

# サイエンティフィックデータベースの ためのデータモデルの一提案

～考古学データベースを例として～

中田 充      宝珍 輝尚      都司 達夫  
福井大学 情報工学科

本論文ではサイエンティフィックデータベースへのさまざまな要求について考察し、この要求を満たす新たなデータモデルを提案する。本データモデルの主な目的は、(1)データベース化を考慮していないデータを柔軟に扱うこと、(2)同じ対象物に対して異なる属性名や属性値を付与可能とすること、(3)データやオブジェクトの妥当性を検証可能とすることである。

(1)のために、データベースをオブジェクトではなくデータの集合ととらえ、未整理のデータを画一的に扱える基本データベースと整理済みのデータを格納する導出データベースを導入する。さらに、名前付けされたデータの集合をオブジェクトとし、オブジェクトの集まりをバンドルとする。(2)のために対象物を表わすオブジェクトに対して視点を導入する。(3)のためにデータやオブジェクトの導出過程をシステムが管理する。

## A Proposal of Data Model for Scientific Databases

- For Archaeological Databases -

Mitsuru NAKATA, Teruhisa HOCHIN, and Tatsuo TSUJI

Department of Information Science, Fukui University

In this paper, the requirements for scientific databases are studied, and a new data model for satisfying these requirements is proposed. The proposed data model has three main purposes. First is that un-arranged data as well as arranged ones can flexibly be manipulated. Second is one target item can be referred through different attribute names and/or values. Third is the validity of data and objects can be verified.

For the first purpose, a database is regarded as a set of data rather than objects. Two types of databases: basic databases and derived databases are introduced. Furthermore, an object is considered as a set of named data, and a bundle as a set of objects. For the second, we introduce perspective to capture an object from several points of views. The derivation processes of data and objects are managed in the system for the third aim.

## 1.はじめに

近年、実験データ、測定データや文献情報データなどの科学技術データをデータベースとして管理したいという要求が強くなってきている。これらのデータをデータベース化しようとする、スキーマを決定せずにデータを格納したい、自分の研究や実験の成果を他の研究者にわかりやすく提示したい、他の研究者の成果を参照したい、といった要求があり、従来のデータベースではこれらの要求を満足できない。

そこで本論文では、このようなデータを管理するサイエンティフィックデータベースへのさまざまな要求について考察し、この要求を満たす新たなデータモデルを提案する。以降、2章では、サイエンティフィックデータベースに対する要求について述べ、3章ではその要求を満たすデータモデルについて説明する。4章で関連する研究との比較を行い、5章でまとめを行う。

## 2.サイエンティフィックデータベースへの要求事項

考古学データや実験データなどを対象としたサイエンティフィックデータベースへの要求として、次の9項目を考えている。

### 1) 半構造化データベース

サイエンティフィックデータベースに格納しようとするデータは、データベースに格納することを前提にして生成されているわけではない。このためにこれらの情報を格納するのに必要な完全なデータベーススキーマの決定が難しい。そこで最初は適当なデータ構造で格納し、後から最適と思われるスキーマを生成するといったことが要求される。

### 2) 段階的なオブジェクト同定

サイエンティフィックデータベースのデータは1)で述べたように半構造化データベースに格納される。そこで、データベース中のデータから試行錯誤的、段階的にオブジェクトを認識し、属性の追加、削除を行った後、それを再びデータベースに格納したいという要求がある。

### 3) 段階的なスキーマ生成

同定されたオブジェクトに対して、試行錯誤的、

段階的に様々な関連付けや分類を行い、段階的にスキーマを生成したいという要求がある。

### 4) オブジェクトの多様性の表現

ユーザは上記の操作を様々な見方や仮定から行う。従って一つのオブジェクトでもユーザの見方が違えばその属性名、属性値も異なる。そのために、各データに対して複数の見方に対応したオブジェクトを段階的に生成したいという要求がある。

### 5) スキーマの曖昧さの容認

同じ見方でオブジェクトを分類、整理する場合、そのオブジェクトの属性名や属性値はユーザによって異なることのほうが多い。しかも複数のユーザが作成したオブジェクトを一つのクラスに属させたいという要求も少なくない。しかしながら、現行のC++をベースにしたオブジェクト指向DBMSでは、クラスの定義は厳密でこのようなことは許されていない。そこで異なる属性を持った複数のオブジェクトを一つのクラスとして扱いたいという要求がある。

### 6) データやクラスの導出過程の管理

サイエンティフィックデータベースでは、複数の研究者が、あるデータからそれぞれ独自に自分の見方や仮定にあったデータを生成したいという要求がある。これをここではデータの導出と呼ぶ。導出されたデータを研究者が利用する場合、データがどのように導出されたかを確認できなければ、そのデータを使用できるかどうかの判断が難しくなる。これは特にそのデータを他の研究者が導出した場合に問題となる。そこで、データの導出過程を保存しそれらを提示してほしいという要求がある。

### 7) スキーマの統合、分割

ある別々の仮定の下に決定された2つのスキーマを一つのスキーマとしたいという要求がある。そこで2つのスキーマの統合機能が必要である。同様にスキーマの分割も必要である。これらはサイエンティフィックデータベースでは頻繁に起こりうると考えられるので、効率の良い方法で実現される必要がある。

### 8) オブジェクトの統合、分割

ある別々の仮定の下に生成された2つのオブジェクトを、後で同一のオブジェクトとして扱いたいという要求がある。そこで2つのオブジェクトを統合する機能が必要である。同様にオブジェクトの分割も必

要である。

### 9) 名前付けの自由度

複数の研究者が、それぞれ自分の仮定に基づいて未整理データからオブジェクトを同定しスキーマを決定していく場合、研究者は他の研究者がどのような名前をつけるか把握していないので、重複した名前を許し、名前付けの自由度を高めたいという要求がある。

## 3.サイエンティフィックデータベースのためのデータモデル

ここでは、サイエンティフィックデータベースの要求を満たすために新しく考案したデータモデルについて述べる。なお、福井県朝倉氏遺跡において発掘された越前焼の染め付けの碗、皿の破片を扱うデータベースの構築を例に説明する。

### 3.1.概要

発掘された陶器の破片のデータをデータベースを用いて管理する場合を考える。従来のデータベースではデータを格納する前にスキーマを決定するが、破片がどのような情報を持っているかは、すべての破片について調べた後でなければわからず、研究者によって異なる可能性もある。また破片が大量にある場合には、スキーマを一度に決定する事も難しい。そこで全ての破片についてその個々のデータを未整理の状態でもりあえずデータベースに格納する。各ユーザはそのデータベースから自分に必要な情報を整理、分別して、独立したデータベースを構築する。本データモデルでは前者を基本データベース（基本DB）、後者を導出データベース（導出DB）と呼ぶ。

ユーザは一つの破片を表わす基本DB中のデータ（以降、基本データ）を、導出DB中のデータエレメントという構成要素に分解する。次にユーザはデータエレメントに名前をつけた名前付きエレメントを作る。これによりデータを名前で識別することが可能になる。

名前付きエレメントは一つの破片の一部の情報のみを表わすので、一つの破片に対応するオブジェクトは名前付きエレメントの集合として定義される。ここで、各破片の持つ情報は研究者の仮定や見方によって名前や値が異なることがある。そこでオブジェクト

に視点を導入する。図1はある破片に対する仮定が2つあり、各仮定を表わすための視点毎に名前や値が違ふことを示している。

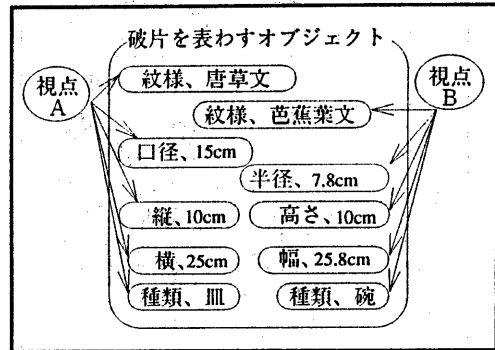


図1：オブジェクトの視点

さらに、共通の性質を持つオブジェクトを一つのグループとして扱うために、複数のオブジェクトからなるバンドルを導入する。バンドルは従来のデータベースのテーブルやクラスに相当する。

以上のように基本DBと導出DBを作成していくことで2章で述べたサイエンティフィックデータベースへの要求を満たすことができる。以降では、基本データ、データエレメント、名前付きエレメント、オブジェクト、バンドルを総称してデータベースエレメントと呼び、それらの作成のことを導出と呼ぶ。導出については3.4節で述べる。図2にデータベースエレメントの相互関係を示す。なお各データエレメントの詳細については後述する。

### 3.2.基本DB

基本DBには、管理したいオブジェクトを表わすイメージデータや音声データ、管理したい対象物から人間が抽出したデータ（対象物の重さ、サイズ、考古学データベースであれば発掘口や、発掘場所など）が未整理の状態に格納される。それらのデータは一見、普遍的なデータであるようにみえるが、例えば重さ、大きさなどは測定した人間やその時の環境によって変化しないとも限らない。そこで、基本DB中のデータにはすべて、そのデータを得た環境や測定条件などを付加しておき、同じ対象物から得られたデータであれば一つの識別子で参照できるようにしておく（図3）。ユーザは一つの識別子で参照できる複数のデータから測定条件をもとに所望するデータを決定する。

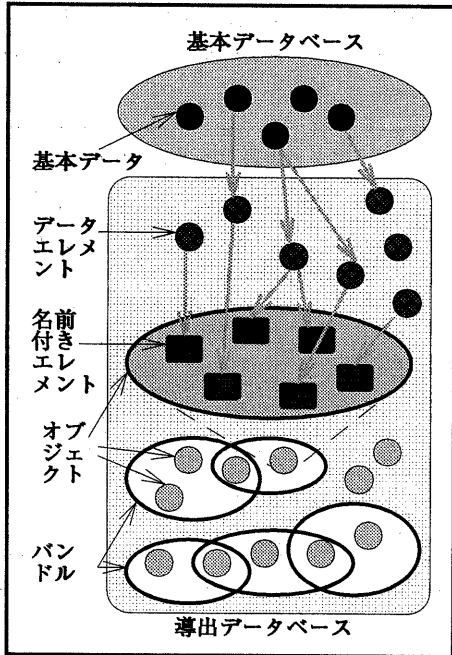


図 2：データベース要素の相互関係

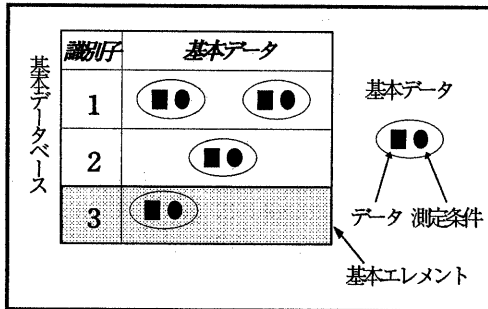


図 3：基本 DB の構造

### 3.3.導出 DB

導出DBは基本DB中の基本データをもとにして、研究者が任意のスキーマを独立して作成するためのデータベースである。ここには各研究者が基本DB中のデータから導出したデータや各々の仮定に基づいたデータが格納される。導出DBは複数の研究者で共有されることが可能であり、また1個以上の導出DBをもとにして新たな導出DBを導出することも可能である。以降では導出DBの構成について述べる。

#### 3.3.1.データエレメントと導出DB

データエレメントは導出DBにおいてデータの実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。

**【定義 1】データエレメント**

$$de = rde \vee vde$$

$rde = (deid, datacell, type, dp)$  : 実データエレメント

$vde = (deid, ddid, deid)$  : 仮想データエレメント

但し、

- deid = データエレメント識別子
- datacell = データセル
- type = データの種類
- dp = 導出過程
- ddid = 導出データベース識別子

#### データエレメント識別子

データエレメントの識別子は導出DB内で一意であり、導出DB単位で独立している。つまり、データエレメントは複数の導出DBにおいて共有されないが、単一導出DB内であれば、データエレメントは複数の名前付きエレメントによって共有される。

#### データセル

データセルはデータエレメントにおけるデータの実体を格納する要素であり、基本DB中の基本データを指す指示エレメントかデータ値とそれらの単位で構成される(図4)。

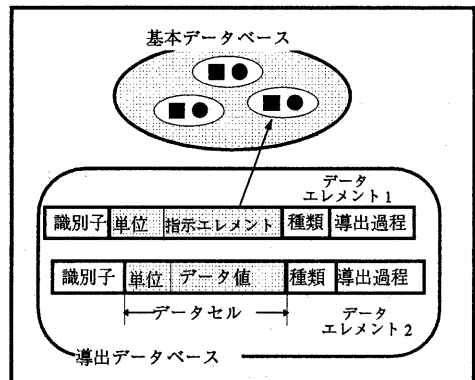


図 4：データエレメントと指示エレメント

データセルが空の場合、データエレメントは実際のデータを格納せず、アクセスされる度に後述の導出過程を用いてデータを作り出す。これは従来の関係データベースにおけるビュー機能に相当する。データセルが空であるデータエレメントを動的データエレメントと呼ぶ。導出過程については3.4節で述べる。

データセル中の単位はそのデータエレメントがデータセルに保持しているデータの単位である。単位と

後述のデータの種類をシステムが管理し、データの演算に利用することによりユーザはデータの単位にかかわる複雑な手続きを意識しなくてよい。

### データの種類

データの種類とはそのデータが長さを表わすものか、または画像データか、といったことである。データの種類の基本的なものとしては、length (長さ)、weight (重さ)、image (静止画)、video (動画)、string (文字列)、numeric (数値) などがあり基本 DB のユーザが独自に定義することもできる。

### 仮想データエレメント

データエレメントが自身の識別子の他に、参照するデータエレメントの識別子とそれが属する導出 DB の識別子から構成されている場合、そのデータエレメントは実体を持たずに他のデータエレメントを参照する仮想データエレメントである。仮想データエレメントは主にデータエレメントを他の導出 DB から参照する場合に用いる。

### 導出 DB

導出 DB はデータエレメントの集合である。定義 2 に導出 DB の定義を示す。

#### [定義 2] 導出 DB

$$DDB = (ddid, ddname, DE)$$

ただし、 $\begin{cases} ddid = \text{導出データベース識別子} \\ ddname = \text{導出データベース名} \\ DE = \{de\} \end{cases}$

導出 DB には、2章で述べたようなサイエンティフィックデータベースに対する要求を満たすために、名前付きエレメント、オブジェクト、バンドルといった要素を含む。これらはデータエレメントやその集合にマッピングされる要素で、データエレメントの保持するデータ値に対して以下の機能を提供するために必要となる。

- 名前の情報を付加する (名前付きエレメント)
- 複数のデータをひとまとめにして一つの対象物の性質を表わす (オブジェクト)
- 似た性質を持つ対象のデータをグループ化する (バンドル)

### 3.3.2.名前付きエレメント

名前付きエレメントはオブジェクトの主要な構成

要素であり、オブジェクトにおいて属性値すなわちデータエレメント、データエレメントの集合、配列、順序集合、そしてオブジェクトを属性名と対応づける。

#### [定義 3] 名前付きエレメント

$$nc = (ncid, name, deid \vee \{deid\} \vee oid)$$

ここで、 $\begin{cases} ncid = \text{名前付きエレメントの識別子} \\ name = \text{名前付きエレメントの名前} \\ deid = \text{データエレメント識別子} \\ oid = \text{オブジェクト識別子} \end{cases}$

その識別子はオブジェクト単位で一意であり、名前付きエレメントは必ず一つのオブジェクトに属さなければならない。これに対して、データエレメントは必ずしも名前付きエレメントに対応しているとは限らず、また複数の名前付きエレメントに対応する場合もある。名前付きエレメントがオブジェクトの識別子を要素として持つ場合、その名前付きエレメントが属するオブジェクトは、オブジェクトの属性値としてオブジェクトを持つ。

### 3.3.3.オブジェクト

オブジェクトは研究者の見方や仮定を反映したデータの単位である。その識別子は導出 DB 内で一意である。オブジェクトの定義を定義 4 に示す。

#### [定義 4] オブジェクト

$$obj = robj \vee vobj$$

$$robj = (oid, NE, BID, PE, dp) : \text{実オブジェクト}$$

$$vobj = (oid, ddid, oid) : \text{仮想オブジェクト}$$

ただし、 $\begin{cases} oid = \text{オブジェクト識別子} \\ NE = \{nc\} \\ BID = \{bid\} \\ PE = \{pc\} \\ dp = \text{導出過程} \\ ddid = \text{導出データベース識別子} \end{cases}$

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。名前付きエレメントの集合は、導出 DB 中のデータエレメントの部分集合にマッピングされる。BID はオブジェクトが属するバンドルの識別子の集合である。オブジェクトは複数のバンドルに同時に属することができる。またどのバンドルにも属さないで存在することもできる。

PE はオブジェクトが持っている視点の集合である。視点はオブジェクトに属する名前付きエレメントの

部分集合を構成するための情報である。視点を通してオブジェクトをみると、オブジェクトに属する名前付きエレメントのうち、その視点に属している名前付きエレメントのみが見える。

オブジェクトが自身の識別子の他に、他のオブジェクトの識別子とそのオブジェクトが属する導出DBの識別子から構成されている場合、そのオブジェクトは実体を持たずに他のオブジェクトへの参照ポイントをもつ仮想オブジェクトである。仮想オブジェクトは主にオブジェクトを他の導出DBから参照する場合に用いる。

### 視点の構成

視点は、オブジェクトの要素である名前付きエレメントの集合の部分集合を構成するための情報を格納する。部分集合を構成する名前付きエレメントはユーザによって指定される。視点の定義は次の通り。

#### 【定義 5】 視点

$$PE = \{pc_n\}$$

$$pc_n = (pc\_name_n, \{neid\})$$

ここで、 $\begin{cases} pc\_name = \text{視点名} \\ neid = \text{視点に属する名前付き} \\ \quad \text{エレメントの識別子} \end{cases}$

### 3.3.4.バンドル

本データモデルのバンドルはオブジェクトの集合である。その定義を定義6に示す。

#### 【定義 6】 バンドル

$$bundle = rbundle \vee vbundle$$

$$rbundle = (bid, ddid, OBJ, RC, dp) : \text{実バンドル}$$

$$vbundle = (bid, ddid, bid) : \text{仮想バンドル}$$

但し、 $\begin{cases} bid = \text{バンドル識別子} \\ ddid = \text{バンドルが属する導出} \\ \quad \text{データベースの識別子} \\ OBJ = \text{バンドルに属する} \\ \quad \text{オブジェクトの集合} \\ RC = \text{バンドルの束縛条件} \\ dp = \text{バンドルの導出過程} \end{cases}$

バンドルは識別子、導出DBの識別子、オブジェクトの集合、束縛条件を要素に持つ。このうちオブジェクトの集合は、導出DBに属するオブジェクトの集合の部分集合であり、その主構成要素はユーザが指定し

た束縛条件を満たすオブジェクトである。束縛条件はバンドルを作成するユーザの見方や仮定により決定される。従って、バンドルはユーザの見方や仮定にマッチしたオブジェクトの集合であるといえる。バンドルの識別子は導出DBで一意である。

しかしながら、どのような見方や仮定にも例外が存在するし、必ずしも的確な条件を指定できるとは限らない。そこで束縛条件を満たしていないオブジェクトをバンドルに無条件に属させることを許す。これにより束縛条件がはっきりしないバンドルでも、とりあえずバンドルを生成し、ある程度オブジェクトの数がそろった段階で、それらのオブジェクトの共通性や性質を考慮した上で改めて正確な条件を定義することが可能になる。

このためにバンドルの束縛条件を、バンドル条件と複数のオブジェクト条件を論理和で結合したものとす。バンドル条件はユーザがその見方や仮定のもとに指定した条件である。オブジェクト条件はバンドル条件は満たさないがバンドルに属するオブジェクトを表わす条件であり、一つのオブジェクト条件は「OID=オブジェクト識別子」という形で表される。ここでOIDはオブジェクトの識別子を表わすタームである。

### 3.3.5.シェイプ

本システムのデータベースはこれまでに述べてきた様な構造をしており、オブジェクトやバンドルは格納しているデータに依存して動的に頻繁に変化する。従って本システムのデータベースは、格納するデータの定義という意味のスキーマを持たない。ここではスキーマの決定よりもデータの格納が先行する。そこで、オブジェクトやバンドル等のデータベースエレメントが構成する導出DBの概要を表わす用語としてシェイプを定義する。

#### 【定義 7】 シェイプ

シェイプは導出DBがどんなデータベースエレメントから構成されているかを示す。

シェイプはデータベースの形を表わすという点ではスキーマと共通点があるが、データベース中のデータに依存するという点でスキーマとは異なる。

## 3.4.導出

導出とは既存のデータベースエレメントを利用し

て新しいデータベースエレメントを導き出すことである。導出には基本DBと導出DB間の**基本導出**と、導出DB内でデータベースエレメントを導出してデータベースの柔軟性を向上させる**データベース内導出**、さらにある導出DBから新たな導出DBを導出するための**データベース間導出**の3種類がある。導出は導出をおこなうための言語である**導出記述言語**で行い、各データベースエレメントの導出過程にはその導出に使用した導出記述言語の構文を格納する。

### 3.4.1.基本導出

基本導出とは、基本データを導出DBで扱えるようにデータエレメントに関係付けを行う操作である。基本導出では指示エレメントを用いる。指示エレメントとは基本データの全体あるいは一部分を指し示す指示子である。例えば文字列データに対する指示エレメントは、文字列の先頭からのオフセットとそのバイト数で構成され(図5)、静止画像に対する指示エレメントは、画像における領域を矩形で表現するための座標である。

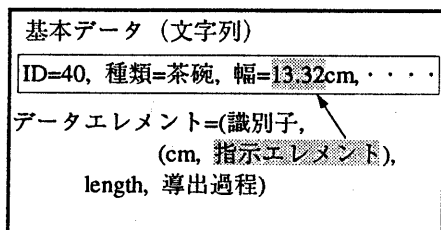


図5:基本導出

### 3.4.2.データベース内導出

データベース内導出とは既存のデータベースエレメントを利用して基本データ以外のデータベースエレメントを新たに作成して導出DBの表現力や柔軟性を向上するための導出である。データベース内導出には次のような導出がある。

#### (1)データエレメント導出

データエレメント導出には新規作成のほかにデータエレメントのコピー、仮想データエレメントの作成、既存のデータエレメントに対する演算によって新しいデータエレメントを導出する方法がある。

#### (2)オブジェクト導出

オブジェクト導出には空オブジェクトの作成と既存のオブジェクトを利用したコピー、仮想オブジェク

トの作成による導出がある。空オブジェクトは名前付きエレメントを持たない。オブジェクトのコピーは全く同じ値を持つオブジェクトを作成するがその要素の実体は別にとられる。

仮想オブジェクトは参照元オブジェクトのオブジェクト識別子を持つオブジェクトでありデータを確保する実体を持たない。これにより導出DBで他の導出DB中のオブジェクトを参照することが可能になる。その他、オブジェクトに名前付きエレメントを追加、削除する導出がある。これは従来のDBMSにおけるスキーマ進化に対応する。

#### (3)バンドル導出

バンドル導出とはバンドルを作成、変更するための導出である。新規に作成されたバンドルには、一つのオブジェクトも属さず束縛条件も空である。

複数のバンドルに演算を施して新しいバンドルを作成した場合、オブジェクトは作成されたバンドルと既存のバンドルの両方に属することになる。この他に既存のバンドルをコピーする方法と既存のバンドルのバンドル識別子を持つ仮想バンドルを生成する方法がある。バンドルの変更は既存のバンドルに対して、オブジェクトの追加や削除、束縛条件の変更を行う。

バンドルの演算とコピーとの違いは、導出したバンドルに属するオブジェクトが、元のバンドルのオブジェクトと同じオブジェクトであるか否かという点である。前者は同じオブジェクトであり、後者は別々のオブジェクトである。

### 3.4.3.データベース間導出

データベース間導出とは、既存の導出DBを利用して新たな導出DBを作成するための導出である。

#### (1)データエレメント導出

データエレメント導出のうち、データベース間導出ではデータエレメントのコピーと仮想データエレメントの作成が利用できる。

#### (2)オブジェクト導出

データベース間導出におけるオブジェクト導出は、オブジェクトのコピーと仮想オブジェクトが利用できる。それ以外は名前付きエレメントの性質により利用できない。

#### (3)バンドル導出

データベース間導出では前節のバンドル導出のう

ち、コピー、仮想バンドルの作成が利用できる。

## 4. 関連する研究

### 4.1. TextLink/Gem

TextLink/Gem[1][2]では半構造化状態のデータベースに格納されている未整理データから以後のデータベース操作において必要となるオブジェクトの同定の実現にアンカーと呼ばれる参照オブジェクトを用いる。またオブジェクトの多様性を表現する手段として視点付きオブジェクトを実現している。しかし、データやスキーマの変更過程は管理していない。

### 4.2. マルチレイヤモデル

マルチレイヤモデル[3]は一つのオブジェクトが状況や役割によって異なる属性値を持つようにするためにレイヤ概念を用いている。ここでは1つのレイヤにひとつの実体の側面を対応させている。レイヤによって同一の側面を持つオブジェクトをまとめることで、実世界の実体の持つ多面性を素直に表現できる。さらに、インスタンス間継承の機構を用いて2つ以上のレイヤを重ね合わせることでさまざまな側面をもつ情報を表現できるが、段階的なオブジェクトの構成やスキーマの変更過程について触れていない。

### 4.3. ファセット

ファセット[4]は永続プログラミング言語上で、一つのオブジェクトが複数の型を持つマルチタイプオブジェクトを実現するための機構である。しかし、ファセットの定義そのものは静的であるため、スキーマが頻繁に変更されるサイエンティフィックデータベースには不向きである。

### 4.4. 代理クラス

代理クラス[5]は、データ知識協調モデルにおいて、コンテキストモジュールと呼ばれるものの多様性を表現するためのものである。コンテキストモジュールとは、独立に存在するデータベース中のデータと知識ベース中の知識の一部をまとめて利用目的に応じたデータと知識の動的な結合を実現するものである。代理クラスは、従来のコピーやビューと異なり、元オブジェクトとは独立しながら必要な場合にのみ元オブジェクトの属性を利用できデータの更新が簡単に実現できる。ここでは半構造化データベース、スキ-

マの変更過程等については触れていない。

## 4.5. ガイアシステム

ガイアシステム[6]がすでに述べた研究と異なるのは、データがどのように作成されたかというデータの由来をメタデータとして管理していることである。しかしながらここではオブジェクトの段階的な同定については触れていない。

## 5. おわりに

以上、本論文ではサイエンティフィックデータベースの要求を考察し、それを満たすようなデータモデルについて述べてきた。今後は、導出において利用するデータベースエレメントの演算、導出記述言語を定義し、様々な種類のデータに対する指示エレメントの構造を設計し、本データモデルを実現するシステムの設計、実装を行う。

## 謝辞

データの収集、分類作業などでお世話になった福井県立一乗谷朝倉氏遺跡資料館の岩田隆氏、水村信行氏に深く感謝いたします。

## 参考文献

- [1]上島紳一 他「歴史的資料を対象としたサイエンティフィックデータベースのシステム設計」, 信学技報, DE93-47 (1993-11).
- [2]銭晴 他「ハイパーテキストデータベースシステム TextLink/Gem におけるオブジェクトとスキーマの動的、段階的な構築機能」, 信学技報, DE92-39 (1993-3).
- [3]奥田太郎 他「インスタンス関係章に基づくオブジェクトモデルとその物理記憶構造」, データベースシステム 91-2 (1991-2).
- [4]有次正義 他「永続プログラミング言語 INADA のマルチタイプオブジェクト」, 情処学アドバンスドデータベースシンポジウム, Tokyo December 8-9 (1992-12).
- [5]彭智勇 他「データ知識協調モデルにおける代理クラスとその応用」, 信学技報, DE93-46 (1993-11).
- [6]Nabil I.Hachem et al. 「Managing Derived Data in Gaea Scientific DBMS」, Proc. of 19th VLDB, pp.1-12 (1993).
- [7]Stanley B. Zdonik et al. 「Incremental Database Systems: Databases from the Ground Up」, Proc. of ACM SIGMOD93, pp.408-412 (1993-5).