

マルチメディア・データベースの枠組み

竹平真則

渡辺豊英

吉田雄二

名古屋大学工学部

マルチメディア・データベースは、従来の数値・文字データに加えて、画像・図形・音声などのデータを、統合的に扱うことを目標に研究されてきた。本稿では、実体をメディア・データとその関連で捉えるという視点で、マルチメディア・データベースの枠組みについて考察した。この考察に基づいて実体と関連を記述子 Block によって定義する方法を検討し、マルチメディア・データベースとして有すべき性質を明確にした。また、Smalltalk-80 を用いて本稿で述べた諸性質をプロトタイプ・システムとして開発した。

A Framework of Multimedia Database

Masanori Takehira Toyohide Watanabe Yuuji Yoshida

Faculty of Engineering, Nagoya University

Furo-cho, Chikusa-ku, Nagoya 464-01, Japan

The study of the multimedia databases has aimed to develop such a database system which can manipulate data of images, graphics, voices as an integrated unit of data, in addition to numerical and string data which have been managed on the traditional database systems.

In our stand point of view, entities and relationships are important notions. We studied constructing a framework of multimedia databases from that point of view. So, this paper describes functional requirements for the data model of multimedia databases and also addresses the implementation issue.

1 はじめに

計算機技術の発展は著しく、記憶媒体の大容量化、演算処理の高速化、入出力装置の多様化、操作環境の高度化などに及び、種々の処理業務を計算機によるデータ処理機能で置き換えてきた。それとともに、従来の処理内容を越えた高度で、多様な情報の扱いを可能にしてきた。また、高性能、多機能を有した入出力装置の開発は、数値・文字データに限らず、図形、画像、音声などのデータも容易に操作可能にしてきた。計算機の応用技術にはこのような様々な形態のデータを効率的に利用できることが、要求されている。

マルチメディア・データベースは従来の数値・文字データに限らず、図形・画像・音声などのデータも同一の操作インタフェースの下に統合的に扱うことを目標として研究されてきた。[1][2]しかし、マルチメディア・データベースの枠組みが必ずしも現在明確になっている訳ではない。本稿では、実世界に観察される実体から数値、文字、図形、画像など、様々な形態で捉えられる特徴を効果的に表現し、利用可能にする表現構造をマルチメディア・データベースの構造とし、その枠組みについて検討する。

2 マルチメディア・データベースへの諸要求

様々な形態のデータを有機的に関連づけ、種々の検索操作、更新操作に対処できる表現構造と処理機能がマルチメディア・データベースに必要である。マルチメディア・データベースに対する検討の視点を明確にするために、初めに諸要求を整理する。

1. マルチメディア・データベースは実世界の実体の現象について様々なメディアを介して観察された情報を的確に関係付けて表現されたデータベースである。
2. データベースとして表現された種々の形態データは同一実体に対する種々の側面に関する意味情報を表している。従って、各データは相互に関連付けられている必要がある。
3. 様々な形態データは相互に補完できなければならない。また、データベースに関する意味の一貫性がなければならない。
4. データの形態・構造が異なっても、データベースの操作においては同一のインタフェース環境を実現できなければならない。すなわち、メディアの種類を意識せずに、統一的に操作できなければならない。

これらの視点は実体に対して、メディア種別によらずに、その特徴・特性の現れの違いとして表現され、本質的な意味に変わりないことを要請する。また、異種メディアを介したデータ表現に対しても同一のユーザ・インタフェース、操作ビューが実現されなければならないことを要請している。図1に、概念的な視点を表した。

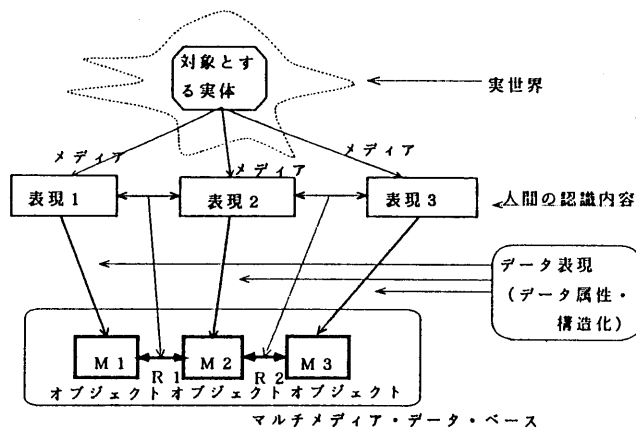


図1：マルチメディア・データベースの概念図

マルチメディア・データベースの実現に際しては、種々のメディア形態に制約されることなく、形態データ間の有機的な関係が保持される構造を採用する必要がある。すなわち、一つのメディア表現の構造化だけでなく、メディア表現相互間の構造化を図ると同時に、構造化データ間の関係を直接的に、また間接的に解釈できる機構が必要不可欠である。図1では、個々のメディア表現をデータベースとしてどのように表現するかという課題に加えて、これらメディア表現が有する相互間の情報の統合的な構造化機構に、マルチメディア・データベースの課題があることを表している。統合的な構造の下に、有機的に構成された各形態データが、関連を介して陽に相互の関係を保存できる。

3 実体・関連の表現

実世界の現象を実体と関連において捉える試みは、データベースに限らず、対象世界をモデリングする方法として近年有効なアプローチとして認識されている。例えば、実体関連（ER）モデルなどの意味データモデルは従来の古典的なデータモデルと比べて、データ間の意味構造を陽に捉えることができる。また、オブジェクト指向パラダイムも実体を直接的に捉える方法として注目されている。マルチメディア・データベースに要請される機構に対して、実体と関連によるモデル手法は少なからずその要求を満たす。[3][4][5] すなわち、我々の立場においては様々な形態データの表現構造として各データ間に対して意味情報を付帯させ、各データの相互関係を解釈できることが必要不可欠である。ここで、意味情報とは、データベースを構成する各要素が相互にどのような内容・役割・関係を有しているかに関する情報である。

3.1 実体の表現

実体はメディアを介してその特徴・特性が抽出される。これら個々に構造化されたメディア・データをオブジェクトと呼ぶ。従って、同じメディアを介しても実体を捉える視点が異なれば、抽出される特徴・特性によって、様々なオブジェクトが生成され、また異なるメディアを介せば異なるオブジェクトが生成される。しかし、これらのオブジェクトは対象とする実体が同一である限り、同じ実体の種々の側面として共通のオブジェクト特性を示し、オブジェクト群を構成する。

個々のオブジェクトは記述子 Block を用いて表現され、一般的な記述形式は次のようである。

```
Block < ブロック名 > :
    AKindOf < ブロック名 >;
    APartOf < ブロック名 >;
    ComposedOf < ブロック名 >, < ブロック名 > .. < ブロック名 >;

    Data
        < 変数名 > : < データ型 > (< ブロック名 >);
        < 変数名 > : < データ型 > (< ブロック名 >);
        . . . . .
        < 変数名 > : < データ型 > (< ブロック名 >);

    end;

    Method
        < 手続き名 > : { < 文 > };
        < 手続き名 > : { < 文 > };
        . . . . .
        < 手続き名 > : { < 文 > };

    end;

    Condition
        < 式 >; < 式 >; . . . . ; < 式 >

    end;

end;
```

Blockでは、関連 AKindOf,APartOf,ComposedOf などによって他の Block との相互関係を指示する。一方、Data 句は Block の内部変数を宣言し、Method 句はこの Block に付帯する手続きを定義する。Condition 句では、この Block に対するアクセス要求を条件付けて制御する。すなわち、Block では個々のオブジェクトの内部構造を Data 句で、そして、その機能を Method 句で定義し、さらに Block のアクセス条件を Condition 句で与えている。

3.2 関連の表現

オブジェクト相互の関連は、Block 内で指示される。オブジェクト相互の上位・下位関係以外に、オブジェクトのクラス関係などが想定される。

1. AKindOf 関係: あるオブジェクトが、他のオブジェクトの特殊化されたものであるときに対応する。例えば、図 2 (a) にあるように、オブジェクト Amp が AudioProducts の一種であれば、
 AKindOf AudioProducts
 と表現する。
 2. APartOf 関係: あるオブジェクトが他のオブジェクトの一部として構成されるときに対応する。図 2 (b) に示すように、Compo は CD, Tuner, Speaker, Amp から構成されていて、CD, Tuner, Speaker, Amp は、Compo に対して APartOf 関係になる。例としてオブジェクト CD の内部において、
 APartOf Compo;
 と、指示する。
 3. ComposedOf 関係: APartOf 関係の反対の関係を表す。例えば、Compo というオブジェクトは、
 ComposedOf CD,Tuner,Speaker,Amp;
 と指示できる。
- 1～3 までは、各関係はオブジェクト各関係は、オブジェクトの性質に関して静的な関係であり、さらにオブジェクト相互の動的な関係が必要である。
4. ReferTo 関係: 異なったオブジェクトが相互に関連する関係を有することを表す。例えば、絵本における文章と挿絵のような関係が対応する。Statement に対して、Picture は次のようである。

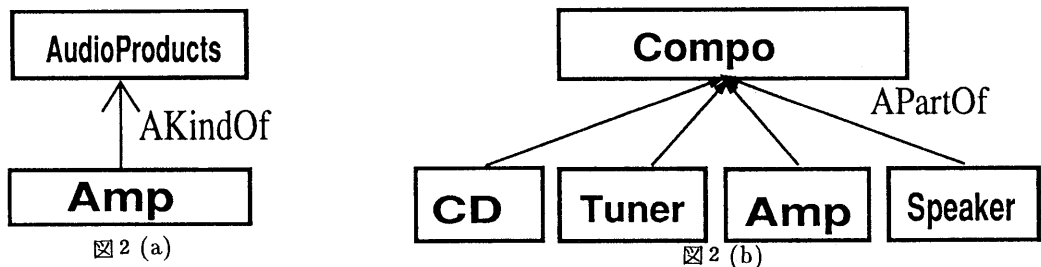


図 2: (a)(b) オーディオ製品オブジェクトの関係

ReferTo Picture;

この関係は、一方が検索・更新されると、必ず参照される方も検索・更新に関係することを表し、オブジェクトの存在従属性を表している。

5. 前後関係 (Before,After): あるオブジェクトが他のオブジェクトに対して位置的に前にあるか後ろにあるかを表す。例えば、上であげた絵本の場合、絵や文の順序が意味があり、Page2 が Page1 の後に続けば、

After page1

となる。

4 データベース表現の例

オーディオ製品のカタログを例に、そこで記載されている個々のオブジェクトを以上の枠組みに従って検討する。第3章に基づきオーディオ製品を捉えて、オブジェクトの構成関係を示したのが図3である。

例えば、このカタログで最上位クラスのオブジェクト AudioProducts をオブジェクト記述の Block を用いて定義すれば次のようになる。ここで、real, integer, string と同様に、image, graphic は基本データ型とする。

Block AudioProduct:

Data

```
price:integer;
name :string;
number:string;
size: record x:integer;
        y:integer;
        z:integer;
```

end;

```
photo:image;
maker:string;
```

end;

end;

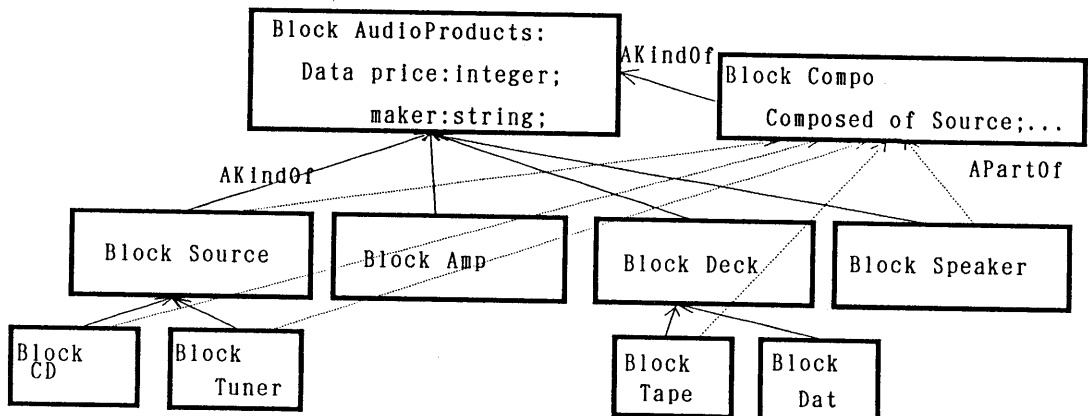


図4：オーディオ製品オブジェクト構成関係

AudioProducts はオーディオ製品カタログ・データすべての最上位クラスであるために、特に関連などは与えず、その製品的特性のみが定義されている。

次に、この AudioProducts の下で特化されたオブジェクトの一部を定義する。例えば、図 2 (a) で示した Amp は以下のようになる。

```
Block Amp:
  AKindOf AudioProducts;
  Data
    f-curve   :graphic;
    d-range   :real;
    gain      :real;
    output    :real;
    inputform :real;
  end;

  Method
    zoughukiritu:
      begin
        temp z;          (* ローカル変数の宣言 *)
        z:=exp( -gain/10);
        return(z);
      end;
  end;
end;
```

以下、同様に定義される。

さらに図 2 (b) に示した Compo を考える。Compo は CD, Tuner, Amp, Speaker の組合せから構成されている。例えば、ここではブロック「Compo」が、価格の問い合わせに対してどのように反応するか考える。

また、CD の出力の形式と Amp の入力の形式が、一致していなければならないことを Condition 句として記述する。

```
Block Compo;
  ConstructOf: CD, Tuner, Amp, Speaker ;
  Data
    Photo: image
  end;

  methods
    price:
      begin
        price := ( CD price ) + ( Amp price )
               + ( Tuner price ) + ( Speaker price )
      end;
  end;

  Condition
    Amp.inputform = CD.outputform
  end;
end;
```

5 データ操作

実体・関連の表現構造の下に構成されたデータベース構造に対して、代表的なデータ操作命令を示す。[6]

1. 検索 retrieve : 例えば、「コンボの中で、ある CD を用いているか、あるメーカー製のスピーカーを用いているものを探し、その写真を示す」とき、

```
retrive photo
from Compo,CD,Speaker
wherex CD.name = "DCD-1100" or Speaker.maker = "JBL"
```

などと記述する。

2. 修正 modify : 例えば、「2万円以上の CD を一律1割引にする」などは、

```
modify CD.price = CD.price * 0.9
from CD
where price > 20000
```

のようになる。

3. 削除 Delete : 「型番"DCD-1100"の CD は生産中止なので削除する」とき、

```
delete
from CD
where CD.name = "DCD-1100"
```

4. 追加 input: 「CD に新しいデータを追加する」とすると、

```
input CD
do name = "DCD-1100";
   price = 39800 ;
   ..... end.
```

と、できるようにする。

さらに、従来のデータベースに備えられている機能、キーワードによるパターン・マッチングによる直接検索などに加えて、データの構造を利用した機構が必要である。それは、Block の階層性を利用しての検索、Block 間の関連を利用しての"意味"検索などが必要である。

6 システム構築への考察

以上に述べたマルチメディア・データベースの枠組みに基づいて、プロトタイプ・システムを Smalltalk-80 によって試作した。Smalltalk-80 はオブジェクト指向言語であり、オブジェクト・クラスの階層関係 AkindOf をそのまま実現できる。すなわち、オブジェクトは、自身の中で定義されていないメッセージを受信したとき、クラスの階層性をたどって、親となるクラスのメソッドに定義されていれば、処理される。

一方、APartOf は、すでに定義されているオブジェクトを構成要素として、新しいオブジェクトを定義するために、メッセージの制御に工夫が必要である。ComposedOf 関係は次のようなクラスを生成し、その下で実現した。まず、Compo を次のように定義する。

```
Object subclass #Compo
  instanceVariableNames: 'source amp speaker'
  classVariableNames:''
  poolDictionaries:''
  category:'MMDBtest'!
```

ここで、source,amp,speaker は、インスタンス変数である。

もし、コンポの値段が、それを構成するオブジェクトの値段の合計であるとして、コンポの値段を返すmethodをpriceとすると、

```
!Compo methodsFor:'accessing'!
price
↑( source price ) + ( amp price ) + ( speaker price ) !
```

とすることで、メッセージを適宜整理・分配することにした。これによって、ComposedOf関係を表現することを試みた。

7 おわりに

本稿では、マルチメディア・データベースの枠組みについて検討した。まだ、十分に表現上の問題、操作機能の種類、及び管理機構とアクセス制御について検討していないが、概略の枠組みは決定された。少なくとも、実体・関連の把握の下にマルチメディア・データベースを捉えることが可能であることが、明らかとなった。

今後、さらに検討しなければならない課題も多い。例えば、カタログを題材にして検討したが、データ間の関係は非常に静的である。より動的な性質のある実体も、問題にしなければならない。

また、マルチメディア・データベースにおいては、ユーザ・インターフェースの問題と密接な関係があるが、ここではほとんど検討されていない。

謝辞

日頃から御指導いただき、名古屋大学工学部 杉江昇教授、稲垣康善教授、鳥脇純一郎教授、並びに中京大学情報科学部 福村晃夫教授に深く感謝するとともに、研究室の皆様に感謝します。

参考文献

- [1] 増永：マルチメディアデータベース総論：情報処理,vol.28, No.6 ,pp.671-684(1987).
- [2] 小島,植村：マルチメディアデータベースのためのデータモデリング：情報処理,Vol.28,No.6 ,pp.685-693(1987).
- [3]Katumi Tanaka ,Masatoshi Yoshikawa ,Kozo Ishihara:Schema Virtualization in Object-Oriented Databases:Proc. of 4th Data Eng. pp23-30 .
- [4] 田口,坂下：OAシステムと文書データベース：情報処理,vol.28, No.6 ,pp.721-729(1987).
- [5]Michael Schrefl,Eric J. Neuhold: Object class definition by generalization using upward inheritance:Proc. of 4th Data Eng. pp4-13.
- [6]Jay Banerjee,Won Kim,Kyung-Chang kim:QUARIES IN OBJECT-ORIENTED DATABASES Proc. of 4th Data Eng. pp31-38.