

## 関数型データベースからの複合オブジェクトの生成

永江 尚義 有澤 博

横浜国立大学 工学部

本稿では、データベースから検索時に複合オブジェクトを生成することができる関数型データモデルについて述べる。我々の提案する関数型データモデル“AIS”では、データベースがデータの二項関係だけによって構成されるために1つのデータベースからユーザの求める様々な形の複合オブジェクトを検索することができる。さらに、データベース中のデータの双方向の二項関係を利用することによりデータの検索時にデータベース・アクセスの効率化を行なうことが可能であることを示す。

## Representation of complex objects on a functional data base

Hisayoshi Nagae Hiroshi Arisawa

Faculty of Engineering, Yokohama National University

In this paper, we describe a functional data model that can generate a complex object when we search the data in a database. In the “AIS” functional data model, we can create any objects which we wish to find, because a database is composed of binary relations of entities. And we will discuss a method of decreasing the number of accessing data using binary relations in a database when we create objects.

## 1 はじめに

最近では、データベースの応用分野の広がりなどを背景に次世代のデータベースシステムとしてオブジェクト指向の概念を導入したオブジェクト指向データベースが注目を集め、盛んに議論されている。オブジェクト指向の最大の特徴は、複雑な構造を持つデータ(複合オブジェクト)をデータベースにおいて内部構造を持ったオブジェクト(object)として直接表現することにある。これによりオブジェクト指向データベースではデータの内部構造を用いた高度なデータ操作などを定義しやすく、現実世界をより自然な形でモデル化することができる。しかし、データベース中のデータが内部構造を持つと複合オブジェクトを表現することは容易になるが、その反面データの中立性には大きな問題を抱えることになる。すなわち、オブジェクト指向データベースではデータベース中のオブジェクトが、ある階層構造の形式に構造化されているためにデータベースのユーザはデータベース中と同じ形式でデータを利用することはできても、それとは異なった形式でデータを利用することが難しくなってしまう。このことは、大量のデータを多数のユーザが様々な形式で利用することを目指しているデータベースの根本的な目的から考えれば非常に重大な問題である。

そこで、本稿ではオブジェクト指向とは異なった立場から次世代のデータベースのためのデータモデルを議論する。以下で議論する AIS モデルは関数型データモデルの範疇に含まれる。関数型データモデルでは、データの意味や構造情報の表現はオブジェクト指向と同様に重視するが、その際「もの」を表す主体(entity)自身には内部構造を持たせず、主体間の二項関係だけでデータをモデリングし、それ以上の複雑な構造をデータベース中に持ち込まない。そのため、そのままでは複雑な構造を持つオブジェクトを取り扱うことはできないが、データベースにデータ検索の際にデータを構造化して取り出す機構を付加し、データベースから何らかの方法によって構造化された検索データを作り出すことができれば、データベース中から自由な形の複合オブジェクトを作り出せる可能性があり、その構造化はオブジェクト指向より遥かに自由で拡張性に富むものとなる。

本稿ではこのような観点から検索時に構造化されたデータを持つ複合オブジェクトを生成できるようなデータベースのためのデータモデルについて議論する。また、AIS モデル中で複合オブジェクトを記述するための主体木式

を定義する。さらに検索処理の最適化の一手法として単純な二項関係によって構成される AIS モデルの特性を利用して、データ検索の際に不要なデータへのアクセスを大幅に削減する方法を示す。

## 2 AIS モデルとデータベース

本章では関数型データモデルとして著者らが提案している AIS モデルの基本概念である主体、連想を定義し、検索されたデータを構造を持ったデータとして記述するための主体木について述べる。

### 2.1 AIS モデルの基本要素

まず本節では、AIS モデルの基本要素(primitives)である主体、主体集合、連想、連想集合を定義する。

#### 2.1.1 主体と主体集合

AIS モデルではデータベース化する実世界において事物・事象として認識される全ての「もの」を主体(entity)と呼ぶ。主体は実世界中に存在する「もの」の代理物としての点であり、内部構造を持たず、この点がオブジェクト指向におけるオブジェクトと最も大きく異なる。また、ある同一の意味を表す主体の有限個の集まりを主体集合(entity set)と呼ぶ。

#### 2.1.2 連想と連想集合

主体間の対応付けは関数(function)として表現される。この関数は  $f(a_1) = b_1, f(a_2) = b_2, \dots$  のようにインスタンスを列挙することにより定義される ( $a_1, b_1$  などは全て主体である)。また、関数は複合値関数(multi-valued function)の場合もある(例:  $f(a_1) = \{b_1 b_2 b_3 b_4\}$ )。ここで、関数におけるインスタンスの対応の順序には意味がある。すなわち  $f(a_1) = \{b_1 b_2\}$  と  $g(a_1) = \{b_2 b_1\}$  は異なる関数である。さらに、AIS モデルの関数は重複を持った対応付けを許す(例:  $f(a_1) = \{b_1 b_2 b_1 b_2\}$ )。

このような点から、AIS モデルの主体の対応付けを表現する関数は従来の関数とは大きく異なるため、本稿では連想(association)と呼ぶ。また、ある主体集合に属する主体の集まりと別の主体集合に属する主体の集まりとの間の対応を表す連想の集まり、すなわち主体集合間の対応付けを連想集合(association set)と呼ぶ。ところで AIS モデルでは、ある連想集合を定義すると自動的にその連想の逆向きの対応付け(これを逆連想集

合 (inverse association set) と呼ぶ) が定義される。この逆連想集合は連想集合名の右肩に “-1” をつけて表す。例えば、連想集合  $f$  の逆連想集合は  $f^{-1}$  となる。

## 2.2 AIS モデルにおける複合オブジェクトの表現

本稿で提案した AIS モデルの特長の 1 つに複合オブジェクトの認識がある。すなわちユーザはデータの二項関係からなるデータベースからデータ値の階層構造で構成されるいわゆる複合オブジェクトを生成することができる。この複合オブジェクトは AIS モデルでは階層構造によって結び付けられている主体の集まりによって表現される。この階層構造の主体を主体木 (entity tree) と呼び、これを記述する表現式が主体木式 (entity tree expression) である。ところで、AIS データベースでは複合オブジェクト自体がデータベース中に蓄積されているわけではなく、検索によってデータベースから一時的に作成されるものにすぎない。

### 2.2.1 主体木を構成する 4 つの基本要素

#### 定義 2.1 主体

主体はデータベース化する実世界中において事物・事象として認識される全ての「もの」を表す概念である。例えば、個々の従業員や学生はそれぞれが 1 つの主体である。ただし、これらの従業員や学生は値によって表現されるわけではなく、それ自身の存在を表示不可能な主体としてデータベース中で直接表現される。表示不可能な主体はデータベース内では実際にはユーザには見えない内部識別子によって表されるが、本稿では便宜的に表示不可能な主体を  $a_1, b_1, c_1$  のような記号によって代用する。

#### 定義 2.2 主体列

主体を順序づけて並べたものを主体列 (entity sequences) と呼ぶ。主体列はその要素となる主体をブラケット (“<”>) で囲んで表す。例えば、主体が  $e_1, e_2, e_3$  という順序で並べられている主体列は  $\langle e_1 e_2 e_3 \rangle$  と表される。主体列において要素である主体の順番は意味を持つため、 $\langle a_1 a_2 a_3 \rangle$  と  $\langle a_2 a_1 a_3 \rangle$  は表す意味が異なる。

#### 定義 2.3 属性

AIS データベース中では、それぞれの主体集合は互いに対等で上下関係は存在しない。これに対し、複合オブ

ジェクトでは各オブジェクト間に明確な上下関係が存在する。当然、複合オブジェクトを表す AIS モデルの主体木でもその主体木中においてだけ各主体間には上下関係があると考えられる。そして、「主体木において上位の主体  $x$  が連想集合  $f$  によって下位の主体  $y_1, y_2, \dots, y_n$  と対応付けられている」とき、「主体  $x$  は属性 (attribute)  $f$  を持ち、その属性値 (attribute value) として  $y_1, y_2, \dots, y_n$  を持つ」といい、次のように記述する。

$$x \rightarrow f : \langle y_1 y_2 \dots y_n \rangle$$

ここで、コロンの (“:”) の左側の  $f$  は属性を表す属性名であり、右側はその属性の属性値である。また、特に連想集合が  $n : 1$  もしくは  $1 : 1$  対応の場合には

$$x \rightarrow g : z_1$$

と記述し、属性値はブラケットで囲まない。

#### 定義 2.4 タプル

AIS モデルの主体木では、属性の組みをタプル (tuple) として表現する。タプルは属性名と属性値を一つにまとめたものを角括弧 (“[ ]”) で囲んだものである。例えば、次のような 2 つの属性  $f, g$

$$x \rightarrow f : \langle y_1 y_2 \dots y_n \rangle$$

$$x \rightarrow g : z_1$$

の組は以下のようなタプルで記述される。

$$x \rightarrow [ f : \langle y_1 y_2 \dots y_n \rangle \quad g : z_1 ]$$

ここで、タプル中の属性の順序には意味がない。

### 2.2.2 主体木式と複合オブジェクト

AIS モデルではデータベースから検索されたデータは階層構造を持ったデータ値の集まりである主体木によって表現される。そして、前節の主体木の基本要素を組み合わせることで、どのような複雑な構造をもつ複合オブジェクトでも表現することができる。ここで、ブラケットや角括弧によって主体木を表現したものを主体木式 (entity tree expression) と呼ぶ。

いま、データベースから主体木を生成する際に最初に生成されるのは、その主体木の中心となる 1 つの主体集合を表す主体列である。これはある主体集合中の全ての主体をブラケットで囲み、属性名として主体集合名がつけられたただ 1 つの主体列だけが存在するデータである。例えば、主体集合  $A$  中の主体が  $a_1, a_2, \dots, a_n$  のとき、主体集合  $A$  を表す主体木は  $A : \langle a_1 a_2 \dots a_n \rangle$  と記述される。

このようにして生成された単純な1つの主体集合を表す主体木に対して、データベース中の連想によって結び付けられている主体群を埋め込むことにより、複雑な構造を持つデータ(すなわち複合オブジェクト)を表現する主体木を作り出すことができる。例えば、図2.1(a)で表されるAからBへの連想が存在するとき、主体木は図2.1(b)のように成長していく。図2.1(b)において角括弧の内側の主体列( $\langle b_1 b_2 \rangle$ ,  $\langle b_3 \rangle$ )はそれぞれ成長前の主体 $a_1, a_2$ と連想 $f$ によって結びつけられているB中の主体の集まりを表し、ブラケット内の主体の順序は成長前の主体と連想 $f$ によって対応づけられていた順序を表している。また、主体列の前に付けられた“f:”は連想集合 $f$ に従って主体木を成長させたことを意味している。また、ここで連想集合 $f$ によって対応づけられている主体が存在しないタプルは自動的に削除される。

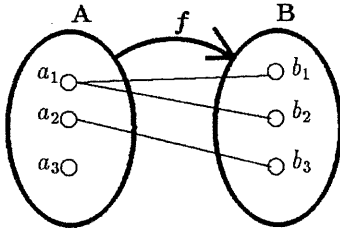


図 2.1 (a) 主体の対応付け

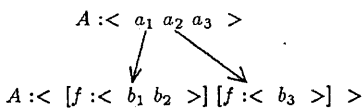


図 2.1 (b) 主体木の成長

また、ある主体を複数の連想集合に従って同時に置き換えると、次のようにタプル中に置き換えられた主体列が複数並んだ主体木が作られる。

$$A : \langle [f : \langle a_1 a_2 \rangle \quad g : \langle b_1 b_2 b_3 \dots \rangle \quad h : \langle c_1 \rangle] \dots \rangle$$

ここで、それぞれ $a_1, b_1, c_1, \dots$ は主体を、 $f, g, h$ は連想集合名を表す。

### 3 二項関係を利用したデータ検索の効率化

一般にデータベースが扱うデータ量は非常に膨大であるため、データベースへの検索を効率よく行なうためには、いかに必要なデータだけをアクセスするかが最大の関心事となる。本章では、データベース検索の最適化の一手法として、関係型データベースにおけるデータの二項関係を利用してデータベースのデータアクセスを軽減する手法について議論する。

### 3.1 データベースアクセスの最小化

いま、次のような検索を考えよう。

「女子学生のうち横浜国立大学院へ進学している学生を検索せよ」

構造をもったオブジェクトの階層構造によってデータベースが構築されている通常のオブジェクト指向データベースでは、このような検索を処理するためにはすべての学生オブジェクトに対してその性別と所属大学を調べるしかない。しかし、学生数が非常に膨大になると、すべての学生を検索処理するためには大量の処理時間と資源が必要になる。ここで、資源の節約のために遅延評価などを用いて検索途中の中間オブジェクトの爆発を押さえる方法などが考えられるが、検索条件を満足する最終的な解を検索するまでにデータベース中のすべてのデータを検索しなければならないことには変わりがない。

いま、データベースから性別が女性であるような学生だけを値の側から検索することができるとすると、この検索結果である女子学生の集合と所属大学側から検索された横浜国立大学院生の集合の論理和を求めればデータベース中のすべてのリンクを検索せずとも検索処理を行うことができる。このような処理方法によって、検索処理においてアクセスしなければならないデータの総量を省力化することができる。ただし、このようなアクセスデータの省力化を実現するためには、オブジェクトが持つインスタンス値やオブジェクトがどのようなものであるかというリンク情報とともに、ある値(もしくはオブジェクト)がどのようなオブジェクトによって所有されているかという逆向きのリンク情報もデータベースが管理していなければならない。我々が提案するAISモデルでは、データベース中には主体間の対等な二項の対応付けだけが蓄積されているためにこのような双方向のリンクがデータベース中の2主体間の対応付けとして自然に表現され、上記の検索処理のためのデータアクセスの省力化を容易に実現することができる。

### 3.2 条件検索と逆連想集合の利用

本節では、AISデータベース中の逆連想集合を使ってデータベースへのアクセスを軽減する手法について述べる。ここで、逆連想集合をどのように利用するかはユーザが記述した検索式中の条件式によって異なる。以下では、検索条件別にどのような形で逆連想集合を利用し、実際にどの程度の効率化が図られるかについて考察する。

### 3.2.1 定数比較を使った条件検索

以下では、図 3.1 のように単純な 2 つの主体集合と 1 つの連想集合だけが存在するようなデータベースを考え、このデータベースに対する検索について議論する。

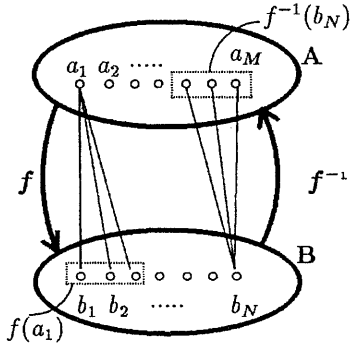


図 3.1 サンプルデータベース

ここで、主体集合  $A$  は印字不可能な主体の集合、主体集合  $B$  は値主体の集合であり、 $A$  から  $B$  への連想集合  $f$  は  $m : n$  対応であるとする。以下では、 $A$  中の主体  $a_1$  が  $f$  によって対応づけられている  $B$  中の主体列を  $f(a_1)$  と表記する。また、 $A$  中の主体の集合  $a_1, \dots, a_n$  を  $S$  とするとき、この主体の集合が  $f$  によって対応づけられている  $B$  中の主体列は  $f(S)$  と表記する。この  $f(S)$  は  $S$  の要素である主体  $a_1, \dots, a_n$  が連想集合  $f$  によって対応づけられる各々の主体列の論理和に等しい。

$$f(S) = f(a_1) \cup f(a_2) \cup \dots \cup f(a_n)$$

まず、検索式中に定数比較が含まれる場合について考える。定数比較とは「学籍番号が 20 番以降の学生だけを検索する」というようにデータベース中に実際に（属性値として）値が存在するものをそのままの形で利用した検索処理の形態を意味する。

#### 手法 1 定数比較

検索式の条件式が " $f(a_i) \theta \alpha$ " ( $\theta$  は任意の比較演算子) で表されるとき、 $b_i \theta \alpha$  を満足する  $B$  中の主体の集合  $S_\alpha$  に対して逆連想集合  $f^{-1}$  によって対応づけられている  $A$  中の主体列  $f^{-1}(S_\alpha)$  が  $f(a_i) \theta \alpha$  を満足する主体列となる。

いま  $B$  中の主体の個数を  $N$ 、条件  $(b_i \theta \alpha)$  を満足する主体の個数を  $N_\alpha$ 、 $B$  中の 1 つの主体が  $A$  中の主体と平均して  $m$  個のリンクによって結びつけられているとき、手法 1 により  $(N - N_\alpha) \times m$  個のリンクのアクセスを軽減することができる。

### 3.2.2 集合値を含む条件検索

次に、「履修単位が 200 を越えている学生を検索する」というようにデータベース中のデータに対して集約関数を適用させて得られるような値を利用した検索処理について考える。

#### 1. 最大値 (maximum)

手法 2  $\max(f(a_i)) = \alpha$

検索式の条件式が " $\max(f(a_i)) = \alpha$ " で表されるとき、 $b_i = \alpha$  を満足する  $B$  中の主体  $b_\alpha$  に対して逆連想集合  $f^{-1}$  によって対応づけられている  $A$  中の主体列  $f^{-1}(b_\alpha)$  と  $b_i > \alpha$  を満足する  $B$  中の主体の集合  $S_{>\alpha}$  に対して逆連想集合  $f^{-1}$  によって対応づけられている  $A$  中の主体列  $f^{-1}(S_{>\alpha})$  との差  $(f^{-1}(b_\alpha) - f^{-1}(S_{>\alpha}))$  を求めれば、 $\max(f(a_i)) = \alpha$  を満足する主体列となる。

いま  $B$  中の主体の個数を  $N$ 、条件  $(b_i \geq \alpha)$  を満足する主体の個数を  $N_\alpha$ 、 $B$  中の 1 つの主体が  $A$  中の主体と平均して  $m$  個のリンクによって結びつけられているとき、手法 2 により  $(N - N_\alpha) \times m$  個のリンクのアクセスを軽減することができる。

手法 3  $\max(f(a_i)) > \alpha$

$b_i > \alpha$  を満足する  $B$  中の主体の集合  $S_{>\alpha}$  に対する  $f^{-1}(S_{>\alpha})$  を求めれば、 $\max(f(a_i)) > \alpha$  を満足する主体列となる。

手法 4  $\max(f(a_i)) < \alpha$

$A$  中のすべての主体の集合  $S_A$  と  $b_i \geq \alpha$  を満足する  $B$  中の主体の集合  $S_{\geq\alpha}$  に対する  $f^{-1}(S_{\geq\alpha})$  との差  $(S_A - f^{-1}(S_{\geq\alpha}))$  を求めれば、 $\max(f(a_i)) < \alpha$  を満足する主体列となる。

#### 2. 最小値 (minimum)

手法 5  $\min(f(a_i)) = \alpha$

$b_i = \alpha$  を満足する  $B$  中の主体  $b_\alpha$  に対する  $f^{-1}(b_\alpha)$  と  $b_i < \alpha$  を満足する  $B$  中の主体の集合  $S_{<\alpha}$  の  $f^{-1}(S_{<\alpha})$  との差  $(f^{-1}(b_\alpha) - f^{-1}(S_{<\alpha}))$  を求めれば、 $\min(f(a_i)) = \alpha$  を満足する主体列となる。

手法 6  $\min(f(a_i)) > \alpha$

$A$  中のすべての主体の集合  $S_A$  と  $b_i \leq \alpha$  を満足する  $B$  中の主体の集合  $S_{\leq\alpha}$  に対する  $f^{-1}(S_{\leq\alpha})$  との差  $(S_A - f^{-1}(S_{\leq\alpha}))$  を求めれば、 $\min(f(a_i)) > \alpha$  を満足する主体列となる。

#### 手法 7 $\min(f(a_i)) < \alpha$

$b_i < \alpha$  を満足する B 中の主体の集合  $S_{<\alpha}$  に対する  $f^{-1}(S_{<\alpha})$  を求めれば、 $\min(f(a_i)) < \alpha$  を満足する主体列となる。

このように、通常データベース中のすべてのリンク情報をアクセスしなければデータベースから計算することができない集約値でさえも、AIS データベースでは双方向の二項関係を利用することによりデータのアクセス量を軽減することができる。

しかし、総数、平均値、総和といった集約値に対しては二項関係を利用しても、それほどアクセス量の大幅な軽減を行なうことはできない。これは、「総数」や「総和」はデータベース中のすべてのリンク情報をすべてアクセスしなければ得ることができない情報であり二項関係をいかに利用しても本質的にこのアクセス量を減らすことはできないためである。総和を総数で除算したものである「平均値」に対しても同様である。

### 3. 平均値 (average)

前述したように、 $ave(f(a_i))$  というような平均値を計算するには A 中の主体  $a_i$  と B の主体列と結びつけられているすべてのリンクを必ずアクセスしなければならない。そこで以下では、検索式の条件文中に  $ave(f(a_i))$  が含まれているとき、二項関係を利用して平均値を求めなければならない A 中の主体を減らし、それによってデータベースへのアクセスを軽減する手法について議論する。

#### 手法 8 $ave(f(a_i)) = \alpha$

$b_i \leq \alpha$  を満足する B 中の主体の集合  $S_{\leq\alpha}$  に対する A 中の主体列  $f^{-1}(S_{\leq\alpha})$  もしくは  $b_i \geq \alpha$  を満足する B 中の主体の集合  $S_{\geq\alpha}$  に対する A 中の主体列  $f^{-1}(S_{\geq\alpha})$  に対してのみ平均値を求めればよい。

#### 手法 9 $ave(f(a_i)) > \alpha$

$b_i > \alpha$  を満足する B 中の主体の集合  $S_{>\alpha}$  に対する A 中の主体列  $f^{-1}(S_{>\alpha})$  に対してのみ平均値を求めればよい。

#### 手法 10 $ave(f(a_i)) < \alpha$

$b_i < \alpha$  を満足する B 中の主体の集合  $S_{<\alpha}$  に対する A 中の主体列  $f^{-1}(S_{<\alpha})$  に対してのみ平均値を求めればよい。

### 4. 総和 (sum)

#### 手法 11 $sum(f(a_i)) = \alpha$

$b_i \leq \alpha$  を満足する B 中の主体の集合  $S_{\leq\alpha}$  に対する A 中の主体列  $f^{-1}(S_{\leq\alpha})$  に対してのみ総和を求めればよい。

#### 手法 12 $sum(f(a_i)) < \alpha$

$b_i < \alpha$  を満足する B 中の主体の集合  $S_{<\alpha}$  に対する A 中の主体列  $f^{-1}(S_{<\alpha})$  に対してのみ総和を求めればよい。

## 4 おわりに

本稿では、構造化されたデータを持つ複合オブジェクトを自由な形で生成するための関数型データモデルについて議論した。また、このデータベースから検索によって生成される複合オブジェクトを記述するための主体木式をあわせて定義した。これにより、ユーザはデータベースから高度にしかも自由な形式に構造化された複合オブジェクトを検索することができるようになる。

さらに、データベース中の二項関係を利用してデータベースへのアクセスを軽減する手法を提案した。また、逆連想を利用したデータベース・アクセスの軽減は単なる定数比較による条件検索だけでなく、集約値を含む条件文による検索の際にも有効であることが示された。

## 参考文献

- 1) Buneman, P., Frankel, R.E. and Nikhil, R.: An Implementation Technique for Database Query Languages, *ACM Trans. Database Syst.*, Vol.7, No.2, pp.164-186 (June 1982).
- 2) Shipman, D.W.: The Functional Data Model and the Data Language DAPLEX, *ACM Trans. Database Syst.*, Vol.6, No.1, pp.140-173 (Mar. 1981).
- 3) Batory, D.S., Leung, T.Y. and Wise, T.E.: Implementation Concepts for an Extensible Data Model and Data Language, *ACM Trans. Database Syst.*, Vol.13, No.3, pp.231-262 (Sept. 1988).
- 4) Arisawa, H., Nagae, H. and Mochizuki, Y.: Representation of Complex Objects on Semantic Data Model AIS and Implementation of Set Operators, *IEICE Trans.*, E-74, Vol.1, pp.191-203 (Jan. 1991).