

メタデータを用いたOLAP操作

柏川伸悟 塩谷勇 三浦孝夫

法政大学工学研究科電気工学専攻
産能大学経営情報学部

概要

データの分析による意思決定のための操作方法として、OLAP操作がある。OLAP操作であるロールアップは、具体的にデータベース操作を用いて定義できる[2]。ドリルダウンは具体的な操作体系の枠組みは存在せず、実体化されたビューの選択を意味する管理操作として扱われている。本稿では、ロールアップ、ドリルダウン操作をメタオブジェクトの機構と関係代数操作を用いて定義する。この基礎には、ロールアップ、ドリルダウン操作の操作対象となるビューは、メタオブジェクトの枠組みを有効に利用する。

OLAP Operation Using Meta Data

Shingo Kashikawa Isamu Shioya Takao Miura

Grad. of Engr., Hosei University
Dept. of Management and Informatics, SANNO College

Abstract

The OLAP operation is known as the operation method for decision support by data analysis. The drill-down that is one of OLAP operation practically can be defined by using database operation[2]. But, the drill-down operation cannot be defined by using database operation. It treats as management operation that means selection of the materialized view.

In this paper, we define the rollup and drill-down by mechanism of meta object, and relational algebra operation. The fundamental way effectively uses the framework of a meta object for the views as the operation object of rollup and drill-down.

1 まえがき

データウェアハウスを用いた意思決定のための操作を、OLAP(Online Analytical Processing)操作という[1]。OLAP操作の中でも、ロールアップ、ドリルダウン操作は特徴的である。ロールアップ、ドリルダウン操作では、求める視点階層の変化を生じる。抽象レベルの上昇はロールアップ操作に対応し、抽象レベルの下降はドリルダウン操作に対応する。SQL言語によるロールアップ操作は、グループ操作により実現が可能であることが知られている[2]。具体的には、階層情報を保持する関係と結合操作し、求めるデータのグループ化、集約関数の適用、そして射影操作を組み合わせて実現できる。

一方で、ドリルダウン操作は具体的な操作体系の枠組みは存在しない。実体化されたビューの選択を意味する管理操作として扱われている。ロールアップ操作によるビューでは、抽象度が上昇し情報を消失してしまい、抽象度を下げるドリルダウン操作を、実際の操作体系で記述することができない。つまり、操作体系に基づくOLAP操作はベースとなる視点が一定で抽象化だけとなる一方向のみの操作しかできない。

これらの問題点を解決するためには、操作結果となるビューを保証する枠組みが必要である。保証する枠組みとは、データウェアハウスを構成するベースとなる関係のデータを、ビューが保持することである。この保証により操作対象となるビューが自由になり、抽象度を下げる操作ができる。

具体的な解決方法として、すでに提案しているHOME(Harmonized Objects and Meta-objects Environment)システムを用いて[4][5]、メタオブジェクトの性質をOLAP操作に対応させることで可能である。ベースとなる関係に対して、結合操作と、すでに提案している関係係数によるグループ操作により、メタオブジェクト生成をする。この操作の目的は、視点となるビューを定めることである。本稿では、この操作をステージアップ操作と呼ぶ。この得られたビューを用いて、OLAP操作の具体的な操作体系を定義することが可能となる。ロールアップ、ドリルダウン操作は、ステージアップ操作によるビューを用いて、メタオブジェクトの操作や、集約関数適用などの具体的なデータベース操作を用いて定義することができる。具体的には、例えばロールアップ操作であれば、メタオブジェクトの集約関数適用と射影操作を用いて定義する。

はじめに第2章でメタオブジェクトとメタ質問、第3章でメタオブジェクト操作によるOLAP操作の特徴、第4章でグループ操作と集約演算、第5章でメタデータ操作によるOLAP操作、第6章でむすびとする。

2 メタオブジェクトとその操作

2.1 メタオブジェクト

オブジェクトとは、データベース中の興味ある情報を含むデータである。オブジェクトは、データベースの設計時に型を定義され分類される。メタオブジェクトとは、オブジェクト間の特性や情報を保持し、データのデータとも呼ばれる。一般的な質問であるオブジェクト質問には、メタオブジェクトの参照が必要で、メタオブジェクトは必要不可欠である。具体的には、メタオブジェクトとは関係や属性などがメタオブジェクトにあたる。質問による結果は、仮想的に結果スキーマ(REsULT)を用いて記述され、一時的なスキーマの管理をする。

メタ質問は、メタオブジェクトを含んだ質問で、関係係数で表現される。メタ質問は、メタオブジェクト、およびそれが表す外延情報を探索対象とする点で、高階の抽象度を有している。

オブジェクトとメタオブジェクトとの両方でシームレスな評価のために、メタオブジェクト m に対して $\$m$ を意味する具体化(Defication)機構を導入する[5]。逆に、オブジェクト $\{o_1, \dots, o_n\}$ に対して、 $\$o = \{o_1, \dots, o_n\}$ である $\{o_1, \dots, o_n\}$ の抽象化(Reification)を意味するメ

タオブジェクトをを導入する。 $\{o_1, \dots, o_n\}$ はメタオブジェクトを意味する。新しく生成された抽象化メタオブジェクトを、 $new\{o_1, \dots, o_n\}$ と表現する。

例1 関係CompanyAで'Fab1, 井上'を評価値にもつ、地区3と製品の評価を射影せよ。

CompanyA			
都道府県	期日	売り上げ	製品
東京	10	40	靴下
千葉	10	30	靴下
東京	11	40	鉛筆
宮崎	10	5	靴下

鉛筆		靴下	
工場	責任者	工場	責任者
Fab1	井上	FabA	林
Fab2	岡田	FabB	高橋

次の質問により求めることができる。

```
Project[都道府県,$製品]
Select['Fab1, 井上' ∈ $製品](CompanyA)
```

(RESULT)

都道府県	製品	
東京	Fab1	井上
東京	Fab2	岡田

製品の各タプルは、メタオブジェクト鉛筆と靴下である。選択条件は \in というオペレータにより、靴下と鉛筆の評価結果と'Fab1, 井上'の包含関係で判別する。状態判別は、靴下に含まれるため答えが得られる。□

2.2 スキーマ管理をするコアセット

メタオブジェクトの完全な意味を記述している本システムのスキーマについて述べる。目的は、我々のスキーマ構造の中だけでスキーマを定義することである。つまり、もし特別なスキーマについて扱うと定義すれば、この構造で全てのスキーマを定義できるということである。我々のスキーマは、これを実現するのに十分であり、これをスキーマ構造のコアセット(core set)と呼ぶ。

スキーマは、3個のリレーションから形成される。

- RelationCatalog
- AttributeCatalog
- DomainCatalog

これで全ての情報を矛盾なく管理できる。
初期状態で RelationCatalog は 3 つのスキーマ関係に相当する 3 つのタプルから構成される。また, RelName, Arity, Width という 3 個の属性を持ち、それらのドメインは name, count である。name ドメインの値がを 32 バイト、count ドメインを 4 バイトの整数型に仮定すると、そのとき Relcat は 40 バイトになる。同様に DomainCatalog 関係も定義できる。

RelationCatalog		
RelName	Arity	Width
RelationCatalog	3	40
AttributeCatalog	5	104
DomainCatalog	3	40

DomainCatalog		
DomName	ValueType	ValueSize
name	CHARACTER	32
count	INTEGER	4
type	TYPE	4

AttributeCatalog は、3 つのスキーマ関係の全ての属性数にあたる 11 のタプルから構成される。

AttributeCatalog				
RelName	AttrName	DomName	Offset	Position
RelationCatalog	RelName	name	0	1
RelationCatalog	Arity	count	32	2
RelationCatalog	Width	count	36	3
AttributeCatalog	RelName	name	0	1
AttributeCatalog	AttrName	name	32	2
AttributeCatalog	DomName	name	64	3
AttributeCatalog	Offset	count	96	4
AttributeCatalog	Position	count	100	5
DomainCatalog	DomName	name	0	1
DomainCatalog	ValueType	type	32	2
DomainCatalog	ValueSize	count	36	3

DomainCatalog において型を表している値 CHARACTER, INTEGER, TYPE は、それぞれドメインとして認められている値の種類を記述している。それぞれ文字列型、整数型、整数型を意味する。

さらに、ユーザーの関係スキーマを保持することができる。例えば、例 1 の関係 CompanyA は RelationCatalog に次のように保持されている。

(CompanyA, 4, 72)

これは、関係 CompanyA はサイズ 96 で 4 つの属性が含まれていることを意味する。関係 Friends のそれぞれの属性は、AttributeCatalog に次のように保持されている。

(CompanyA, 都道府県, name, 0, 1)
 (CompanyA, 期日, count, 32, 2)
 (CompanyA, 売り上げ, count, 36, 3)
 (CompanyA, 製品, name, 40, 4)

(CompanyA, 期日, name, 32, 2) は、関係 CompanyA に属性として期日があり、それぞれのタプルで 32byte 目から始まり、二つ目の属性ということの意味する。さらに、この属性の領域は count である。この領域名は、DomainCatalog で定義される。

3 メタオブジェクト操作による OLAP 操作

オブジェクトに対して、メタオブジェクトは抽象レベルが一階高く、メタオブジェクトのメタオブジェクトはさらに抽象レベルが増していく。そのため、あるメタオブジェクトに視点をおいた場合、抽象レベルの高いメタオブジェクト、低いメタオブジェクトと分けることができ、これらは全体として抽象化階層を形成する。

ロールアップ、ドリルダウン操作は、ある軸を基準にして階層の上げ下げをする操作である。本稿では、メタオブジェクトの抽象化階層と、ロールアップ、ドリルダウン操作による階層の変化操作とを対応させる。

図 1 に従来の操作による OLAP 操作と、メタオブジェクト操作による OLAP 操作との違いを例 1 の関係 CompanyA の都道府県を軸にした抽象化階層の変化による図で示す。

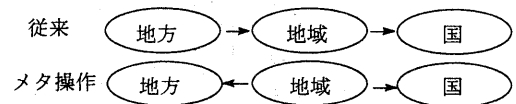


図 1 抽象化階層の変化

データベース操作によるロールアップ操作は、図 1 に示したように、上位レベルの地域、そのビューを用いて国というように上位レベルへ向かう一方の操作しかできない。メタオブジェクト操作による OLAP 操作は異なり、ビューの中立化が可能になり、現在の階層のビューが地域であると仮定すると下位レベルの都道府県を求めることが可能である。

4 グループ操作と集約関数

この章では、すでに提案したグループ操作 Group によるメタオブジェクトの生成と、集約関数操作 Apply について定義する [4]。

4.1 グループ操作

Group は、動的にタブルを集めメタオブジェクトを生成する。具体化によってメタオブジェクトが集合に相当するのと同じように、タブルの集合のグループ操作によってメタオブジェクトを動的に生成する抽象化 (Reification) の一種である。

```
group [ A ] RelExpr
      A:Attributes
```

RelExpr がスキーマ S を持つ。RelExpr のすべてのタブルは、 A の値 v_1, \dots, v_k を基にしてグループに分けられ、 $(S - A)$ も R_1, \dots, R_k で分割される。

結果の関係のスキーマは、 A と ∇_g となる。 ∇_g は、 $\{t_1, \dots, t_k\}$ の抽象化で生成されるメタオブジェクトである。ただし、 ∇_g は $\text{new } \{t_1, \dots, t_k\}$ として考え、 ∇_g は 1 属性 ∇_g からなる関係スキーマとしてみなす。さらに、 ∇_g で $\nabla_g(S - A)$ を意味する。

例 2 例 1 の関係 CompanyA に対して、地区毎にグループ分けせよ。

```
group [都道府県] (CompanyA)
```

この質問により、"期日, 売り上げ, 製品" で定義される 4 つのメタオブジェクト t_1, \dots, t_4 が生成される。

(Grouping)

都道府県	∇_g
東京	t_1
千葉	t_2
宮崎	t_3

t_1	期日	売り上げ	製品
	10	40	靴下
	11	40	鉛筆
t_2	期日	売り上げ	製品
	10	30	靴下
t_3	期日	売り上げ	製品
	10	5	靴下

□

4.2 集約関数

Apply は、メタオブジェクトを入力とし、それらを評価することで行う。Apply は次のように定義される。

```
apply[ Attribute:= FuncExpr ] RelExpr
```

RelExpr は、スキーマ S を持つ。RelExpr のそれぞれのタブルに対して FuncExpr を当てはめ、新しい属性の値として結果を得る。結果の関係スキーマは、 S と新しい属性からなる。

FuncExpr は、二つもしくは三つからのパラメータから形成される。

```
func( F, A, C )
```

入力として 1 つのタブルをとる。 $F(\subseteq S)$ は、func で処理される属性を意味する。集約関数の場合、 F の値は、いくつかのスキーマ T で定義されるタブルの集合に一致するメタオブジェクトを含んでいる。 $A \subseteq T$ であり、 A の値が求められる。グループ操作のあとでは、func は ∇_g 上のメタオブジェクトをとり、集約値を計算する。パラメータ C は、context と呼びどのように関数を適用するかを意味する。 $\nabla_g(T)$ は、暗黙にスキーマ情報 T を保持している。 $\nabla_g[C]$ ならば、 C 上で $\$t_i$ が射影される。つまり、 C はスコープとなる。

例 3 例 3 の CompanyA を基に、地区毎に売り上げの合計を求めよ。

グループ操作結果を用いて、次のようにして答えを求めることができる。

```
project [都道府県, 収入]
apply [収入:= SUM( $\nabla_g$ , "売り上げ")]
group[都道府県] (CompanyA)
```

このときの結果となる関係は次のようになる。

(RESULT)

都道府県	収入
東京	80
千葉	30
宮崎	5

同様に CompanyA で売り上げの合計を求めたい場合は、次のように記述する。

```
project [Total]
apply [Total:= SUM("CompanyA", "売り上げ") ] ({} )
```

(RESULT)

Total
115

□

5 メタデータを用いたOLAP操作

本章では、ロールアップ、ドリルダウン操作を矛盾無く定義するため、メタオブジェクトによる抽象階層管理を仮定した代数操作で意味を与える。

5.1 ステージアップ操作

ロールアップ、ドリルダウン操作の操作対象となるデータを生成する操作として、ステージアップ操作を導入する。ステージアップは、ベースとなる関係と抽象化を用いて新たなメタオブジェクトを生成する。実際には、結合操作と抽象化操作の一種であるメタオブジェクトを生成するグループ化操作で定義する。生成されたメタオブジェクトにより、OLAP操作に対応する抽象化階層を形成する。ステージアップ操作は、次の関係代数操作で定義される。

この操作を一つのオペレータ $stageup(W)$ (R, S) として定義すると次のように定義できる。

$$group[W] \text{ join}(R, S)$$

R は抽象階層構造を示す関係で、 S はベースとなる下位レベルの関係と仮定する。ただし、 R のスキーマを R とし、 S に含まれている属性を S とすると $S \in R$ でなければならない。つまり、属性 S は抽象階層の軸となる属性である。結合操作で、関係 R により S の各タプル $\{s_1, \dots, s_n\}$ に対して上位レベルにあたる値が付加される。この付加された値をそれぞれ $\{r_1, \dots, r_n\}$ とすると、抽象レベルの関係は $r_1 > s_1, \dots, r_n > s_n$ である。 W は、3つからなるグループ化属性を意味する。3つのグループ化属性とは、視点となる求める階層レベルの軸の属性と他の異なる軸の2つの属性である。例えば、次に示した図2であれば地域が求める軸であれば他の異なる軸は製品と時間にあたる。

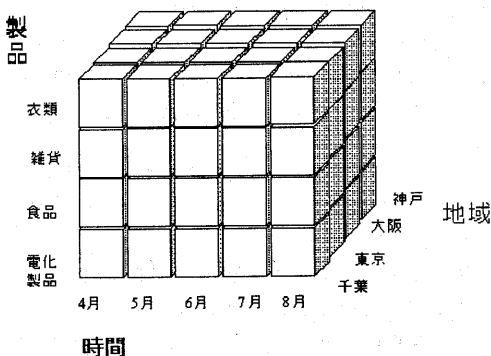


図2 キューブ

例4 例1の関係 $CompanyA$ と $Hierarchy$ から、地区の上位レベルである地域にステージアップする例を示す。

$stageup[地域, 製品, 期日](CompanyA, Hierarchy)$

$CompanyA$ と $Hierarchy$ から結合操作を行い、地区に対して2レベル抽象度の高い上位レベルの地域と対応づける。そして、抽象度の軸となる属性の地域と他の軸の属性の製品、期日でグループ化される。ここで、後の例題で用いるため操作結果を地域別という関係とする。

Hierarchy

都道府県	地域
東京	関東
千葉	関東
宮崎	九州

(Join)

地域	都道府県	期日	売り上げ	製品
関東	東京	10	40	靴下
関東	千葉	10	30	靴下
関東	東京	11	40	鉛筆
九州	宮崎	10	5	靴下

地域別

地域	製品	期日	∇_g
関東	靴下	10	t_α
関東	鉛筆	11	t_β
九州	靴下	10	t_γ

t_α	都道府県	売り上げ
	東京	40
	千葉	30

t_β	都道府県	売り上げ
	東京	40

t_γ	都道府県	売り上げ
	宮崎	5

□

5.2 ロールアップ操作

ロールアップ操作は、集約関数適用と射影操作がなされる。ロールアップ操作 $rollup[\alpha, \beta, \mathcal{F}] RelExpr$ を、次の関係代数操作で定義する。

$project[\alpha, \beta] apply[\beta := FuncExpr] RelExpr$

FuncExpr で適用される集約関数 \mathcal{F} は、適用される集約関数の関数を意味し COUNT, MAX, MIN, SUM, AVG である。集約関数へ入力は、ステージアップ操作によって生じたメタオブジェクトでありその評価されたメタオブジェクトの数値を計算する。 β は、階層のレベルが上がったために、視点となる階層にメタオブジェクトを適用させた集約関数の集約値を含む属性を意味する。つまり、Stateup のスキーマを U とすると、 $\alpha \subseteq U$ でなければならない。 α は、apply に適用される関係のスキーマ ∇_g を除いた全ての属性を意味する。

例 5 地域を視点として、期日、製品毎の売り上げを求めよ。

例 4 のステージアップ結果であるビューの地域別を用いて、次の操作でロールアップ操作を行う。

rollup["地域, 製品, 期日", 地域売り上げ, SUM] (地域別)

(RESULT)

地域	製品	期日	地域売り上げ
関東	靴下	10	70
関東	鉛筆	11	40
九州	靴下	10	5

集約関数 SUM により、例 4 のメタオブジェクト $t_\alpha, t_\beta, t_\gamma$ を入力として、地域階層のレベルにあわせた売り上げが算出されて地域売り上げの値が求められる。
□

5.3 ドリルダウン操作

rollup と同様に、ステージアップで得られたビューのメタオブジェクトを評価することで、実際の操作としてドリルダウン操作 $\text{drilldown}[P, Q]$ RelExpr を定義できる。実際には、メタオブジェクトの評価である具体化操作を含む射影操作により行うことができ次の式で定義する。

project[P, \$Q] RelExpr

Q は、メタオブジェクトを値とする属性を意味する。RelExpr のスキーマを U であるとする、 $Q \in U$ でなければならない。 P は、階層レベルを意味する軸の属性ではなく、他の異なる軸の属性を意味する。drilldown の射影属性は、軸の属性と、メタオブジェクトを評価した属性である。

例 6 視点となる地域から、下位レベルの都道府県のレベルへドリルダウンせよ。

例 5 と同様に、ステージアップ結果である例 4 のビューの地域別から、メタオブジェクトを評価しドリルダウン操作を行う。

drilldown["製品, 期日", ∇_g] (地域別)

地域別のメタオブジェクトを値とする属性 ∇_g が、 Q にあたる。 P は、メタオブジェクトの属性に含まれていない軸の属性の製品、期日である。

(RESULT)

製品	期日	∇_g [都道府県, 売上]
靴下	10	東京 40
靴下	10	千葉 30
鉛筆	11	東京 40
靴下	10	宮崎 5

□

6 ステージアップとドリルダウン

本稿では、ドリルダウンとステージアップ操作は、互いに逆操作である。

ステージアップ操作をする前のスキーマを $S(A_n, B, C, D)$ とし、タプルを t_1, t_2, \dots, t_n とする。 A_n は視点となる軸の属性で、 B と C は他の軸の属性である。 D は、その階層から算出される集約値を値とする属性である。ステージアップ操作によりスキーマは、上位レベルの属性 A_{n+1} という属性が新たに加わり、メタオブジェクトを値としそのメタオブジェクトのスキーマは A_n と D からなる。この時のタプルは、 u_1, u_2, \dots, u_m から構成されるとする。つまり、ステージアップによりスキーマは $(A_{n+1}, B, C, \nabla_g[A_n, D])$ で構成される。一段上位レベルに移動したステージアップの結果に対して、ドリルダウン操作を行う。ドリルダウン操作により、属性 A_{n+1} は削除される。属性 ∇_g の値であるメタオブジェクトは評価され、その属性のスキーマは A_n, D で構成する。つまり、ドリルダウン操作でスキーマは (B, C, A_n, D) で構成される。 u_1, u_2, \dots, u_m で構成されたタプルは、メタオブジェクトの評価により t_1, t_2, \dots, t_n になる。

よって、ステージアップ操作を行う前のスキーマとドリルダウン操作によるスキーマは一致し、タプルの数も一致する。

同様に、ドリルダウン操作をする前のスキーマを $T(A_n, \nabla_g[A_{n-1}, D], B, C)$ であるとする、ドリルダウン操作によりスキーマは、 $T(A_{n-1}, D, B, C)$ になり、これをステージアップ操作によりスキーマは

$T(A_n, B, C, \nabla_g[A_{n-1}, D])$ となりドリルダウン操作をする前と一致する。タプルも同様に、ドリルダウン操作をする前のタプルを v_1, v_2, \dots, v_m であるとする。ドリルダウン操作により w_1, w_2, \dots, w_n となりステージアップ操作により v_1, v_2, \dots, v_m となる。

この結果から、ステージアップ操作の逆操作はドリルダウン操作の逆操作であることがわかる。

7 むすび

本稿は、メタオブジェクト操作を用いて、新たに加えた視点を定めるステージアップという新しい操作を提案し、その操作と連携することで、OLAP 操作のロールアップ、ドリルダウンを定義することを試みた。

その結果、メタオブジェクトはデータウェアハウスの次元構造と対応づけることができ、関係代数操作とメタデータ操作、グループ、集約関数操作を用いてドリルダウン、ロールアップ操作を定義することができた。

参考文献

- [1] S. Chaudhuri et al: An Overview of Data Warehousing and OLAP Technology. ACMSIGMOD Record, Vol. 26 No.1 P.65-74, March 1997
- [2] Jim Gray, Surajit Chaudhuri, Adam Bosworth, Andrew Layman, Don Reichart, Murali Venkatesh, Frank Pellow, Hamid Pirahesh: Data Cube: A Relational Aggregation Operator Generalizing Group-by, Cross-Tab, and Sub Totals. Data Mining and Knowledge Discovery 1(1): 29-53 (1997)
- [3] Yihong Zhao, Prasad Deshpande, Jeffrey F. Naughton: An Array-Based Algorithm for Simultaneous Multidimensional Aggregates. SIGMOD Conference 1997: 159-170
- [4] 柏川伸悟, 松本渉, 三浦孝夫, 塩谷勇: データベースにおけるグループ化と集約関数の形式化 DEWS2001
- [5] Miura, T., Matsumoto, W.: Managing Meta Objects for Design of Warehouse Data, *proc. DaWaK* (1999), pp.33-40