

# インスタンスベースのデータベースにおける 柔軟なスキーマについて

3R-6

宝珍輝尚 中田充 都司達夫

福井大学 工学部 情報工学科

## 1 はじめに

近年、データベース(DB)応用分野が急速に広がってきている。この中には、データ定義(スキーマ)をあらかじめ全て決定することが困難な分野がある。このような分野では、スキーマをあらかじめ決定しなければならない従来のDBを利用することは困難である。

これに対処するために、著者らはDREAMモデルを提案している[1]。DREAMモデルでは、未整理データを格納する基本DBと基本DB中のデータを徐々に整理可能とする導出DBを考える。導出DBでは、従来n項組で表された事物を事物を集合として表現し、集合演算を用いて処理を行うことにより、DBの柔軟性を高めている。また、データ実体から従来のスキーマに相当する情報をボトムアップに生成して利用する枠組みをシェイプとして検討してきた[2]。シェイプはデータ実体の生成・削除・変更により変化し、柔軟性という点では優れている。しかし、従来からのDB設計手法を利用することが困難であるという問題がある。

そこで、本論文では、トップダウンなDB設計が可能でありながら、データ実体の生成・削除・変更によりボトムアップに変更可能なシェイプ(硬シェイプ)を提案する。以降、2で導出DBについて述べ、3でシェイプについて述べ、4で硬シェイプを提案する。

## 2 導出データベース

ここでは、シェイプを考えるために最低限必要な導出DBの定義を示す。データエレメントは導出DBにおいてデータの实体を格納する要素であり、導出DBにおけるデータの取り扱いの最小単位である。ここでは、簡単のため、データエレメントはデータ型とデータ値の組とする。データエレメントに名前を付けたものが名前付きエレメントである。オブジェクトは基本的には名前付きエレメントの集合である。そして、オブジェクトの集合はバンドルとしてまとめられる。

[定義1](データベースエレメント)

データエレメント  $de$  は、 $dt$  をデータ型、 $value$  をデータ値とすると、 $de = \langle dt, value \rangle$ 。名前付きエレ

メント  $ne$  は、 $name$  を名前、 $de$  をデータエレメントとすると、 $ne = \langle name, de \rangle$ 。導出データベース  $db$  は名前付きエレメントの集合であり、 $db = \{ne\}$ 。オブジェクト  $obj$  は、 $db$  の冪集合の要素( $obj \in 2^{db}$ )である。すなわち、オブジェクトは導出データベースに属する任意個の名前付きエレメントを要素とする集合であり、 $obj \subseteq db$ 。バンドル  $bndl$  は、オブジェクトの集合を  $OBJ$  とすると、 $OBJ$  の冪集合の要素と名前の組である。すなわち、バンドルは導出データベースに属する任意個のオブジェクトを要素とする集合と名前の組であり、 $bndl = \langle name, \{obj\} \rangle$ 。名前付きエレメント、オブジェクト、バンドルを導出データベースのデータベースエレメントと呼ぶ。□

本来、DBはデータエレメントの集合である[1]が、ここでは、名前のないデータの扱いを考えず、DBを名前付きエレメントの集合と定義している。

オブジェクトには視点を設定できる。視点は、オブジェクトに属する名前付きエレメントの集合である。

[定義2](視点)

オブジェクト  $obj$  の視点  $view(obj) = \langle name, NE \rangle$ 。ここで、 $name$  は視点の名前、 $NE \subseteq obj$ 。□

## 3 シェイプ

次に、シェイプをフォーマルに定義する。シェイプとは、データの形を記述する情報である。ここでの情報とは名前とデータ型である。

シェイプを定義する前に、若干の準備を行う。データエレメント  $de$  のデータ型を  $dt(de)$  と記述する。また、名前付きエレメント  $ne$  のデータエレメントを  $de(ne)$  と記述し、名前付きエレメント、視点、または、バンドル( $t$ と表記)の名前を  $name(t)$  と記述する。さらに、視点  $view$  に属する名前付きエレメントの集合を  $NE(view)$  と記述し、バンドル  $bndl$  に属するオブジェクトの集合を  $OBJ(bndl)$  と記述する。

まず、オブジェクトのシェイプを定義する。これは、オブジェクトに属する名前付きエレメントの名前とデータ型である。ここで注意しなければならないのは、同

一の名前を持つ名前付きエレメントのデータエレメントの型 (データ型) が一つとは限らないことである。

[定義3] (オブジェクトのシェイプ)

オブジェクト  $obj$  のシェイプ  $S(obj) = \{ \langle l, DT(l) \rangle \mid l = name(ne_i), DT(l) = \{ dt \mid dt = dt(d), d = de(ne_j), name(ne_j) = l \}, ne_i \in obj, ne_j \in obj \}$ . また, 導出データベース  $db$  全体のオブジェクトのシェイプ  $S_{obj}(db) = \{ S(obj) \mid obj \in 2^{db} \}$ .  $\square$

$S_{obj}(db)$  は集合を要素とする集合であることに注意。 $\langle l, DT(l) \rangle$  は, 従来の属性に対応する。すなわち,  $l$  は属性名,  $DT(l)$  は定義域に相当する。

次に, バンドルのシェイプを定義する。バンドルのシェイプは, 基本的にはシェイプに属するオブジェクトのシェイプと考えて定義できる。

[定義4] (バンドルのシェイプ)

バンドル  $bndl$  のシェイプ  $S(bndl) = \{ \langle l, DT_{bndl}(l) \rangle \mid DT_{bndl}(l) = \bigcup_{obj \in OBJ(bndl)} DT(obj, l) \}$ . また, 導出データベース  $db$  全体のバンドルのシェイプ  $S_{bndl}(db) = \{ \langle name(bndl), S(bndl) \rangle \mid OBJ(bndl) \in 2^{db} \}$ . ここで,  $DT(obj, l)$  はオブジェクト  $obj$  のシェイプ  $S(obj)$  の中で名前  $l$  を持つ  $\langle l, DT(l) \rangle$  の  $DT(l)$ .  $\square$

次に, 視点のシェイプを定義する。視点は基本的には名前付きエレメントの集合であり, オブジェクトのシェイプと同様に定義できる。

[定義5] (視点のシェイプ)

オブジェクト  $obj$  の視点  $view$  のシェイプ  $S(view(obj)) = \{ \langle l, DT(l) \rangle \mid l = name(ne_i), DT(l) = \{ dt \mid dt = dt(d), d = de(ne_j), name(ne_j) = l \}, ne_i \in NE(view), ne_j \in NE(view) \}$ . また, オブジェクト  $obj$  の全ての視点のシェイプ  $S_{view}(obj) = \{ \langle name(view_i), S(view_i(obj)) \rangle \mid view_i \text{ は } obj \text{ の視点} \}$ . さらに, 導出データベース  $db$  全体の視点のシェイプ  $S_{view}(db) = \{ \langle name(view_i), \bigcup S(view_i(obj)) \rangle \mid obj \in 2^{db} \}$ .  $\square$

導出DB全体のシェイプは, オブジェクト, 視点, バンドルのシェイプで表現できる。

[定義6] (導出データベースのシェイプ)

導出データベースのシェイプ  $S(db) = \langle S_{obj}(db), S_{view}(db), S_{bndl}(db) \rangle$ .  $\square$

次に, シェイプの振舞いについて述べる。ここでは, 説明を簡単にするために, 名前付きエレメントの名前に注目し, バンドルのシェイプ  $S(bndl)$  について考える。

シェイプは次の2つの条件を満足する。

条件1  $\forall d \in DE(bndl) \exists A \in S(bndl) (Dom(d) \in D(A))$

条件2  $\forall d \in DE(bndl) \forall A \in S(bndl) \forall B (Dom(d) \notin D(B) \Rightarrow \neg (B \subseteq A))$

ここで,  $DE(bndl)$  はバンドル  $bndl$  に属するオブジェクトのデータエレメントの集合を表す。また,  $Dom(d)$  は, データエレメント  $d$  のデータ型の定義域を表す。 $D(A)$  は,  $A = \langle l, DT_{bndl}(l) \rangle$  とすると,  $DT_{bndl}(l)$  の要素 (すなわち, データ型) の定義域の集合を表す。条件1では, 現在のシェイプに存在しない名前やデータ型を持つ名前付きエレメントが挿入されると, その名前やデータ型が含まれるようにシェイプが変わらなければならないことを表している。条件2では, 名前付きエレメントの削除によりその名前付きエレメントの名前やデータ型を持つような名前付きエレメントが存在しなくなった場合に, シェイプからその名前やデータ型を削除しなければならないことを表している。

以上, バンドルについて述べてきたが, 同様のことが, オブジェクト, 視点に関しても適用される。

#### 4 硬シェイプ

シェイプはデータの挿入・削除・変更によって変化する。これは, 柔軟性という点で有効であるが, データの削除によって属性がなくなるという問題や, DB設計手法を利用できないという問題がある。後者に対しては, 設計に合致するシェイプを生成するような疑似データの挿入により対処できるが, 好ましくない。

そこで, この問題を解決する硬シェイプを提案する。前述の2つの条件のうち条件1のみを満足する場合を”弱束縛である”と呼び, 弱束縛のシェイプを硬シェイプと呼ぶ。硬シェイプでは, データの削除によってシェイプを変更しなくても構わないし, データ挿入に先立ってシェイプが存在していても構わない。硬シェイプにおいては, いわゆる属性の削除は自動的に行わないので, 陽に属性の削除を行うインタフェースが必要である。

硬シェイプを導入することで, 柔軟性を保ちつつ, 従来からのDB設計手法を利用でき, また, 必要な属性の削除を避けることができる。

#### 5 おわりに

本論文では, インスタンスベースのデータベースにおいて, トップダウンなDB設計手法が利用可能でありながら, データ実体の生成・削除・変更によりボトムアップに変更が可能な硬シェイプを提案した。

今後は, シェイプの操作言語の検討等が課題である。

#### 参考文献

- (1) 中田, 宝珍, 都司: サイエントフィックデータベースのためのデータモデルの一提案, 情報学 DBS 研究会 101-9 (1995).
- (2) 中田, 宝珍, 都司: サイエントフィックデータモデルの一評価, 信学 データ工学研究会 DE 95-64 (1995).