5H−1 拡張可能データベース管理システム構築についての一考察

宝珍 輝尚
NTT情報通信処理研究所

1．はじめに

データベース管理システム(DBMS)がマルチメディアデータを扱おうとすると、その応用分野にDBMSが依存してしまうという問題が指摘され、解決アプローチの一つとして拡張可能なDBMSの研究が行われている[1]。

本稿では、現状のDBMSの拡張性に関する問題点について考察し、DBMSの核となるモジュール群を用いてDBMSを構築する方法について述べる。

2．DBMSの拡張性に関する問題点

2.1 現状のDBMSの問題点

応用パッケージから見た現在のリレーショナルDBMSの主な問題点は以下の通りである。

① データ型が限定されている。
　DBMS上の応用パッケージがアプリケーションプログラムとDBMS間のデータ型のギャップを埋めている。

② データに適用可能なオペレーションが限定されている。
　数値に対する大小比較，文字コードの大小比較，文字列のパターン比較のみである。

③ データへのアクセス方法が限定されている。
　CAD等の分野では、格納するデータによってはB木アクセスよりも優れたアクセス方法が研究されているがDBMSのアクセス方法として実現できない。

④ 応用分野へのDBの適用に時間がかかり、ad hocな応用システムの提供が困難である。

これらの問題点を解決するために、DBMSに高度の拡張性を持たせる研究が行われている[1]。

2.2 拡張可能DBMSの研究動向

拡張可能DBMS(Extensible DBMS)の研究では、主に以下のアプローチ方法がある。

[1] full-function DBMSの拡張アプローチ[2][3]
　DBMSとしての機能をすべて持ったDBMSに機能追加を可能とするアプローチであり、POSTGRES[2]，STARBURST[3]に代表される。

[2] DBMS generatorアプローチ[4][5]
　DBMSを構成するモジュールを組み合わせることによってカスタムDBMSを作り上げるアプローチで、GENESIS[4]，EXODUS[5]に代表される。

ここで、full-function DBMSの拡張アプローチには、拡張可能範囲が限定されてしまうという欠点がある。さらに、抽象化データ型としてユーザ定義のデータ型及び該データ型のオペレータをサポートしているが、エンドユーザに必要であるか疑問である。一方、DBMS generatorアプローチでは、DBインプリメンタ(DBI)の負担が膨大であり、負荷軽減を考慮する必要がある。

2.3 現状のDBMSの拡張性に関する問題点

著者は、現状のDBMSの拡張性に関する主な問題点を以下のように考えている。

【問題点1】DBMSのインプリメントにおいて、データ独立の考え方が導入されていない
・データおよびモジュール間のインタフェースが固定的であり、インタフェースの変更によるモジュール修正コストが高く、拡張性を阻んでいる。

【問題点2】DBMSの処理に使用するデータおよびモジュールの操作方法が統一的でない。
・データおよびモジュールの操作方法の違いがデータおよびモジュールの利用方法の理解を困難にしており、拡張性を阻んでいる。

そこで次章では、以上の問題点を解決し、拡張可能DBMSを実現する方法について述べる。

3．拡張可能DBMSの実現方法

本稿では DBMSgeneratorアプローチを採る。また、DBIの存在を仮定し、エンドユーザではなくDBIに拡張可能性を提供することを考える。

3.1 基本構想

前章で述べた問題を、DBMSの処理に使用するデータおよびプログラムのデータベース化で解決する。本データベース化により、スキーマで定義された論理的なデータ構造を介してデータ等を操作でき、インタフェースの柔軟化を図れる。また、データベース化によりデータ等の操作を統一化できる。

3.2 DBMSコアモジュール群

前節で述べたデータベース化を実現するためにDBMSコアモジュール群を導入する。DBMS構築に必要なDBMSコアモジュール群の例を表1に示す。

DBMSコアモジュールは、各々特定の種類のデータに対する管理及び処理を行う。また、DBMSコアモジュール群では、①データ検索(SELECT)，②データ格納(INSERT)，③データ更新(UPDATE)，④データ削除(DELETE)，⑤プログラム実行(EXECUTE)という統一的なデータ操作方法を設ける。

DBMSコアモジュールの管理するデータの例を図1に示す。図1には、プラン管理モジュールの管理するデータ

A Study on the Implementation methods of Extensible DBMS
Teruhisa HOUCHIN
NTT Communications and Information Processing Laboratories

のメタスキーマの例をＩＦＯモデルの表現で示した。図１ではプラン木（データベース処理の実行プランを表現する木）をノードとエッジで管理しており、ノードではノード自身の持つ情報，ノードを構成する構造情報およびノードに対応する処理を実行するプログラムが管理されている。このようなデータ管理を行うことにより、論理的なデータ構造を通してＤＢＭＳコアモジュールの管理するデータにアクセス可能である。

3.3 ＤＢＭＳの構築方法

ＤＢＭＳはＤＢＭＳコアモジュール群とそれらを使用するモジュール（ＤＢＭＳ実現モジュール）から構成される。ＤＢＭＳコアモジュールを用いたＤＢＭＳ実現モジュールの作成例として、プラン実行モジュールの概要を図２に示す。図２に示すように、ＤＢＭＳ実現モジュールではプラン実行の制御とＤＢＭＳコアモジュールの管理するデータの操作命令を記述する。また、ＤＢＭＳを拡張するには、ＤＢＭＳコアモジュールへのデータ追加およびＤＢＭＳ実現モジュールの追加または修正を行うことになる。

4．おわりに

本稿では、ＤＢＭＳに拡張性を持たせるために、ＤＢＭＳの核となるモジュール群を用いてＤＢＭＳを構築する方法を提案した。

今後の課題は、本稿で述べた考え方をもとにしたＤＢＭＳコアモジュール群のプロトタイプの作成，ＤＢＭＳの拡張性の評価方法等についての検討である。

参考文献

(1) D.S.Batory and M.Mannino,"Panel on Extensible Database Systems", Proc. ACM SIGMOD 1986.
(2) M.R.Stonebraker and L.A.Rowe, "The design of POSTGRES", Proc. ACM SIGMOD 1986.
(3) B.Lindsay et al,"A Datamanagement Extension Architecture", Proc. ACM SIGMOD 1987.
(4) D.S.Batory et al,"Implementation Concepts for an Extensible Data Model and Data Language", ACM TODS, vol.13, No.3, 1988.
(5) M.J.Carey et al,"A Data Model and Query Language for EXODUS", Proc. ACM SIGMOD 1988.

表１　ＤＢＭＳコアモジュール群の例

No	ﾓｼﾞｭｰﾙ名	機能概要
1	字句管理	ﾃﾞｰﾀﾍﾞｰｽ言語の字句の管理
2	構文木管理	言語処理に使用する構文木の管理
3	ｽｷｰﾏ管理	ﾃﾞｰﾀ定義情報の管理
4	ﾃﾞｰﾀ型管理	ﾃﾞｰﾀ型に関する情報の管理
5	ｱｸｾｽｺｽﾄ管理	ﾃﾞｰﾀｱｸｾｽｺｽﾄ等の管理
6	ﾌﾟﾗﾝ管理	実行ﾌﾟﾗﾝ,ﾌﾟﾗﾝ木の管理
7	ﾃﾞｰﾀ管理	ﾃﾞｰﾀへのｱｸｾｽ処理
8	資源管理	ﾃﾞｰﾀﾍﾞｰｽ資源の管理
9	障害管理	障害関係の制御
10	ﾌﾟﾛｸﾞﾗﾑ管理	ﾌﾟﾛｸﾞﾗﾑの管理
11	ﾌｧｲﾙ/ﾒﾓﾘ管理	ﾌｧｲﾙ,ﾒﾓﾘの管理

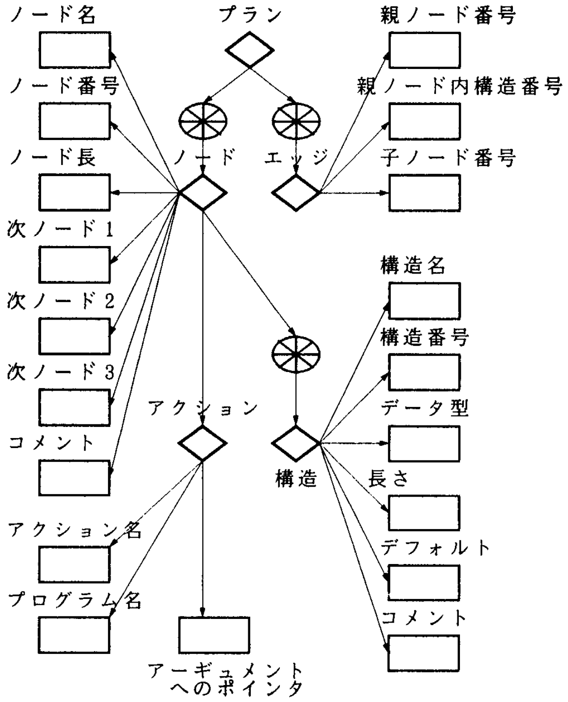

図１　プラン管理モジュールで管理するメタスキーマ例

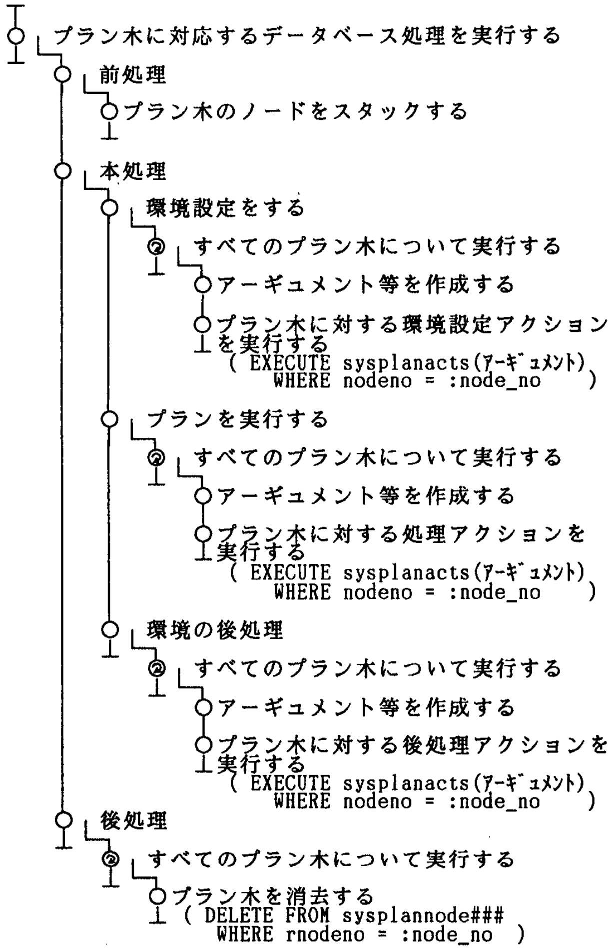

図２　プラン実行モジュールの概要