

3M-4

知識ベース管理システムKappa
— 試作システムの内部モデル —

小澤 守* 金枝上 教史** 根本 仁* 川村 達* 杉崎 元*

* 三菱電機東部コンピュータシステム部

** 勸新世代コンピュータ技術開発機構

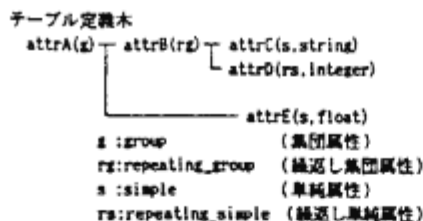
1.はじめに

知識ベース管理システムKappaの試作システムでは最下位層のデータ・モデルを内部モデルと呼んでいる。このデータ・モデルは種々の知識情報処理システムの処理効率を向上させるために非正規形モデルにもとづいている。本発表では試作システムで実現した機能と非正規形データの扱いについて述べる。

2.内部モデルのデータ構造

木構造や階層構造を持つレコード構造をどの様に表現すべきかと言う問題についてはいくつかの提案がなされている。Kappaでは木構造で表現される非正規形テーブルのレコード構造と属性値を格納するために、各属性値をdepth_firstの順で並べて1つのレコードを作っている。

レコードは複数個を並べて1つの物理ページに格納している。属性値とレコードの長さは固定ではないので、長さの情報を各属性値と各レコードの先頭に付けている。内部モデルにおけるレコードの格納例を図2.1と図2.2及び図2.3に示す。



レコードの値

attrA(s)			attrE(s)
attrB(rg)		attrC(s)	
attrD(s)			attrE(s)
"x"	0 1	0.9999	
"y"	2 3 4		

図2.1 テーブル定義木とレコードの値

レコードの木表現

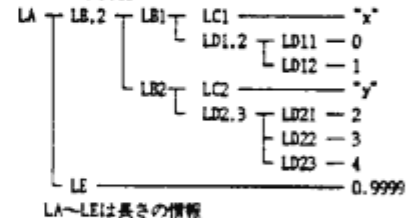


図2.2 レコードの木表現

物理レコードの形式

LA-LB-2-LB1-LC1-["x"]-LD1-2-LD11-[0]-LD12-[1]-LB2-LC2-["y"]-LD2-3-LD21-[2]-LD22-[3]-LD23-[4]-LE-[0.9999]
 () 内は属性値
 LA~LEは長さの情報

物理ページの形式

1バイト 4096バイト

Lblk	Lrec1	RECORD1	Lrec2	RECORD2	...
------	-------	---------	-------	---------	-----

Lblkは1ページ内の有効ブロック長
 Lrecn (Ln) 番目のレコード長
 RECORDn (Ln) 番目の物理レコード

図2.3 レコード及びページの物理形式

3.内部モデルのデータ操作とコマンド形式

Kappaでは内部モデルのデータを操作するコマンドとして原始コマンドを用意している。関係代数を使わずにコマンド形式とした理由は拡張性やチューニングなどを考慮したためである。

このコマンドは読み系、検索系、更新系の3つに大別されている。ある条件に合うレコードを捜すのであれば検索系のコマンドを使えば良い。検索後の結果としてRID (レコード番号) とオカレンス (繰返し属性におけるある属性値) 番号の集合が求まる。求まった集合が何件であるかの情報も得られるので、検索条件を変更し絞り込みを行なうことも容易である。

この集合からRIDを取り出して読み系コマンドを使えば該当レコードの値を得ることができる。この場合レコード全体を読むのか、或はある一部分の属性だけを読むのかの

Knowledge Base Management System Kappa

- Internal Model of Prototype System -

Mamoru OZAWA* Atsushi KANAEGAMI** Hitoshi NEMOTO* Toru KAWAMURA* Gen SUGIZAKI*

* Mitsubishi Electric Computer Systems(Tokyo) Corporation

** Institute For New Generation Computer Technology

指定も行なえる。レコードの値を確認して変更や削除が必要であるならば、読んだ直後に更新系のコマンドを発行すれば良い。主なコマンドの形式と機能を以下に示す。

3.1 読み系

読み系には次の3つの種類がある。

①レコード読み

実際の属性値を指定したフォーマットに従い構造を持った形で読む。

②インデックス読み

指定した属性の値により昇順・降順などの順序で属性値や件数及びレコードのRID（レコード番号）などを返す。

③定義読み

原始テーブルにおける、属性名、タイプなどの定義情報を返す。

読みを行なう場合、現在読んでいる位置情報を示すものとしてタップ（Tap）と呼ぶオブジェクトを使っている。このタップの中には読みの順序（昇順、降順、物理順、論理順、RID 順）や読みの形式（レコード全体、特定の属性だけ）なども指定されている。読み系の原始コマンドの形式を次に示す。

(1) タップの作成

```
create_tap(Kappa, Table_id, Format, Position, ^Tap)
Table_id ::= 原始テーブルID
Format ::= レコードの読み込み形式
Position ::= レコード読みの場合は物理順、RID
            順などを指定する。インデックス読みの場合は昇
            順、降順などを指定する。定義読みの場合は何の
            定義を読むかを指定する。
Tap ::= 作られたTap のIDが返される。
```

(2) レコード読み

```
read_record(Kappa, Tap, Format, ^Record)
Tap ::= 使用するタップのIDを指定する。
Format ::= {}の時はTap 内のFormatが使われる。
            そうでなければここで指定した形式で読み込む。
Record ::= レコード値を構造を持った形で返す。
```

(3) インデックス読み

```
read_index(Kappa, Tap, ^Record)
Tap ::= 使用するタップのIDを指定する。
Record ::= [値]、[値、件数]又は[値、件
            数、RID リスト]が返される。
get_cardinality(Kappa, Table_id, Attr_name, ^Cdn)
Table_id ::= 原始テーブルID
Attr_name ::= 属性名
Cdn(Cardinality) ::= 指定した属性が持っている
                    値の種類が返される。
```

3.2 検索系

検索の種類には検索の対象となる属性がインデックスとになっているインデックス検索とそうでないデータ検索の2

種類がある。また検索の結果得られた集合にたいして集合間の演算を行なうコマンドも用意されている。試作システムではインデックス検索の機能を実現し評価している。検索系の原始コマンドの形式を次に示す。

(1) 検索

```
search(Kappa, Table_id, Conds, ^Set_name, ^Quantity)
Table_id ::= 原始テーブルID
Conds ::= 検索の条件(conditions)を指定す
            る。値の範囲指定、前方一致、後方一致及び検索
            条件の論理積、論理和、論理差などの指定を行
            う。
Set_name ::= 検索の結果作成される集合名。
Quantity ::= 検索の条件にあうレコードの件数。
```

(2) 集合演算

```
set_op(Kappa, Set_op, ^Set_name, ^Quantity)
Set_op ::= 集合間の演算式を指定する。
            (演算子は and, or, difference である。)
Set_name ::= 演算の結果得られる新たな集合名。
Quantity ::= 演算結果のレコード件数。
```

3.3 更新系

更新系には内部テーブルの使用を宣言するオープン・クローズとトランザクションの制御及び追加・更新・削除などのコマンドがある。更新系のコマンドの形式を次に示す。

(1) 使用宣言と制御

```
open(Kappa, Table_list)
close(Kappa)
begin_transaction(Kappa, Table_list, Ename, ^Edata)
end_transaction(Kappa, Ename, Edata)
```

(2) 更新系

```
add(Kappa, Tap, Value_list)
replace(Kappa, Table_id, Tap, Format, Value_list)
Value_list ::= 使用するタップのフォーマットにあ
            う形の更新属性値のリスト
delete(Kappa, Tap)
```

4. おわりに

内部モデルは従来のデータベース技法にもとづいて試作した。全体システムでは機能を整備し、更に効率の向上を目指す為に主記憶の利用[2]を検討している。

参考文献

- (1) 根本、金枝上、宮地、篠野、合田“知識ベース管理システムKappa —試作システム基本のモデル”情報処理学会第35回全国大会、3M-5、1987
- (2) 金枝上、横田、河村、横塚“知識ベース管理システムKappa —大規模主記憶を利用した全体システム”情報処理学会第35回全国大会、3M-9、1987