

## ボトムアップデータベースにおける型情報を用いた検索

中田 充<sup>†</sup> 宝珍輝尚<sup>††</sup> 都司達夫<sup>††</sup>

<sup>†</sup>福井大学大学院工学研究科 <sup>††</sup>福井大学情報工学科

サイエンティフィック DBMS DREAM は、スキーマを定義することなく、データを格納可能なボトムアップデータベース管理システムである。DREAM では、データの検索に属性名や型に関する情報であるシェイプを利用することができる。本論文では、関連を陽に表現可能とするために DREAM のデータモデルを拡張し、シェイプを再検討する。DREAM では、オブジェクトの集合をバンドルで表現するが、バンドルの入れ子を可能とすることで関連の表現に対処する。また、シェイプは、値に基づく表現を行っているが、識別子を導入してオブジェクトに基づく表現とすることで関連に対処する。さらに、本論文では、DREAM モデルにおける関連の表現ならびに検索について検討する。

## Retrieval on Bottom Up Databases Using Data Type Information

Mitsuru NAKATA<sup>†</sup> Teruhisa HOCHIN<sup>††</sup> Tatsuo TSUJI<sup>††</sup>

<sup>†</sup>Graduate School of Engineering, Fukui University

<sup>††</sup>Department of Information Science, Fukui University

Scientific DBMS DREAM is a bottom up database management system for storing data without schema definitions. The shape is the information about data types and attribute names of objects. Users can query on a bottom up database by using the shape. In this paper, we will extend the data model of DREAM, and revise the shape in order to express relationships among objects directly. In the DREAM, a set of objects is the bundle. We extend the bundle to be able to include other bundles, and the shape to include identifiers of other shapes. These extensions enable users to express relationships directly, and to use them in retrieval.

## 1 はじめに

科学技術データを扱うサイエンティフィックデータベースでは、スキーマを定義することなく未整理のデータを格納し、後からデータを整理分類することでデータベースを構築する機構が必要とされる。すなわち、データベースの構造はあらかじめ決定されるのではなく、データの整理分類に沿って徐々に決定される。このように徐々に構造を決定していけるデータベースをボトムアップデータベース[1]と呼ぶ。筆者らは、科学技術データを柔軟に管理することを目的として、サイエンティフィック DBMS DREAM[2][3]を設計実現しているが、DREAMもボトムアップデータベースを管理するDBMSである。従来のデータベースでは、データの検索において、データベースのスキーマ、すなわち、属性名やデータの型情報を利用する。しかし、ボトムアップデータベースにはスキーマが存在しないため、この方法を利用することができない。さらに、ボトムアップデータベースを利用するアプリケーションプログラムは、C言語などの型情報を定義するプログラミング言語で実現されるのが一般的である。したがって、ボトムアップデータベースにおいても何らかの形で型情報を用いた検索が必要とされる。DREAMでは、スキーマに代わるデータの属性名や型に関する情報(シェイプと呼ぶ)を導入している[3][4]。

シェイプは、データの挿入・削除・変更により変化し、スキーマのようにデータ挿入に先立って存在しなくてよい。シェイプを用いることで従来のDBのような問い合わせがボトムアップDBに対しても可能であることを示してきた[3]。しかし、文献[2]で示したDREAMのデータモデル(DREAMモデル)では、関連を陽に表現することができないという問題がある。また、文献[3]で示したシェイプでは、いわゆる複合オブジェクトに対する問い合わせが煩雑であるという問題がある。

そこで本論文では、DREAMモデルにより高い表現能力を持たせるために、関連を陽に表現できるようにモデルを拡張し、シェイプを再検討する。

DREAMでは、オブジェクトの集合をバンドルとして表現可能であるが、バンドルの入れ子を可能とすることで関連の表現に対処する。また、シェイプは値に基づく表現を行っていたが、識別子を導入してオブジェクトに基づく表現とすることで対処する。さらに、DREAMモデルにおける関連の表現ならびに検索について検討する。

以降、2で本論文の前提として、DREAMモデルについて述べる。3ではシェイプの定義を行う。4では、DREAMモデルを用いた関連の表現とそれらのシェイプを示し、5でシェイプを用いた検索の例を示す。6はまとめである。

## 2 DREAMモデル

DREAMはデータモデルとして集合をもとにしたDREAMモデルを採用している。DREAMモデルには、データエレメント(de)、名前付きエレメント(ne)、視点(pe)、オブジェクト(obj)、バンドル(bndl)の5つの要素がある。データエレメントはデータ値を格納するための要素であり、組(id, value, type)である。ここで、idは識別子、valueはデータ値、typeはデータ型である。データエレメントはオブジェクトなどの他の要素から独立して存在できる。名前付きエレメントはデータエレメントとオブジェクトの集合に名前を付ける要素であり、3つ組(id, name, D)である。ここで、Dはデータエレメントまたはオブジェクトからなる集合である。nameは名前付きエレメントの名前であり視点の中で一意である。

視点は名前付きエレメントの集合に名前を付けたものであり、対象物の一つの側面を表す。視点の構造は3つ組(id, name, NE)である。nameは視点名でありオブジェクト内で一意である。NEは名前付きエレメントの集合を表す。一つの対象物を表すオブジェクトは、3つ組(id, name, PE)である。PEは視点の集合である。オブジェクトの名前nameは、一意である必要が無く空であってもかまわない。オブジェクトの集合は、DREAMモデルではバンドルで表現される。バンドルは、3つ組(id, name, S)である。ここで、nameはバンドル

ルの名前でありデータベース中で一意である。S はオブジェクトの集合である。

これまでに述べた DREAM モデルは、フォーマットの定まらない半構造のデータを柔軟に表現可能である。しかし、関連を陽に表現できないという問題がある。そこで、バンドルを入れ子が可能なように拡張することで関連を表現する。

#### [定義 1] バンドルの拡張

バンドル  $bnd$  は 3 つ組  $(id, name, S)$  である。ここで、 $id$  はバンドルの識別子、 $name$  はバンドルの名前である。S は  $\{x \mid x \in OBJ \text{ 又は } x \in BND\}$  である。ここで、OBJ はオブジェクトの集合であり、BND は  $bnd$  以外のバンドルからなる集合である。□

### 3 シェイプ

#### 3.1 シェイプエン트리

シェイプエント리는、シェイプの基本的な構成子である。シェイプエント리는、名前付きエレメントの名前とデータ型に関する情報を表す。名前付きエレメント  $ne$  のシェイプエン트리  $se$  の定義を以下に示す。

#### [定義 2] シェイプエン트리 $se$

名前付きエレメント  $ne$  のシェイプエン트리  $se$  は 3 つ組  $(id, nm, DT)$  である。ここで、 $id$  はシェイプエント리의識別子、 $nm$  は名前付きエレメント  $ne$  の名前であり、 $DT$  は名前付きエレメント  $ne$  に含まれる要素のデータ型の集合を表す。以降では、名前付きエレメント  $ne$  のシェイプを  $se(ne)$  で表す。□

#### 3.2 強和集合

シェイプエント리의名前は一意ではなく、同じ名前を持つ複数の異なるシェイプエント리가同時に存在しうる。ここで、シェイプエント리의強和集合を導入する。シェイプエント리의強和集合では、同じ名前を持つ複数のシェイプエント리는存在しない。以下にシェイプエント리의強和集合の定義を示す。

#### [定義 3] シェイプエント리의強和集合

シェイプエント리의強和集合の定義は、以下の

とおりである。

1) 空集合は、強和集合である。

2) 強和集合 S とシェイプエン트리  $se_1$  を考える。 $se_1$  と同じ名前を持つシェイプエント리가 S 中にない場合、S と  $\{se_1\}$  の和集合は強和集合である。

3) シェイプエン트리  $se_2$  が強和集合 S 中のシェイプエン트리  $se_1$  と同じ名前を持つとすると、集合  $\{x \mid x \in S, x \neq se_2\}$  と、 $se_2$  と同じ名前を持ち DT が  $\{dt \mid dt \in DT(se_1) \vee dt \in DT(se_2)\}$  であるシェイプエン트리からなる集合の和集合は強和集合である。

ここで、 $DT(se_1)$  は  $se_1$  のデータ型の集合である。

S' をシェイプエント리의集合とした場合、S' の強和集合を  $\{s \mid s \in S'\}$  や  $S'$ 、または、 $S'$  と表す。□

たとえば、 $se_1:(id1, \text{"weight"}, \{int, float\})$ 、 $se_2:(id2, \text{"weight"}, \{double\})$ 、 $se_3:(id3, \text{"length"}, \{int\})$  を考えた場合、 $\{se_1, se_3\}$  は強和集合であり、 $\{se_1, se_2, se_3\}$  は強和集合ではない。したがって、以下の 3 式が成立する。

$$\{se_1, se_3\} = \{se_1, se_3\}$$

$$\{se_1, se_2, se_3\} \neq \{se_1, se_2, se_3\}$$

$$\{se_1, se_2, se_3\} = \{(id4, \text{"weight"}, \{int, float, double\}), (id3, \text{"length"}, \{int\})\}$$

ここで、 $(id4, \text{"weight"}, \{int, float, double\})$  は、 $se_1, se_2, se_3$  からなる集合が強和集合となるために、同じ名前を持つ  $se_1$  と  $se_2$  が一つに統合された結果生じたシェイプエン트리である。

#### 3.3 サンプルデータベース

ここでは、以降の説明に利用するサンプルのデータベース  $ddb$  について述べる。データベース  $ddb$  には、3 人の学生 (学生 A, B, C) と 2 つの科目 (情報実験と物理学) に関する情報が格納されている。データベース  $ddb$  には以下のような要素がある。

#### データエレメント

$(deid1, A, string)$ ,  $(deid2, XXX, string)$ ,  $(deid3, 001,$

$int)$ ,  $(deid4, 1234, string)$ ,  $(deid5, B, string)$ ,

$(deid6, YYY, string)$ ,  $(deid7, 002, int)$ ,  $(deid8, 1235,$

$string)$ ,  $(deid9, C, string)$ ,  $(deid10, ZZZ, string)$ ,

$(deid11, 003, int)$ ,  $(deid12, 1236, string)$ ,  $(deid13, 情報$

$実験, string)$ ,  $(deid14, 4, int)$ ,  $(deid15, 物理学, string)$ ,

(deid16, 2, int)

### 名前付きエレメント

(neid1-1, “氏名”, {de1}), (neid1-2, “住所”, {de2}),  
 (neid1-3, “No”, {de3}), (neid1-4, “TEL”, {de4})  
 (neid2-1, “氏名”, {de5}), (neid2-2, “住所”, {de6}),  
 (neid2-3, “No”, {de7}), (neid2-4, “TEL”, {de8})  
 (neid3-1, “氏名”, {de9}), (neid3-2, “住所”, {de10}),  
 (neid3-3, “No”, {de11}), (neid3-4, “TEL”, {de12})  
 (neid6-1, “科目名”, {de13}), (neid6-2, “単位”, {de14})  
 (neid7-1, “科目名”, {de15}), (neid7-2, “単位”, {de16})

### 視点

(peid1-1, “def”, {ne1-1, ne1-2, ne1-3, ne1-4}),  
 (peid2-1, “def”, {ne2-1, ne2-2, ne2-3, ne2-4})  
 (peid3-1, “def”, {ne3-1, ne3-2, ne3-3, ne3-4})  
 (peid6-1, “def2”, {ne6-1, ne6-2}),  
 (peid7-1, “def2”, {ne7-1, ne7-2}),

### オブジェクト

obj1 = (oid1, “”, {pe1}), obj2 = (oid2, “”, {pe2}),  
 obj3 = (oid3, “”, {pe3}), obj6 = (oid6, “”, {pe6}),  
 obj7 = (oid7, “”, {pe7})

### バンドル

bndl1 = (bndl1d1, “学生”, {oid1, oid2, oid3})  
 bndl2 = (bndl1d2, “科目”, {oid6, oid7})

通常、オブジェクトには複数の視点が属しており、その形も様々であるが、ここでは簡単化のために、各オブジェクトは、名前defまたはdef2の視点を一つだけ持ち、学生や科目などの対象ごとに同じ構造を持つとする。また、学生、科目のそれぞれのオブジェクトは、バンドル学生、バンドル科目に属している。

## 3.4 シェイプの定義

シェイプは、基本的には名前とシェイプエントリの集合の組である。[定義 4] にシェイプの構造の定義を示す。

### [定義 4] シェイプの構造

S をシェイプとすると、シェイプ S は 3 つ組(id, name, SE)である。ここで、id はシェイプの識別子、name はシェイプの名前、SE はシェイプエントリの強和集合である。□

シェイプには、1)視点のシェイプ、2)オブジェ

クトのシェイプ、3)バンドルのシェイプ、4)データベースのシェイプの 4 種類があるが、1)-3)の構造は [定義 3] に従う。以降では、3.3のデータベース ddb1 を例に視点とバンドルのシェイプについての説明する。

## 3.5 視点のシェイプ

視点 pe のシェイプとは、視点 pe に属する名前付きエレメントのシェイプエントリを pe のレベルでまとめたものである。

### [定義 5] 視点のシェイプ s(pe)

視点 pe のシェイプ(id, name, SE)は以下の条件を満たす。

- id はシェイプの識別子である。
- name は視点 pe の名前である。
- SE は視点 pe に属する名前付きエレメントのシェイプエントリの強和集合であり、SE = [se(ne) | ne ∈ pe.NE] である。

以降では、視点 pe のシェイプを s(pe)と表す。□

つぎに、視点のシェイプに関する表記法を示す。S<sub>pe</sub>({pe1})は、視点 pe1 のシェイプの強和集合を表す。また同様に、S<sub>pe</sub>(obj1)はオブジェクト obj1 に属する視点のシェイプの強和集合を表す。

S<sub>pe</sub>(bndl1)はバンドル bndl1 に属する視点のシェイプの強和集合を表し、S<sub>pe</sub>(ddb1)はデータベース ddb1 に属する視点のシェイプの強和集合を表す。以下にデータベース ddb1 における視点のシェイプに関する演算 S<sub>pe</sub>(bndl1)を示す。

S<sub>pe</sub>(bndl1) = [(sid, “def”, [(“氏名”, {string}), (“住所”, {string}), (“No”, {int}), (“TEL”, {string})]), (sid, “def2”, [(“科目名”, {string}), (“単位”, {int})])]

## 3.6 バンドルのシェイプ

バンドル bndl のシェイプとは、バンドル bndl に属する名前付きエレメントのシェイプエントリを bndl のレベルでまとめたものである。ただし、バンドルはバンドルを含むことが可能である (バンドルの入れ子)。そのため、バンドルのシェイプには、バンドルに属する名前付きエレメントのシェイプエントリだけでなく、入れ子になっているバンドルのシェイプも含まれる。

**[定義 6] バンドルのシェイプ s(bndl)**

バンドル bndl のシェイプを s(bndl)と表す. バンドル bndl のシェイプ(id, name, X)は以下の条件を満たす.

- id はシェイプの識別子である.
- name はバンドルの名前である.
- X は, バンドル bndl に属する名前付きエレメントのシェイプエントリの強和集合 SE と, バンドル bndl に属するバンドルのシェイプの識別子の集合との和集合である. □

$S_{bndl}(\{bndl1\})$  は, バンドル bndl1 のシェイプの強和集合であり,  $S_{bndl}(bndl1) = [s(bndl1)]$  である.

$S_{bndl}(ddb1)$  はデータベース ddb1 に属するバンドルのシェイプの強和集合を表す. 以下に, データベース ddb1 におけるバンドルのシェイプの表記法の例を示す.

$$S_{bndl}(bndl1) = [(sid, "学生", [{"氏名", {string}}, {"住所", {string}}, {"No", {int}}, {"TEL", {string}}])]$$

**3.7 シェイプエントリ se の再定義**

名前付きエレメントは, データエレメントだけではなくオブジェクトも要素として含むことが出来る. ここで, シェイプエントリの再定義を行う.

**[定義 7] シェイプエントリ se の再定義**

名前付きエレメント ne のシェイプエントリ se は 3 つ組(id, nm, X)である. ここで, id はシェイプエントリの識別子, nm は名前付きエレメント ne の名前である. また, X は  $X = DT + S$  である. ここで, DT は名前付きエレメント ne に含まれる要素のデータ型の集合であり, S は SID1 と SID2 の和集合である. SID1 は名前付きエレメント ne に属するオブジェクトのうち, どのバンドルにも属さないオブジェクトのシェイプの識別子の集合であり, SID2 は名前付きエレメント ne に含まれるオブジェクトが属するバンドルのシェイプの識別子の集合である. □

**4 関連の表現**

ここでは, 図 1 のような「学生が科目を履修する」という関連を DREAM モデルで表現する. 以降では, 表記の簡単化のために, 値 50 を持つデ

ータエレメントを参照し, 名前が "AA" である名前付きエレメントを, (id, "AA", {50}) と表す.

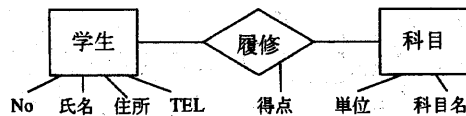


図 1 関連の例

**4.1 関連の表現方法**

関連の表現方式には以下のものがある.

- 1) 関連を表現するオブジェクトを作成する.
- 2) オブジェクト間の関連をオブジェクトの相互参照で表現する.
- 3) バンドルにより関連を表現する

1), 2) の表現方式には, オブジェクト同士の繋がりを, I) 識別子で表現する方法と, II) 値によって表現する方法とがある. したがって, 関連の表現方式には, 1)-I), 1)-II), 2)-I), 2)-II), 3) の 5 通りの方式がある. ここでは, これら 5 通りの方式について上で挙げた関連を表現し, それらのシェイプを示す. オブジェクト間の関連としては, 関連1: 「学生A,B,Cが情報実験を履修する」, 関連2: 「学生A,Cが物理学を履修する」の2つを考える.

**4.1.1 方式 1)-I)を用いた表現**

関連 1 を表現するために, 以下のような履修オブジェクトを作成し, それらをバンドル履修に属させる (図 2).

$$objX = (oidX, "", \{(peidX-1, "def", NEX)\})$$

$$NEX = \{(neidX-1, "学生", \{oid1\}), (neidX-2, "科目", \{oid6\}), (neidX-3, "得点", \{60\})\}$$

$$objY = (oidY, "", \{(peidY-1, "def", NEY)\})$$

$$NEY = \{(neidY-1, "学生", \{oid2\}), (neidY-2, "科目", \{oid6\}), (neidY-3, "得点", \{80\})\}$$

$$objZ = (oidZ, "", \{(peidZ1, "def", NEZ)\})$$

$$NEZ = \{(neidZ-1, "学生", \{oid3\}), (neidZ-2, "科目", \{oid6\}), (neidZ-1, "得点", \{85\})\}$$

バンドルのシェイプを以下に示す.

$$s(bndl学生) = (sid1, "学生", [{"氏名", {string}}, {"住所", {string}}, {"No", {int}}, {"TEL", {string}}])$$

$$s(bndl科目) = (sid2, "科目", [{"科目名", {string}}, {"単位", {int}}])$$

$s(\text{bndl履修}) = (\text{sid3}, \text{"履修"}, \{(\text{"学生"}, \{\text{sid1}\}), (\text{"科目"}, \{\text{sid2}\}), (\text{"得点"}, \{\text{int}\})\})$

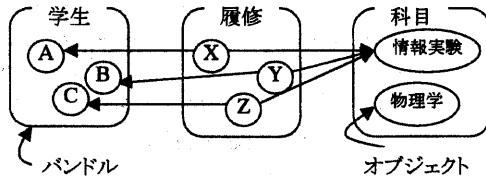


図 2 関連の表現 1

#### 4.1.2 方式 1)-II)を用いた表現

以下のような履修オブジェクトを作成することで関連 1 を表現する。方式 1)-I)と異なるところは、オブジェクト同士のつながりを値によって表現している点である。

$\text{objX} = (\text{oidX}, \text{""}, \{(\text{peidX-1}, \text{"def"}, \text{NEX})\})$

$\text{NEX} = \{(\text{neidX-1}, \text{"学生"}, \{\text{A}\}), (\text{neidX-2}, \text{"科目"}, \{\text{情報実験}\}), (\text{neidX-3}, \text{"得点"}, \{60\})\}$

$\text{objY} = (\text{oidY}, \text{""}, \{(\text{peidY-1}, \text{"def"}, \text{NEY})\})$

$\text{NEY} = \{(\text{neidY-1}, \text{"学生"}, \{\text{B}\}), (\text{neidY-2}, \text{"科目"}, \{\text{情報実験}\}), (\text{neidY-3}, \text{"得点"}, \{80\})\}$

$\text{objZ} = (\text{oidZ}, \text{""}, \{(\text{peidZ-1}, \text{"def"}, \text{NEZ})\})$

$\text{NEZ} = \{(\text{neidZ-1}, \text{"学生"}, \{\text{C}\}), (\text{neidZ-2}, \text{"科目"}, \{\text{情報実験}\}), (\text{neidZ-3}, \text{"得点"}, \{85\})\}$

このときのシェイプは以下ようになる。

$s(\text{bndl学生}) = (\text{sid1}, \text{"学生"}, \{(\text{"氏名"}, \{\text{string}\}), (\text{"住所"}, \{\text{string}\}), (\text{"No"}, \{\text{int}\}), (\text{"TEL"}, \{\text{string}\})\})$

$s(\text{bndl科目}) = (\text{sid2}, \text{"科目"}, \{(\text{"科目名"}, \{\text{string}\}), (\text{"単位"}, \{\text{int}\})\})$

$s(\text{bndl履修}) = (\text{sid3}, \text{"履修"}, \{(\text{"学生"}, \{\text{string}\}), (\text{"科目"}, \{\text{string}\}), (\text{"得点"}, \{\text{int}\})\})$

#### 4.1.3 方式 2)-I)を用いた表現

関連 1 を表現するために、学生、科目オブジェクトを以下のように変更する。

$\text{obj1} = (\text{oid1}, \text{""}, \{(\text{peid1-1}, \text{"def"}, \text{NE1})\})$

$\text{NE1} = \{\text{ne1-1}, \text{ne1-2}, \text{ne1-3}, \text{ne1-4}, (\text{neid1-5}, \text{"履修"}, \{\text{obj6}\})\}$

$\text{obj2}, \text{obj3}$ についても同様である。

$\text{obj6} = (\text{oid6}, \text{""}, \{(\text{peid6-1}, \text{"def"}, \text{NE6})\})$

$\text{NE6} = \{\text{ne6-1}, \text{ne6-2}, (\text{neid6-3}, \text{"履修"}, \{\text{obj1}, \text{obj2}, \text{obj3}\})\}$

さらに、バンドル履修を作成しバンドル学生とバンドル科目を属させることで学生と科目との関連を表現する(図 3)。

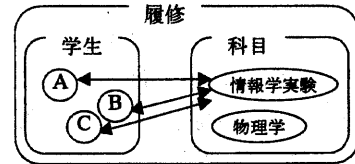


図 3 関連の表現 2

しかしながら、これでは関連“履修”の属性“得点”が表現できない。そこで以下のように、学生、科目オブジェクトを変更し、属性“得点”を表現するためにオブジェクトの作成を行う。

$\text{obj1} = (\text{oid1}, \text{""}, \{(\text{peid1-1}, \text{"def"}, \text{NE1})\})$

$\text{NE1} = \{\text{ne1-1}, \text{ne1-2}, \text{ne1-3}, \text{ne1-4}, (\text{neid1-5}, \text{"履修"}, \{\text{objX1}\})\}$

$\text{objX1} = (\text{oidX1}, \text{""}, \{(\text{peidX1-1}, \text{"def"}, \text{NEX1})\})$

$\text{NEX1} = \{(\text{neidX1-1}, \text{"科目"}, \{\text{obj6}\}), (\text{neidX1-2}, \text{"得点"}, \{60\})\}$

同様の変更、ならびに、オブジェクトの作成が $\text{obj2}, \text{obj3}$ についても必要である。そして、これらのオブジェクトをバンドル“科目の得点”に属させる。

科目オブジェクトは以下のように変更される。

$\text{obj6} = (\text{oid6}, \text{""}, \{(\text{peid6-1}, \text{"def"}, \text{NE6})\})$

$\text{NE6} = \{\text{ne6-1}, \text{ne6-2}, (\text{neid6-3}, \text{"履修"}, \{\text{objY1}, \text{objY2}, \text{objY3}\})\}$

$\text{objY1} = (\text{oidY1}, \text{""}, \{(\text{peidY1-1}, \text{"def"}, \text{NEY1})\})$

$\text{NEY1} = \{(\text{neidY1-1}, \text{"学生"}, \{\text{obj1}\}), (\text{neidY1-2}, \text{"得点"}, \{60\})\}$

$\text{objY2} = (\text{oidY2}, \text{""}, \{(\text{peidY2-1}, \text{"def"}, \text{NEY2})\})$

$\text{NEY2} = \{(\text{neidY2-1}, \text{"学生"}, \{\text{obj2}\}), (\text{neidY2-2}, \text{"得点"}, \{80\})\}$

$\text{objY3} = (\text{oidY3}, \text{""}, \{(\text{peidY3-1}, \text{"def"}, \text{NEY3})\})$

$\text{NEY3} = \{(\text{neidY3-1}, \text{"学生"}, \{\text{obj3}\}), (\text{neidY3-2}, \text{"得点"}, \{85\})\}$

ここで、 $\text{objY2}$ の $\text{neY1-2}$ の値60を格納するデータエレメントは、 $\text{objX1}$ の $\text{neX1-2}$ と共有されることに注意。

$\text{objX1}$ と同様に、これらのオブジェクトを、バンドル“学生の得点”に属させる(図 4)。

このときのシェイプは以下ようになる。

```

s(bndl学生) = (sid1, "学生", {"("氏名", {string}), ("住所", {string}), ("No", {int}), ("TEL", {string}), ("履修", {sidXX})})
s(bndl科目の得点) = (sidXX, "", {"("科目", {sid2}), ("得点", {int})})
s(bndl科目) = (sid2, "科目", {"("科目名", {string}), ("単位", {int}), ("履修", {sidYY})})
s(bndl学生の得点) = (sidYY, "", {"("学生", {sid1}), ("得点", {int})})

```

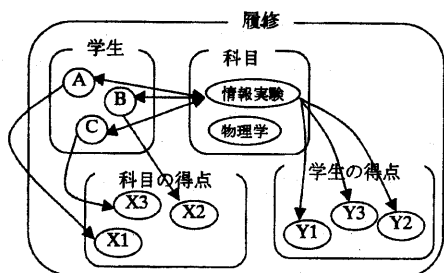


図 4 関連の表現 3

#### 4.1.4 方式 2)-II)を用いた表現

方式 2)-I)と同様の変更を学生、科目オブジェクトに行うことで関連 1 を表現できる。ただし、オブジェクト同士のつながりは、値によって表現される。たとえば、学生 A のオブジェクト obj1 に属する名前付きエレメントの集合 NE1 は、{ne1-1, ne1-2, ne1-3, ne1-4, (neid1-5, "履修科目名", {情報実験, 物理学})} となり、学生と科目の得点を関連づける (関連の属性を表現する) オブジェクト objX1 は、(oidX1, "", {(peidX1-1, "def", NEX1)}) となる。ここで、NEX1 = {(neidX1-1, "科目名", {情報実験}), (neidX1-2, "得点", {60})} である。オブジェクトは、値によって結びつけられている。この場合のシェイプを以下に示す。

```

s(bndl学生) = (sid1, "学生", {"("氏名", {string}), ("住所", {string}), ("No", {int}), ("TEL", {string}), ("履修科目名", {string})})
s(bndl科目の得点) = (sidXX, "", {"("科目名", {string}), ("得点", {int})})
s(bndl科目) = (sid2, "科目", {"("科目名", {string}), ("単位", {int}), ("履修者名", {string})})
s(bndl学生の得点) = (sidYY, "", {"("学生名", {string}), ("得点", {int})})

```

```
), ("得点", {int}))
```

#### 4.1.5 方式 3)を用いた表現

この方式では、個々の学生オブジェクトと科目オブジェクトの関連を表現するバンドルを作成し、さらにそれらをバンドル履修に属させる (図 5)。さらに、関連の属性“得点”を表現するためにオブジェクト objV, objW, objX, objY, objZ を作成する。

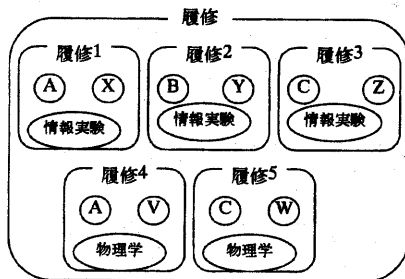


図 5 方式 3)を用いた関連の表現

```
objX = (oidX, "", {(peidX, "def", NEX)})
```

```
NEX = {(neidX-1, "得点", {60})}
```

```
objY = (oidY, "", {(peidY, "def", NEY)})
```

```
NEY = {(neidY-1, "得点", {80})}
```

```
objZ = (oidZ, "", {(peidZ, "def", NEZ)})
```

```
NEZ = {(neidZ-1, "得点", {85})}
```

```
bndl履修1 = (bndlid_a, "履修1", {obj1, obj6, objX})
```

```
bndl履修2 = (bndlid_b, "履修2", {obj2, obj6, objY})
```

```
bndl履修3 = (bndlid_c, "履修3", {obj3, obj6, objZ})
```

```
bndl履修 = (bndlid_f, "履修", {bndlid_a, bndlid_b, bndlid_c, bndlid_d, bndlid_e})
```

オブジェクト V, W や bndl 履修 4, 5 も同様である。

このときのシェイプは以下ようになる。

```
s(bndl履修1) = (sid_a, "履修1", {sid1, sidX, sid2})
```

```
s(bndl履修2) = (sid_b, "履修2", {sid1, sidY, sid2})
```

```
s(bndl履修3) = (sid_c, "履修3", {sid1, sidZ, sid2})
```

bndl 履修 4, 5 のシェイプも同様である。

```
s(bndl履修) = (sid_f, "履修", {sid_a, sid_b, ..., sid_e})
```

ここで、sid1, sid2 は 4.1.1 のものと同じであり、sidX は objX のシェイプの識別子である。

#### 5 シェイプを用いた検索

ここでは、1)-I), 1)-II), 2)-I), 2)-II), 3) の方式で表現された関連について以下の検索を行う。

検索 1: 情報実験を履修している学生の名前と No を検索する。

検索 2: 学生 A の全科目の得点を検索する。検索の結果は科目名と得点とする。

以降では、これらの検索をSQL風に表現する。

方式 1)-I)では、検索 1 は、科目名が情報実験である科目オブジェクトを含むオブジェクトをバンドル履修から探し、そのオブジェクトが含む学生オブジェクトの名前を表示することになる。その検索は以下のように記述できる。

```
select 履修学生氏名, 履修学生No from 履修
where 履修科目.科目名 = "情報実験";
```

検索2は以下のようになる。

```
select 履修科目.科目名, 履修得点 from 履修
where 履修学生氏名 = "A";
```

方式 1)-II)における検索 1,2 を以下に示す。

```
select 履修学生, 学生No from 履修, 学生
where 履修科目 = "情報実験"
and 履修学生 = 学生学生名;
```

```
select 履修科目, 履修得点 from 履修
where 履修学生 = "A";
```

方式 2)-I)における検索 1,2 を以下に示す。

```
select 学生氏名, 学生No from 学生
where 学生履修科目.科目名 = "情報実験";
select 学生履修科目.科目名, 学生履修得点
from 学生 where 学生氏名 = "A";
```

方式 2)-II)における検索 1,2 を以下に示す。

```
select 学生氏名, 学生No from 学生
where 学生履修科目名 = "情報実験";
select 科目の得点科目名, 科目の得点得点
from 学生, 科目の得点
where 学生氏名="A"
and 学生履修科目名=科目の得点科目名;
```

3)の方式における検索 1 を以下に示す。

```
select 履修学生氏名, 履修学生No from 履修
where 履修学生 in (select 学生 from 学生)
and 履修科目 in (select 科目 from 科目
where 科目名 = "情報実験");
```

この検索は、学生オブジェクトを含み、かつ、情報実験に関係するバンドルをバンドル履修から

検索し、それらに含まれる学生オブジェクトの氏名と No を表示している。下線が引かれた部分は、バンドル科目から、科目名が“情報実験”である科目オブジェクトを検索する。検索 2 についても、同様に検索可能であるが、ここでは省略する。

## 6 おわりに

本論文では、サイエンティフィック DBMS DREAM において関連を表現するための検討を行った。従来はオブジェクトの集合であったバンドルをバンドルの入れ子が可能であるように拡張し、シェイプには識別子を導入してオブジェクトに基づく表現が可能となるようにした。これにより、関連を陽に表現することが可能となった。つぎに、DREAM における関連の表現の 5 つの方式を示した。そして、表現方式が異なればシェイプも異なり、検索方法も異なることを実例を用いて示した。以上より、再検討したシェイプを用いることで関連の検索が容易に可能となることを明らかにした。

関連は、その表現方式が異なればシェイプも異なる。そのため、同じ関連を表現した要素であっても、表現方式が異なるものは統一的に検索が行えない。異なる表現方式で表現された関連の統一的な扱いは今後の課題である。

## 謝辞

本研究は、一部、文部省科学研究費重点領域研究(1) (課題番号 09204113) による。

## 参考文献

- [1] Stanley B. Zdonik et al. : Incremental Database Systems: Databases from the Ground Up, Proc. of ACM SIGMOD93, pp.408-412 (1993).
- [2] 中田充 他 : 名前付き集合モデルを用いた DREAM モデルの定義, 情処研報, 97-DBS-112, pp. 1-8 (1997).
- [3] Mitsuru Nakata, et al. : Bottom-Up Scientific Databases Based on Sets and Their Top-Down Usage, Proc. of the 1997 Int'l Database Engineering & Applications Symposium, pp.171-179 (1997).
- [4] 宝珍輝尚 他 : インスタンススペースのデータベースにおける柔軟なスキーマについて, 情処第 53 回全大, 3R-6, pp3-29-3-30 (1996).