

## 空間的・時間的オブジェクトの包含関係に基づく継承機構と その物理的記憶構造

大本 英徹  
神戸大学大学院自然科学研究科

奥田 太郎 田中 克己  
神戸大学工学部

オブジェクト指向データベースにおける型/クラス階層は、オブジェクトの集合的操作や組織的屬性構造定義などに大きな役割を果たしている。本稿では、空間的・時間的広がりを持つオブジェクトの包含関係に注目し、従来のクラス階層に基づくものとは異なった、属性の半順序関係による属性及びその値の継承機構について述べる。この機構はクラスビューと呼ばれ、属性値間に包含関係などの半順序関係が存在する場合にインスタンス間で属性を継承し、オブジェクト自身の評価値を仮想化するものである。ここでは、この機構の基本的定義について述べ、さらにその実現として、我々の開発したビデオデータベースシステムとこの半順序関係継承機構を取り入れた空間的オブジェクトの物理的記憶構造と地図データベースへの応用について述べる。

## An Inheritance Mechanism Based on Inclusion Relationship among Spatial/Temporal Objects and Its Physical Storage Structure

Eitetsu Oomoto  
Graduate School of Science and Technology,  
Kobe University

Taro Okuda and Katsumi Tanaka  
Faculty of Engineering,  
Kobe University

In this paper, we focus on the inclusion relationships among time/spatial objects, and introduce an inheritance mechanism based on the relationships. This inheritance does not depend on a class hierarchy, but a partial order relationship between attribute values. If there exists a partial order, for example, inclusion relationship, between values of some attribute of instances, the attributes and their values are inherited from larger object to smaller one. first, the basic idea of this mechanism will be described in this paper. We also describe the video database system, which we have developed, and the example of our inheritance. Also, we will show a physical storage structure for supporting our partial order inheritance, and the application to a geographical database.

## 1 まえがき

次世代データベースシステムとして期待されるオブジェクト指向データベースシステム(OODB)の初期段階における研究開発は一通りの区切りを見せつつあり、いくつかの商用 OODBMS システムを用いた応用システムも開発されつつある。一方、データベース研究の立場からは、さらなる OODB の発展を目指し、新たな研究段階へ入りつつあると言える。現在、OODB において未だ決定的ではないものの、その特徴・概念の大まかな合意が形成されつつあり[1]、次のようなものが挙げられる。

- オブジェクト識別性
- 複合オブジェクト
- データと手続きの一体化
- 型/クラス
- 型/クラス階層と継承

これらの諸概念は非常に有用であるが、本稿では特に継承(インヘリタンス)機構に着目し、クラス階層ではなくインスタンス間の半順序関係に基づく継承機構を提案し、その基本的概念について述べる。

さらにその継承機構のビデオデータベースにおける実現、最後にこの継承メカニズムをとりいれた物理的記憶構造とその地図データベースへの応用について述べる。

## 2 基本概念

ここでは時間的・空間的広がりを持つオブジェクトを管理するデータベースを考える。時間的広がりを持つオブジェクトとは、例えばビデオ映像や音声などがあり、空間的オブジェクトとは地理データベースにおける公園や工場等の領域が考えられる。

まず最初に議論を進めるうえで以下のような仮定を置く。

- 全属性値の集合を  $\mathcal{D}$  とする。
- 全てのオブジェクト識別子の集合  $ID$ 。
- 全属性名集合  $\mathcal{A}$ 。
- 全クラス集合を  $\mathcal{C}$  とする。

クラス階層とは有向非巡回グラフ  $T = (V, E)$  で表わす。ここで  $V$  はクラスに対応する節点集合、 $E$  は is-a 有向枝の集合である。ここでいうクラス階層は既存のオブジェクト指向データベースが提供するクラス階層に相当し、オブジェクトの属性

構造の継承関係を定義するものとする。さらにあるクラス  $C$  に属する全インスタンスの集合を  $instance(C)$  とし、クラス定義中の全属性集合を  $attr(C)$  とする。

### 定義 1 オブジェクト

オブジェクトとは次のような対  $o = (i, v)$  である。

- $i$  はオブジェクト識別子(oid)である<sup>1</sup>。すなわち  $i \in ID$ 。
- $v$  は  $n$ -組  $[A_1: v_1, \dots, A_n: v_n]$  は値である。各ラベル  $A_1, \dots, A_n$  は属性名であり、すなわち  $A_1, \dots, A_n \in \mathcal{A}$  である。各属性値  $v_i$  は以下のように再帰的に定義される。
  1. 記号  $nil$  は値である。空値を表わす。
  2.  $x \in \mathcal{D}$  なる任意の要素  $x$  は値である。
  3. 任意のオブジェクト  $o$  は値である<sup>2</sup>。
  4. 値の集合  $\{v_1, \dots, v_n\}$  は値であり、これを集合値(set value)と呼ぶ。
  5. 値の  $n$ -組  $[A'_1: v'_1, \dots, A'_n: v'_n]$  は値であり、これを組値(tuple value)と呼ぶ。

ここで任意のオブジェクト

$$o = (i, [A_1: v_1, \dots, A_n: v_n])$$

に対し、 $o$  の有する全属性の集合  $\{A_1, \dots, A_n\}$  を  $attr(o)$  と表記する。各属性値はドット記法を用いて  $o.A_i = v_i$  と表わし、さらに全ての可能なオブジェクトの集合を  $\mathcal{O} = \{o_1, \dots, o_n\}$  と表記する。

### 例 1

例えば都市地図データベースにおいて、図 1 のような 2 次元矩形オブジェクトの例が考えられる。

$o_1 = (i_1, v_1)$ ,  
 $v_1 = [$  名前: 神戸市,  
人口: 1400000,  
市外局番: 078,  
領域: [ Left: 10, Top: 15,  
Right: 100, Bottom: 50] ]

$o_2 = (i_2, v_2)$ ,  
 $v_2 = [$  名前: 灘区,  
領域: [ Left: 60, Top: 30,  
Right: 90, Bottom: 45] ]

<sup>1</sup>本論文では、議論を簡単にするために  $v$  を組値に限定する。

<sup>2</sup>O<sub>2</sub>[3] のデータモデルでは oid 自身を値としているが、本論文では、第 2 節後半で述べる種々の操作の定義の簡化のため、このような定義としている。

この例では市クラスのインスタンスである神戸市オブジェクトと、区クラスのインスタンスである灘区オブジェクトを考える。灘区オブジェクトは神戸市オブジェクトに地理的に包含されている。ここで神戸市の市外局番は078であるが、それは灘区においても同様である。市外局番の更新問題を考えると、“市外局番”属性を各オブジェクトが独自に持ち値を複製して持つことは、あまり合理的でない。むしろ神戸市オブジェクトのみに属性及び属性値を持たせておき、灘区オブジェクトはそれを共有ないしは神戸市オブジェクトから継承するほうが望ましいと考える。

複数のオブジェクト間で属性値を共有する方法として、例えば SmallTalk-80 ではクラス変数を用いる方法がある。しかし区クラスを市クラスのサブクラスとしなければならず、市クラスのインスタンスとして大阪市オブジェクトと神戸市オブジェクトが存在する場合は、大阪市の市外局番は06であり神戸市は078であるから、市クラスにクラス変数“市外局番”を定義したとしてもうまく表現できない。ここではオブジェクトの包含関係に基づき、あるオブジェクトへ、それを包含するようなオブジェクトからの継承ないしは情報伝搬が必要である。

さらにこの例ではオブジェクトは2次元的であったがオブジェクトの包含関係は一般に  $n$  次元領域について考えることができる。領域を表わす値が  $m$ -組で表現できるならば、その  $m$ -組値を属性値とする属性をオブジェクトに定義することで、オブジェクト間の包含関係のある属性の値の大小関係、正確には半順序関係に帰着できる。ここでオブジェクト領域を決定する属性が、必ずしも空間的座標値である必要は無いことに注意を要する。すなわち各ドメイン中の要素間に半順序関係が定義可能であるような任意の属性に対して“領域”を定義可能である。任意の属性について半順序関係が定義可能であれば、その関係に基づいた情報の継承が可能であると考えられる。

## 定義2 オブジェクト集合上の半順序関係

オブジェクト集合  $O$  上の半順序関係を以下のように定義する。

- 属性  $A$  のとりうる値の集合（定義域）上にある半順序関係  $\preceq$  が定義されているとする。 $O$  内の任意のオブジェクト  $o_1, o_2$  に対して、 $o_1.A \preceq o_2.A$  ならば、 $o_1 \preceq_A o_2$  と表記する。

## 例2

例1において、“領域”属性に次のような半順序

関係を定義する。

```

For   $v_1 = [ \text{Left: } x_1, \text{Top: } y_1, \text{Right: } z_1, \text{Bottom: } w_1 ],$ 
       $v_2 = [ \text{Left: } x_2, \text{Top: } y_2, \text{Right: } z_2, \text{Bottom: } w_2 ],$ 
IF   $x_2 \leq x_1$  and  $y_2 \leq y_1$  and
       $z_1 \leq z_2$  and  $w_1 \leq w_2$ 
THEN
       $v_1 \preceq v_2$ 

```

関係、[Left: 60, Top: 30, Right: 90, Bottom: 45]  $\preceq$  [Left: 10, Top: 15, Right: 100, Bottom: 50] が成り立つので、オブジェクト  $o_1, o_2$  には  $o_2 \preceq$  領域  $o_1$  なる関係が存在する。この場合、この  $\preceq$  領域とは、矩形オブジェクト間の包含関係となる。

我々は、半順序関係の一つとして一次元・二次元的広がりを持つオブジェクトの包含関係について、これに基づく継承問題に特に注目する。このような包含関係に基づく継承問題に対し、包含関係を一般化したオブジェクト間の半順序関係に基づく継承機構として、クラスビュー (Class View) と呼ぶ概念を導入する。

## 定義4 クラスビュー

ある属性  $A$  上に半順序  $\preceq_A$  が定義されているとき、クラスビュー  $V$  とは以下の4つ組  $V = (T, Attr, S, \preceq_A)$  である。ここで  $T \in C$  はターゲットクラス、 $Attr = \{A_1, \dots, A_n\}$  は指定された属性集合で、このような属性をここでは継承可能属性 (Inheritable Attribute) と呼ぶ。 $S$  はソースクラス集合  $\{C_1, C_2, \dots, C_n\}$ 、 $\preceq_A$  は属性  $A$  上に定義された半順序関係である。

ソースクラス集合  $S = \{C_i\} (1 \leq i \leq n)$  内の任意のクラス  $C_i$  に属するインスタンス  $o_i \in instance(C_i)$  をソースインスタンスと呼ぶ。

クラスビュー  $V$  とは直観的には半順序関係  $\preceq_A$  によって、ターゲットクラス  $T$  のインスタンスからソースクラス群のインスタンスへ情報を継承させて、オブジェクトを仮想的に見せる機構であると言える。

## 定義5 クラスビューインスタンス

クラスビュー  $V = (T, Attr, S, \preceq_A)$  上において、ターゲットクラス  $T$  の任意のインスタンス  $o$  に対し、クラスビュー  $V$  のインスタンス、すなわち、 $V$  のクラスビューインスタンスの集合を  $instance(V)$  と表わすと、 $instance(T)$  から  $instance(V)$  の上への1対1写像  $f$  が存在する。 $f$  は以下の条件を満足する。

- $f(o) = o'$  とすると  $o$  と  $o'$  のオブジェクト識別子は等しい。

2.  $A \in \text{attr}(o)$  なる任意の属性  $A$  に対して,  
 $o'.A = o.A$ .
3.  $A \notin \text{attr}(o)$  なる  $\text{attr}(C_i) \cap \text{Attr}(C_j)$  ( $C_i$  はソースクラス) 内の属性  $A$  に対して, クラスビューインスタンス  $o' \in \text{instance}(V)$  は以下のよう属性値定義を持つ.

$$o'.A = \max_{\leq A_i} \{o''.A_i \mid o'' \text{ は } S \text{ 内のある } C_j \text{ のインスタンスでかつ } o \leq_A o'' \text{ なる } \leq_A \text{ 上での極小なオブジェクト} \}$$

ここで  $\max_{\leq A_i} \{ \}$  とは, 属性  $A$  の定義域の任意の部分集合から半順序関係  $\leq_{A_i}$  において極大な要素を取り出す操作である.

### 例 3

例 1 を再び考える. この時, 市クラスに定義された属性 “市外局番” は, 市クラスのインスタンスの内側にある区クラスインスタンスに継承される. しかし, 人口属性や名前属性は継承されない. 区クラスのクラスビュー  $V_1$  は, 例えば次のように書ける.

$$V_1 = (\text{区}, \{ \text{市外局番} \}, \{ \text{市} \}, \leq_{\text{領域}})$$

クラスビューとはあるオブジェクト (インスタンス) が, 別のオブジェクトから属性とその値を継承する様式を定義するものである. ターゲットクラスとは, 属性を継承する側のインスタンスの属するクラスを指定する. この場合, 区クラスのインスタンスに対して継承が起こる. それに対しソースクラスとは属性を継承させる側のクラスを決定する. この場合, 市クラスのインスタンスより, 継承させるべき属性と値が提供される.

また, 全ての属性が継承されるわけではない. 継承して意味がある属性とそうでない属性の2つのタイプがあると我々は考える. 先の例では市外局番属性は継承されるべきだが, 人口属性は継承したとしても全く無意味である. 継承可能属性とは継承しても意味のある属性の指定となる.

これら3つを決定することで, 無制限にインスタンス間の属性継承が起こらないように適当な制御を行なうことが可能になると考える. またインスタンス自身が独自に持つ属性値が存在すれば継承は起こらず, そちらが優先される. 例えば “名前” 属性が継承可能属性であったとしても, “神戸市” は灘区オブジェクトには継承されない.

ここで問題になるのが, 継承すべき属性が同時に複数のインスタンスにあった場合である. インスタンス  $o$  に対して2つの, インスタンス  $o_1, o_2$  から属性  $A$  が継承される場合を考える. 図 2 のように  $o_2$  が  $o_1$  に完全に包含される場合,  $o_2$  から属性値  $a_2$  が継承されるとしている. また  $o_1, o_2$  に互いに包含関係 (すなわち半順序関係) にない場合 (図 3), 属性値集合  $I = \{o_1.A, o_2.A\}$  中の,

半順序関係  $\leq_A$  における極大な値をとるとしている. 但し, この部分の定義はアプリケーションに依存する要因が大きいと考えられ,

$$o.A = \text{inherit}(I)$$

として  $\text{inherit}()$  をユーザ定義の関数にする (例えば,  $\text{inherit}(I) = \text{nil}$  とする) ことも考えられる.

このクラスビューによる半順序関係に基づく継承の中でも特に空間的オブジェクトの包含関係に基づくものを, 本稿では包含インヘリタンスと呼んでいる.

また属性構造や属性値の継承以外に制約 (の継承) 問題を考える. 都市地図データベースにおいて地域  $A$  及び地域  $B$  が存在すると考える. 各地域内の家屋の建蔽率はそれぞれ 60% 以下及び 70% 以下でなければならないと仮定する (図 4). このような制約は, 従来, スキーマ情報の一部でありスキーマに付随して定義されるものであった. 例えば図 5 のようなクラス住宅地を考え, そのクラス定義の一部としてインスタンスが満足すべき条件を記述する.

この方法で建蔽率制約を表現することは不可能ではない. それに対し, 我々はそれぞれの住宅地もオブジェクトと捉え, 建蔽率制約は住宅地  $A, B$  それぞれのインスタンスが有する一種の情報, すなわち属性値であると考え. 住宅地オブジェクトと家屋オブジェクトの地理的包含関係によって継承された建蔽率属性の値を基に起動する, 制約メカニズムを用意すれば, このような制約も十分に我々の半順序関係継承により表現できると考える.

### 定義 5 クラスビューインスタンスの選択操作

クラスビューインスタンスの集合を  $I$  に対するオブジェクト選択操作 (Selection) とは, 以下のよう,  $I$  から, その部分集合への写像である.

$$\sigma_p(I) = \{o \mid o \in I, p(o) = \text{true}\}$$

ここで  $p$  はオブジェクトの属性に関する述語である.

例えば図 6 のような例で  $I = \{o_1, o_2, o_3, o_4\}$  とする. この場合,

$$\sigma_{\text{市外局番}='078'}(I) = \{o_1, o_2\}$$

となる.

選択演算の特殊な例として, ある領域,

$$r = [\text{top}:r_1, \text{left}:r_2, \text{bottom}:r_3, \text{right}:r_4]$$

に関して,  $r$  に包含される (contain),  $r$  を包含する (overlap),  $r$  と重複がある (intersect) 2次元矩形オブジェクトを選択するような操作が考えられる.

### 定義 6 オブジェクト射影操作

クラスビューインスタンス  $o \in \text{instance}(V)$  が,

属性  $A$  及び属性値  $a$  に対し  $a \leq o.A$  である時、 $o$  の  $A = a$  上への射影操作 (Projection) は、 $o$  からクラスビューインスタンス  $o'$  への写像  $\pi_{A=a}(o)$  であり、

$$o' = \pi_{A=a}(o)$$

は以下のように定義される。

- $o'.A = a$
- $o'.A' = o.A' \ (A' \text{ は } A' \neq A \text{ なる継承可能属性})$

#### 定義 7 オブジェクト合成操作

クラスビューインスタンス  $o_1$  と  $o_2$  の属性  $A$  上での合成 (Merge)  $o$  は、以下のような条件を満足するクラスビューインスタンスである。

- $o.A = o_1.A \cup o_2.A$
- $o.A' = o_1.A' \sqcap o_2.A' \ (A' \neq A \text{ かつ } A' \in \text{Attr}(o_1) \cap \text{Attr}(o_2))$

ここで  $o.A = o_1.A \cup o_2.A$  とは、属性  $A$  の値が時間区間や矩形領域の場合、それらを点集合と考えた集合和とする。また  $o_1.A' \sqcap o_2.A'$  とは、 $x \leq_{A'} o_1.A'$  かつ  $x \leq_{A'} o_2.A'$  を満たす極小の値  $x$  をとる演算であるとする。

### 3 包含インヘリタンス機構のビデオデータベースにおける実現

一次元的広がりを持つデータとして音声やビデオ映像等がある。我々はビデオディスク上に格納されたビデオ映像を時間的広がりを持つオブジェクトと捉えて記述する、プロトタイプのビデオデータベースシステム OVID を開発している。ここでは OVID におけるデータモデルとシステム機能について述べる。

#### 3.1 ビデオオブジェクト

我々はビデオディスクなどに格納されたビデオ映像の任意の部分を、独立に認識・記述できる存在と考え、これをビデオオブジェクトと呼ぶ。ビデオオブジェクトとはオブジェクト識別子とビデオフレーム区間集合、及び、その意味を記述する組値の 3 つ組から成る。例えば、ビデオオブジェクトの例として、図 7 のような例を考える。この例では、ビデオシーン  $i_8$  は田中角栄首相に関するビデオ映像であり、 $I_{14}$  は、三木武夫首相に関する映像群である。また  $i_9$  は特に田中首相が自宅で寛いでいる部分であり、 $i_{15}$  は三木首相の散歩である

とする。オブジェクトの有する意味は各属性の属性値として記述されるが、ビデオオブジェクトそのものを属性値として入れることにより、従来の文字列などでは表現困難なオブジェクトの意味をビデオ映像そのもので表現できる。

#### 3.2 インターバル包含インヘリタンス

ビデオデータベースを実際に構築することを考えると、その労力は多大なものとなることが予想される。なぜなら基本的にユーザが全てのビデオシーンを見てオブジェクトとして記述してゆくしかないと考えられるからである。そこで OVID システムでは、オブジェクト間で共有できる情報は共有・継承して差分的に記述すれば、個々のオブジェクトが情報を継承し詳細な記述となってゆく機構として、ビデオフレーム区間集合の包含関係に基づくインターバル包含インヘリタンスを導入している。これは第 2 節で述べたクラスビューによる属性継承に対応する。

ここではビデオフレーム区間集合属性に対し、包含関係すなわち半順序関係を考えている。我々のビデオオブジェクトモデルでは、クラス階層は存在せず、唯一 VideoObject クラスのみがあるとす。継承可能な属性の集合を  $\{a_1, a_2, \dots, a_n\}$  とすると、クラスビューは次のように与えられる。

$$V_{\text{video}} = ( \text{VideoObject}, \{a_1, a_2, \dots, a_n\}, \{ \text{VideoObject} \}, \leq_{\text{videoFrame}} )$$

インターバル包含インヘリタンスの例を図 8 に示す。 $i_1$  は佐藤栄作首相に関するビデオ映像であり、 $i_3$  はその一部であり、彼がアメリカを訪問した際の映像記録ビデオである。 $i_6$  を特に佐藤首相が声明発表を行なっているシーンであるビデオオブジェクトとして記述するとすると、この映像も、やはり佐藤氏のアメリカでの映像であることに変わりはない。そこで Prime.Minister と Location 属性を継承可能と指定することで、それら属性と属性値がオブジェクト  $o_6$  に継承され、 $o_6$  を evaluate した際にあたかもそれらの属性を  $o_6$  が有しているかのように見える。

#### 3.3 ビデオオブジェクトの操作

我々のビデオオブジェクトモデルにおいて、原子値は汎化関連により階層構造化されており、これに基づく set や tuple 値を含めた属性値間の汎化関連を定義している。この下にビデオオブジェクトの操作として、Interval Projection, Merge, Overlap の 3 つの操作を定義・実装している。

- Interval Projection: あるビデオオブジェクトの一部を切り出し、別のオブジェクトとする操作 (図 9)。
- Merge: 2つのオブジェクトを結合する操作である。直観的にはビデオ映像の編集作業に相当する操作である (図 10)。
- Overlap: 2つのビデオオブジェクトの互いに重複する部分を切り出す操作である。

### 3.4 OVID システムの機能

現在 OVID システムには VideoChart 及び VideoSQL の 2つのサブシステムが実装されている。

- VideoChart ビデオデータベースのブラウジング及びビデオオブジェクトに対する各操作を行なうためのチャート型視覚的インターフェイスである (図 11)。各チャートがそれぞれビデオオブジェクトに対応しており、これらの属性閲覧や再生、さらに Merge, Overlap 操作が行なえる。
- VideoSQL ビデオオブジェクト検索のための検索言語 (図 12)。属性名と属性値を指定することで、その条件を満足するオブジェクトが検索される。この時、オブジェクトの評価には、包含インヘリタンスが作用しており、たとえ自身に属性定義が存在しなくとも、それを含むようなオブジェクトに適当な定義が存在した場合は検索結果に含まれる。

## 4 包含継承機構のための物理的記憶構造とその地理 DB への応用

本節では、第 2 章で述べた概念を、主に地理データベースへ応用することを考慮した物理的記憶構造について述べる。なお、この地理データベースに関しては、領域オブジェクトの半順序関係として領域の包含関係を採用しているので、この章に関する限り、半順序関係を包含関係として扱うことにする。

### 4.1 多次元インデックス $R^+$ -木

本研究においてそのデータの格納構造としては、索引構造  $R^+$ -木 [4] を使用した。この  $R^+$ -木は、一定の広がりを持つような、いわゆる領域オブジェクトの格納機構として、Timos Sellis らが 1987 年に提案したものであり、その構成は B-木のそれと非常に似通っている。以下、その構成を簡単に説明する。

- 頂点 (非葉節点) と葉 (葉節点) で構成される木構造である。各頂点はそれぞれ複数の枝によって子節点と接続している。一番上の頂点を特に根 (root) と呼ぶ。
- 各頂点は枝と枝の境界を示すキー値と、次の頂点を示すポインタの対を持つ。葉には実際のデータのキーと、その値が入る。
- 各頂点が持ち得る最大の子節点の数は、どの頂点でも同じである。各頂点が持ち得る最小の子節点の数は、1つである。
- 木の高さ (レベル) はどの節点も同じである。
- キーは多次元 (の領域) である。検索は、まずある領域を与え、根から始めて、だんだん下のレベルへ領域を絞りこむように下りていく手順である。

本研究では、特にキーが 2 次元 (すなわち、長方形オブジェクトの左上と右下の座標のペア) の場合を考察した。この場合の  $R^+$ -木の構成を図 13 に示す。

### 4.2 オブジェクト間の包含関係に基づく継承を考慮したクラス編成

本節では、2 章で述べた包含関係に基づく継承の概念を実現するためのクラス構成図を図 14 に示し、その詳細を以下に説明する。なお、これらのクラスは OODBMS Versant[5] 上で実現したものであり、現在も稼働中である。

1. gObject クラス  
gObject クラスは、スーパークラスである。すなわち、ユーザはこのクラスの下にアプリケーションのためのクラス階層を構築する。また、このクラスには、包含関係に基づく継承に必要な属性情報を管理する属性を設けた。この属性には、文字列型の属性名とオブジェクト (抽象) 型の属性値のペアのリストが入る。ユーザが新しい領域オブジェクト  $o$  を生成すると、属性情報を管理する属性がクラス階層によって継承され、その値として、 $o$  の全ての属性名と属性値のペアのリストが格納される。
2. Rectan クラス  
gObject クラスのサブクラスである。ユーザはこのクラスの下に、領域を属性として持つクラスをサブクラスとして、定義する (領域を属性として持たないクラスは直接 gObject クラスのサブクラスとして定義する)。
3. InheritCltn クラス  
このクラスのインスタンスは、それぞれ独

立したクラスビューを表現する。ソースクラスとその属性名、さらにターゲットクラス名のリストを1つのオブジェクトとして生成する。例えば、InheritCltn クラスの1つのオブジェクトとして  $i$  というものがあり、その値が、

ソースクラス名:  $C_1$ , 継承可能属性名:  $A_1$   
 ターゲットクラス名:  $\{C_2, C_3\}$

とする。この時、クラス  $C_1$  の1つのインスタンス  $o_1$  に対し、クラス  $C_2$  のインスタンス  $o_2$  とクラス  $C_3$  のインスタンス  $o_3$  が存在して、

$o_2 \preceq_{region} o_1$ ,  
 $o_3 \preceq_{region} o_1$

なる関係が成立すれば、属性  $A_1$  はインスタンス  $o_2, o_3$  に継承されることになる。すなわちこのインスタンスによって、

$V_1 = (C_2, \{A_1\}, \{C_1\}, \preceq_{region})$ ,  
 $V_2 = (C_3, \{A_1\}, \{C_1\}, \preceq_{region})$

という2つのクラスビューが形成されたことになる。

4. List クラス  
 $R^+$ -木の最小構成単位である。1つの List インスタンスは、1つの領域オブジェクトをポインタで格納する。葉節点の要素に当たる。
5. Node クラス  
 $R^+$ -木で、節点を実現するクラスである。1つの Node インスタンスには、List クラスのインスタンスのリストが格納される。木構造への探索や挿入、節点分割などの主なオペレーションをインプリメントする。
6. Rtree クラス  
 Node クラスのインスタンスのうち、その根をポインタで管理する。さらに、データベース内に複数の木構造を存在させることを可能にするため、tree-id をその属性として持つ。

#### 4.3 包含関係に基づく属性継承の実現

$R^+$ -木を用いた包含関係に基づく継承機構を以下に示す。ここでは、クラス  $C_2$  の1つのインスタンス  $o_2$  に対して行なった場合を想定する。

1.  $o_2.Region$  を探索キーとして、 $R^+$ -木に探索メッセージを送る。
2.  $R^+$ -木の根を始点として、 $o_2.Region$  と交差するような、あるいはそれを覆うようなキーを持つ節点へと下っていく。

3. 葉節点まで下り、 $o_2.Region$  を包含するようなキーを持つ節点の要素が見つければ、そのクラス名（ここでは仮に  $F$  とする）を得て、step 4 へ行く。もし見つからなければ、終了。
4. InheritCltn クラスのインスタンス（クラスビュー）のうち、ソースクラスの値が  $F$ 、かつターゲットクラスの値が  $C_2$  であるものがあるかどうか調べる。あれば、step 5 へ行く。もしなければ見つけてきた  $F$  クラスのインスタンスを除外して、step 1 へ行く。
5. クラスビューをもとにして、 $F$  クラスの継承可能な属性を調べ、その値を  $o_2$  に継承して終了。

#### 4.4 ユーザ・インタフェイスの実現

本システムを実現するにあたり、簡単なユーザ・インタフェイスを製作した。このインタフェイスではデータの読み込みや、簡単な質問検索を可能にしている。

例として、当大学当学部の4階をモデルとした。このアプリケーションでは、1つ1つの部屋を1つのオブジェクトとし、その位置（座標）によって、 $R^+$ -木を構築している。そのクラス階層を図15に、検索結果を図16に示す。この例では、Chair（講座）クラスのsupervisorという属性が、Study（研究室）クラスのインスタンスに継承されている。

#### 4.5 問題点

本システムには、なお改善すべき部分が幾つかあると思われる。以下にその問題点を列挙する。

- 継承可能属性の複数継承  
 このシステムでは、5.3節でみたように、1つの継承可能属性が見つかった時点でそのメカニズムを終了してしまう。しかし、2つ以上の継承可能属性を持つ場合も対処できるようにするべきである。
- クラスに未定義の属性の追加と削除  
 本システムでは、特定のオブジェクトに対してクラス定義に現れない属性及び属性値の付加も可能となっている。元来、スキーマをあらかじめ設計しにくいデータベースへの応用等には有用であると考えられ、今後本論文で述べたモデルとの関連を議論する必要がある。

### 5 結 論

本稿では、属性の半順序関係に基づく継承機構を提案し、その基本的概念について述べた。従来

のクラス階層に基づく継承機構では困難な、クラスビューと呼ぶインスタンス間の属性及び属性値の継承機構について、未だ不十分ながら基本的定義を行ない、さらにいくつかの基本的操作についても考察した。

また、その具体例として我々の開発したビデオデータベースにおけるデータモデルとシステムについて簡単に述べた。さらに、半順序関係の特別な場合として、空間的オブジェクトの包含関係を考え、それに基づく継承を考慮した  $R^+$ -木を基本とする物理的記憶構造とその実現についても述べた。

我々の提案した継承機構は、クラス階層における継承と相反するものではなく、相互に直交する概念であると考えるが、その性質など、さらに詳しい検討が今後の課題である。

## 参考文献

- [1] Atkinson, M. et al., *The Object-Oriented Database System Manifesto*, The first International Conference On Deductive and Object-Oriented Databases (DOOD89), pp. 40-57 Dec. 1989.
- [2] 大本, 田中, *A Video Object Model and Its Application to a Video Database System*, 信学技報, DE-90-28, pp.33-40 (1990)
- [3] Deux, O. et al., *The Story of O<sub>2</sub>*, Trans. on Knowledge and Data Engineering, Vol. 2, No. 1, pp. 91-108 ( March 1990 ).
- [4] Timos, S. Nick, R. Christos F., *The R<sup>+</sup>-Tree: A Dynamic Index for Multi-Dimensional Objects*, Proc. of The International Conference on Very Large Databases, Brighton, England 1987
- [5] Versant Object Technology Corp., *VERSANT System Manual*, Sept. 1990.

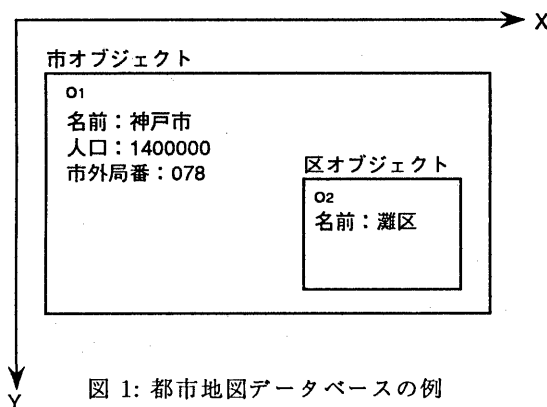


図 1: 都市地図データベースの例

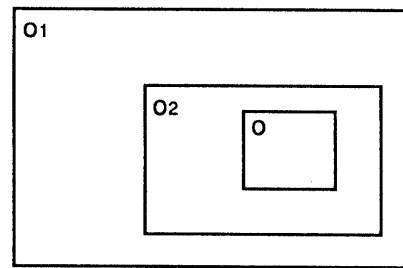


図 2: 複数オブジェクトの包含関係 1

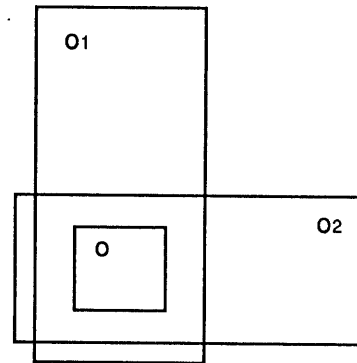


図 3: 複数オブジェクトの包含関係 2

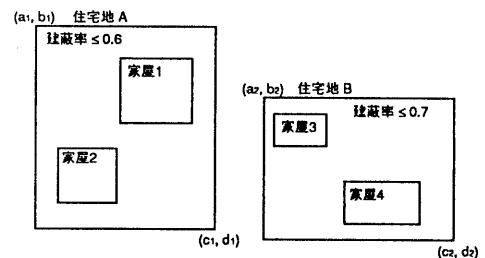


図 4: 建蔽率制約の例

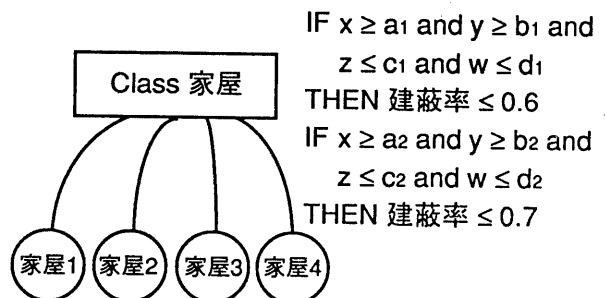


図 5: クラス定義による制約

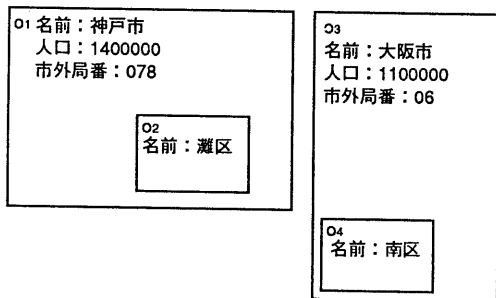


図 6

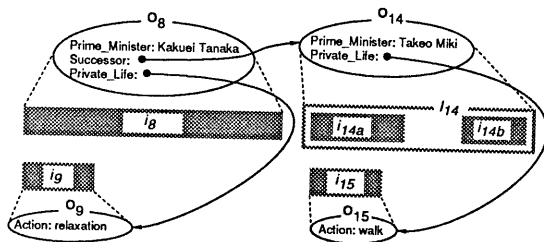


図 7: ビデオオブジェクト

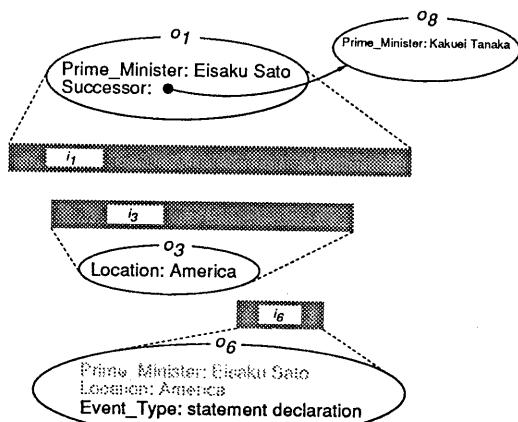


図 8: インターバル包含インヘリタンス

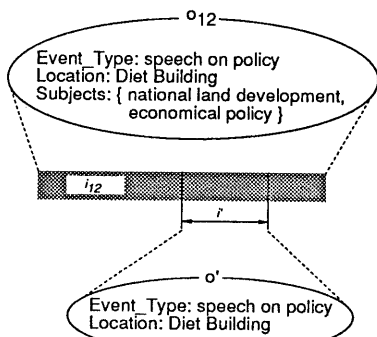


図 9: Interval Projection

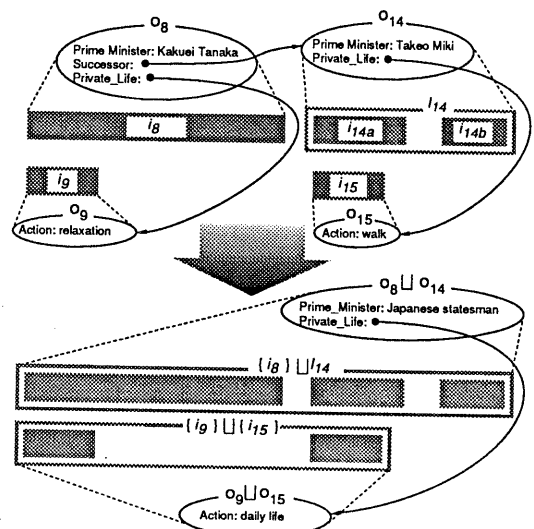


図 10: Merge 操作

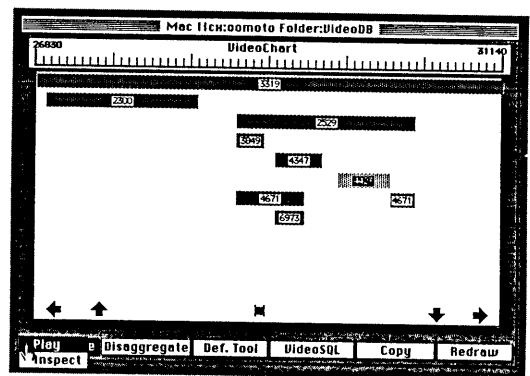


図 11: VideoChart

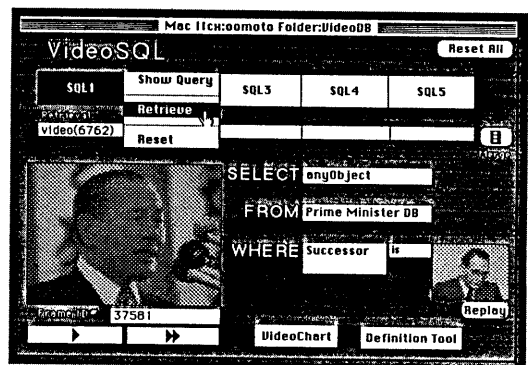


図 12: VideoSQL

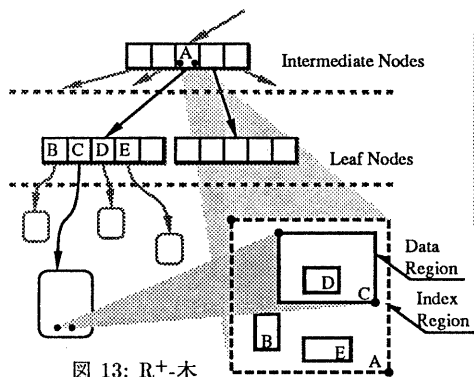


图 13: R+-木

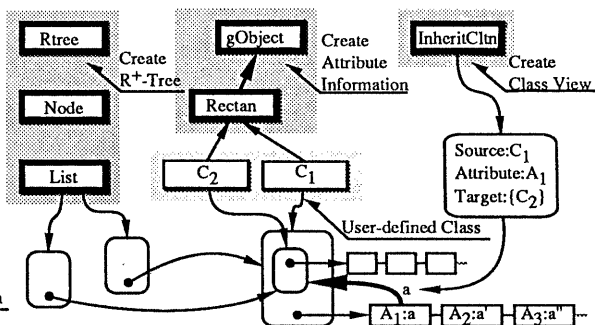


图 14: Class Overview

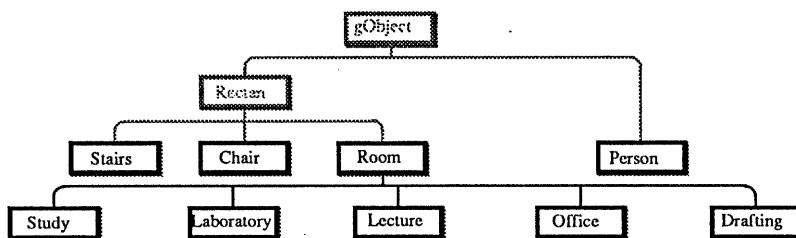


图 15: Application Class

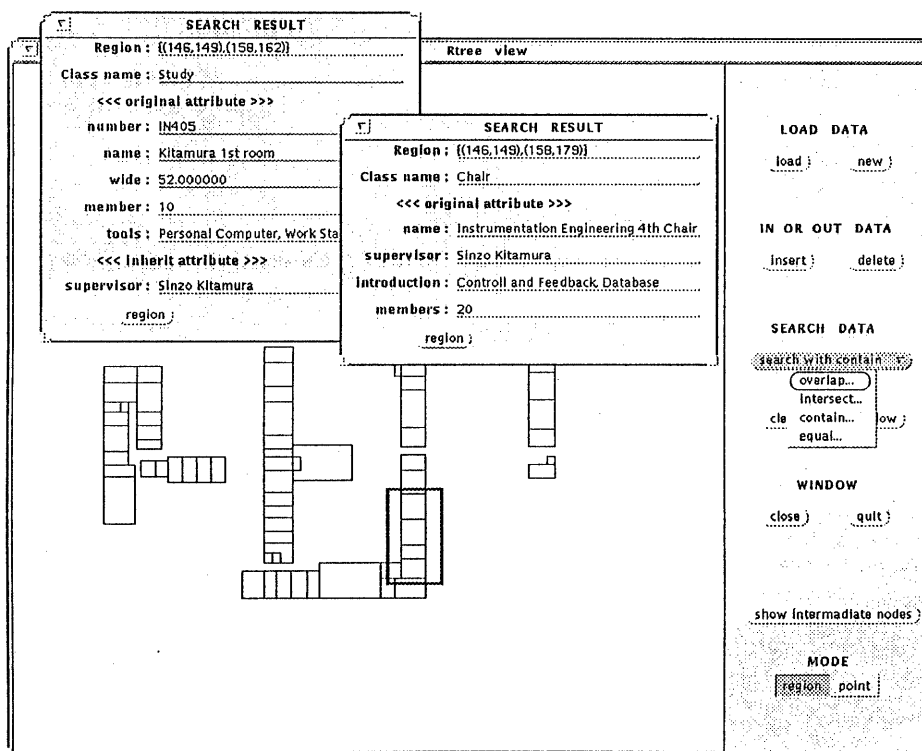


图 16: 検索結果