用するためには、そのためのカーネルインターフェイスを記述しなければならない (図 3.)。

4. おわりに

本稿では拡張可能 DBMS MODUS の構成について述べた。従来の DBMS では、データモデルは固定されており、そのデータモデルを操作するものとして DDL、DML が実現されていた。しかし、MODUS ではデータモデルを設計し、それにあわせて DDLP や DMLI を作り上げカーネルを拡張することによって、さまざまな応用分野に適応した DBMS を構築することが可能となっている。

現在、MODUS は開発途上にあり、カーネルインターフェイスの仕様や機能の詳細などについての検討が進められている。

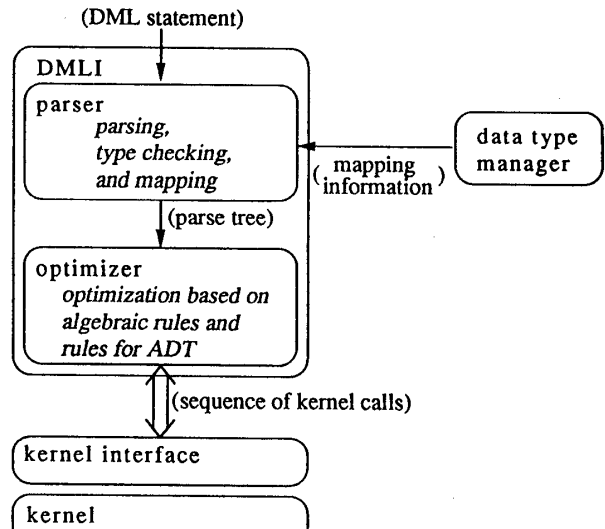


図 2. DMLI の動作

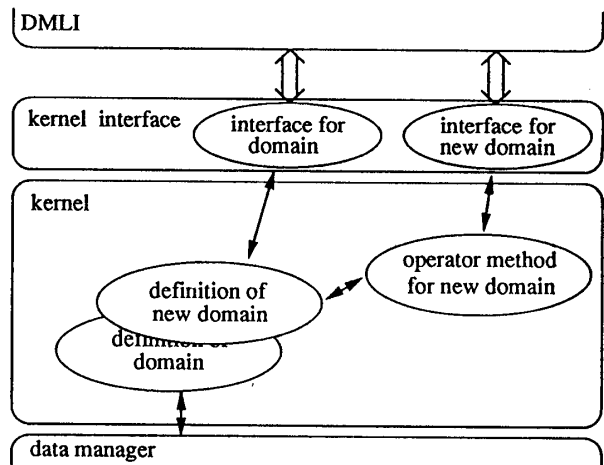


図 3. 新しいドメインの導入

5. 参考文献

- [1] D. S. Batory et al., "GENESIS: An Extensible Database Management System," *IEEE Trans. on Software Engineering*, Vol. 14, No. 11, Nov. 1988.
- [2] 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.
- [3] M. Stonebraker et al., "The Design of POSTGRES," in *Proc. of the 1986 SIGMOD Conf.*, May, 1986.
- [4] X. Yu, et al., "Integration of Functional Data Model into Programming Environment: Application to Solid Database," in *Proc. of the IFIP Conf. on Visual Database Systems*, Apr., 1989.
- [5] X. Yu, "The Designing of extensible DBMS," Database Lab. technical report, Univ. of Tsukuba, June, 1989.
- [6] 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.