

伝播ビュー：オブジェクト指向データベースにおける 半順序関係に基づくインスタンス間継承機構

大本 英 徹[†] 田 中 克 己^{††}

近年、オブジェクト指向データベース管理システム (OODBMS) の基本的機能の大まかな合意が形成されつつあり、また多くの OODBMS が商用化されている。既存の OODBMS の提供する諸機能のうち、継承機構はデータベースの論理的構造記述として与えられたクラス(型)階層に基づき、オブジェクトの属性構造やメソッドの差分的定義機能を提供する。しかし、継承がクラス間のみで固定され利用者による変更ができない等、いくつかの問題がある。本論文では、クラス階層ではなくインスタンスの属性値間の半順序関係に基づく継承機構として、伝播ビューと呼ぶ機構を提案する。伝播ビュー機構では、属性および属性値の継承がインスタンス間でなされ、その基盤となるインスタンスの意味的関連が固定的でなく、ある属性の定義域上の半順序関係として任意に定義可能である。また継承対象となるインスタンス群や属性群を変更可能であり、実行時に継承すべき属性や属性値を決定し、仮想的に継承された属性と属性値が存在しているかのように見せかけるようになっている。また、伝播ビュー機構の具体的な適用例として、我々の開発したピアオアデータベースシステムと配置図面データベースシステムに関して、その有効性を述べており、さらに、伝播ビューの妥当性に関する議論も行った。

Propagation-View: An Instance-based Inheritance Mechanism by Partial Order Relationships in Object-Oriented Databases

EITETSU OOMOTO[†] and KATSUMI TANAKA^{††}

In this paper, we propose an inheritance mechanism, called "propagation-view," based on partial order relationships among attribute values of objects. In our propagation-view mechanism, some attributes and their values of an object are inherited by other objects in a dynamic and virtual manner, and the semantic relationships among objects, which cause the attribute/value inheritance, are not limited to is-a relationship but any partial order defined over the domain of user-specified attribute. Also, our propagation-view is a kind of virtualization mechanism. The instance, which receives the inherited attributes and their values, does not have them physically, but is virtualized at run time as if they had those attributes and attribute values. We also discuss about the validity of a collection of propagation-view's, and describe the actual systems, which we have developed in order to show the usefulness of the propagation-view mechanism.

1. はじめに

現在、オブジェクト指向データベース管理システム (Object-Oriented Database Management System, OODBMS) の研究開発が活発に行われており、未だ決定的ではないものの、OODBMS の有すべき基本的機能について大まかな合意が形成されつつある¹⁾とともに、多くの OODBMS が商用化されている²⁾。

その中に現れる主要な概念として、オブジェクト識別性、複合オブジェクト、カプセル化、型/クラス、型/クラス階層と継承などがある。ここで、オブジェクト識別性は、実世界の実体を、その属性値とは独立に表現できる能力を表し、複合オブジェクトは集合、組、リストなどの構成子を再帰的に用いて、より複雑な構造を有するオブジェクトを構成したものを意味する。カプセル化はデータと、それに対する操作(メソッド)を一体化できることを表す。また、型やクラス^{*}は、同じ性質(属性構造やメソッド)を有するオブジェクト群の共通特性を集約したものであり、型階層/ク

[†] 京都産業大学工学部情報通信工学科
Department of Information and Communication
Sciences, Faculty of Engineering, Kyoto Sangyo
University

^{††} 神戸大学工学部情報知能工学科
Department of Computer and Systems Engineer-
ing, Faculty of Engineering, Kobe University

* クラスは、通常、型の概念に加えて、オブジェクトの生成や対応するオブジェクト集合の外延的表現等も意味する。

ラス階層は、型やクラス同士を汎化関連 (IS-A 関連, generalization relationship) で結び階層構造としたもので、階層構造中の上位のクラスの有する属性定義やメソッドは下位のクラスに継承 (inheritance) される。

本論文では、これらの諸概念のうち、特に継承機構に注目する。現在の OODBMS では、データベースの論理構造記述としてのスキーマは、クラス (型) 階層を用いて表すものがほとんどであり、これを基盤とする継承機構は、オブジェクトの属性構造の差分的定義や、メソッドの継承によるプログラムコードの共有といった強力な機能を提供する。しかし、OODBMS が提供する強力なモデル化能力に比べて、その継承機構の能力が限定されていることや、多様な分野での OODBMS の適用を考慮すると、現在の OODBMS が提供するクラス階層に基づく継承機構は、以下のような点で不十分であると考えられる。

●インスタンス間の継承機構

継承が上位・下位クラス間のメソッドや属性構造の継承に限定されており、インスタンス間で属性値の継承や伝播をサポートする機能はほとんど提供されていない。

●特定の意味的関連による継承のみのサポート

クラス階層に基づく継承では、クラス間の IS-A 関連のみに基づいており、その他の意味的関連に基づく継承は、考慮されていない。例えば、空間的オブジェクト間の包含関係やクラス外延以外の任意の集合間の包含関係に基づく継承を実現することは一般に容易ではない。

●継承様式が固定的

現在の継承機構は、例えば、ある属性定義は継承するが他のものは継承しないなどという、継承を選択的に行うことが原則的に困難である*。すなわち、継承様式をユーザの必要に応じて修正・変更できる機能が乏しい。

●動的・仮想的な継承機能の欠如

オブジェクトが何を継承するかがスキーマを作成した段階で静的に決定され、それがオブジェクトの有する属性構造やメソッドを決めてしまう。個々のオブジェクトが継承すべき属性構造や属性値が、任意の時点で変更・決定できるような機能は提供されていない。かなり強引な方法として、メソッドを用いた“仮想属性”という方法もあり得るが、スキーマ

やメソッドの再定義・再コンパイルが必要となり、現状の OODBMS では対応が難しい。

本論文では、IS-A 関係によるクラス階層ではなく、インスタンスの属性値間の半順序関係に基づく動的な継承機構を提案し、その基本的概念について述べ、この継承機構の概念に基づいて実際に開発した配置図面データベースシステムやビデオデータベースを具体例として、この概念の有効性について論じる。

本論文で提案する伝播ビュー機構の特徴は以下のようまとめられる。

●インスタンス間で作用する継承機構

属性および属性値の継承が、クラスからクラスでなくインスタンス同士の間で行われる。

●属性値の半順序関係に基づく継承

継承が行われるインスタンス間の意味的関連が固定的でなく、ある属性の属性値間の半順序関係であり、ユーザが任意に定義可能である。

●継承様式を必要に応じて変更可能

継承の対象となるインスタンス群や属性群が固定的でなく、利用者が必要に応じて変更することにより継承の適当な制御が可能である。

●属性と属性値の動的・仮想的継承の実現

継承される属性が、継承する側のインスタンスに実際に作成され属性値が代入されるのではなく、いかなる属性や属性値が継承されるかを実行時に決定し、その時点において仮想的に属性と属性値が存在しているかのごとく見せかける機構である。

●属性の自動的継承によるオブジェクトの差分的記述

オブジェクト間で共有できる情報が、属性値間の半順序関係に基づいて自動的かつ動的に継承され、差分的にオブジェクト記述を行うだけで個々のオブジェクトとしては詳細な記述を得ることが可能。

●検索時におけるスキーマの強い認識が不要

さまざまな属性と属性値がインスタンス間で継承される結果、検索時におけるスキーマの認識はそれほど重要ではなくなる。

以下、第2章では、伝播ビューと呼ぶ、インスタンスの属性値間の半順序関係に基づく動的な継承機構の基本概念について論じる。第3章においては、伝播ビューの概念に基づいて、実際に開発したビデオデータベースシステムおよび配置図面データベースシステムを具体例として、伝播ビュー機構の有効性について述べる。第4章では、伝播ビューの定義の妥当性について議論する。さらに第5章では、関連する研究と我々

* 上位クラスにおける定義を、下位クラスで独自に再定義することは可能。

の提案する継承機構との位置づけについて論じる。第6章はまとめである。

2. 伝播ビュー

本章では、伝播ビューと呼ぶ、インスタンスの属性値の半順序関係に基づく動的な継承機構の基本的概念について論じる。

2.1 例

ここではまず、伝播ビューの概念を直観的に説明するために、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、これに相当する。

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^{*}オブジェクトを次のような二つの組型オブジェクト o_1, o_2 で表現した例を考える。

o_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 =[名前：灘区，
領域：[Left：60， Top：30， Right：90，
Bottom：45]]

この例では、“行政区分単位”一般の属性構造を定義する行政区分単位クラスおよびそのサブクラスとして市クラスと区クラスを仮定する(図2)。この場合、オブジェクト o_1 は、“神戸市”に対応する市クラスのインスタンスということになる。同様に灘区オブジェクト o_2 は区クラスのインスタンスである。

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。市外局番更新の問題を考えると、“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、あまり合理的ではない。むしろ神戸市オブジェクトのみに市外局番属性および属

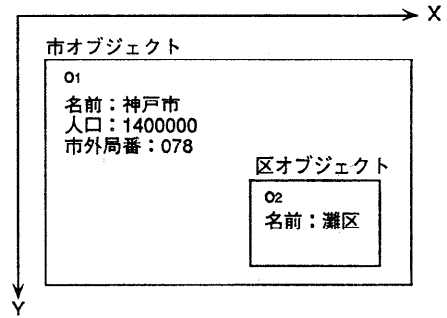


図1 都市地図データベースの例
Fig. 1 An example of a city-map database.

性値を持たせておき、灘区オブジェクトは、それを共有または継承するのが望ましいと考えられる。

ここで、このような継承を行うためには、通常のクラス階層に基づく継承機構では不十分なことに注意を要する。例えば、属性“市外局番”の継承を行うために、区クラスを市クラスのサブクラス、または、区クラスを市クラスのサブクラスとするクラス階層を定義することは不自然である。なぜなら、元来、市クラスと区クラスの間には意味的な関連として IS-A 関連は存在しないためである。

また、複数のインスタンス間で属性値を共有する方法として、例えば SmallTalk-80¹¹⁾ではクラス変数や、C++¹⁰⁾では静的メンバ (static member) を用いる方法がある。しかし、その場合、区クラスを市クラスのサブクラスとせざるを得ないが、これも同様に不自然

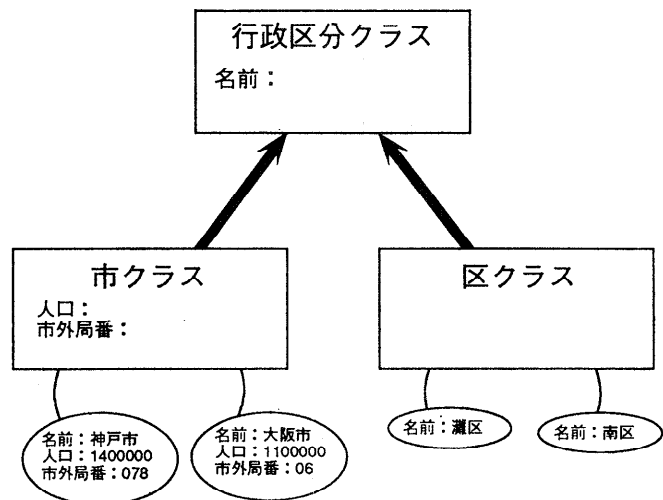


図2 都市地図データベースのクラス階層
Fig. 2 An example of a class hierarchy on a city-map database.

* ここでは議論を簡単にするために、各地図領域を矩形領域で表現する。

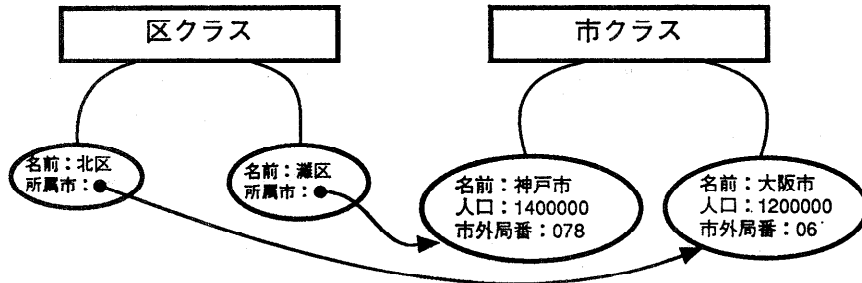


図 3 属性間リンクを用いたオブジェクトの参照
Fig. 3 Object reference by attribute reference.

である。さらに、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、大阪市、神戸市、それぞれの市外局番が同じになってしまい、うまく表現できない。

さらに、図 3 に示すような属性間リンクを用いて継承すべき属性を有するインスタンスを陽に属性値として参照し、ある適当なメソッド定義によって、参照するオブジェクトの“市外局番”属性を仮想属性としてアクセスする方法も考えられる。しかし、その場合、継承関係を定義・変更しようとするにはスキーマの再定義やメソッドの追加等が必要となり、さらに種々の更新操作手続きの中に適切な継承関係を属性リンクとして表現・維持するためのコードを含めておかねばならない。

ここでは、クラス階層やオブジェクトの構造に基づくのではなく、オブジェクトの地理上の二次元的包含関係に基づき、あるオブジェクトに対して、それを包含するようなオブジェクトからの情報継承ないしは伝播機構が必要である。

また、二次元的に包含されるオブジェクトは必ずしも市外局番属性に、それを包含する市オブジェクトと同一の市外局番を属性値として持つとは限らない。例えば、神戸市内に X 町という町オブジェクトが存在して、例外的に X 町内では市外局番は 0789 であるといった場合は十分考えられ、X 町独自の属性値を有することが許されるべきである。すなわち、オブジェクトが属性を継承する以前に、その属性と属性値をすでに有していた場合は、それらが優先されるべきである。

上記の例ではオブジェクトは二次元的であったが、オブジェクトの包含関係は一般に n 次元領域オブジェクト間の包含関係を考えることができる。さらに、単なる包含関係でなく、より一般的な属性値間の包含関係（正確には半順序関係）にまで一般化することができる。また、半順序関係を決定する属性値が、必ずしも空間的座標値である必要はない。空間的包含関係以外の半順序関係に基づく伝播ビュー定義の例として企業の従業員データベースを考える。各プロジェクトグループ毎の平均残業時間の上限は `overtime_avg_limit` 属性で表されるとする。伝播ビュー定義

(Project, {Department}, {overtime_avg_limit}, \subseteq member)

により、各部署内のメンバーで構成されるプロジェクトグループの残業平均時間の上限値は、原則として部署で決められた上限値に一致する（継承する）ということが表現できる（図 4 参照）。ここで、`projectB` のようにプロジェクトグループ独自に上限値を決めていることも考えられ、さらに `projectC` のようにプロジェクトグループは複数部署のメンバーから構成されることも考えられる、これらの場合はいずれの部署に課せられた上限値も継承するべきではない。このような例では継承はプロジェクトに参加しているメンバー

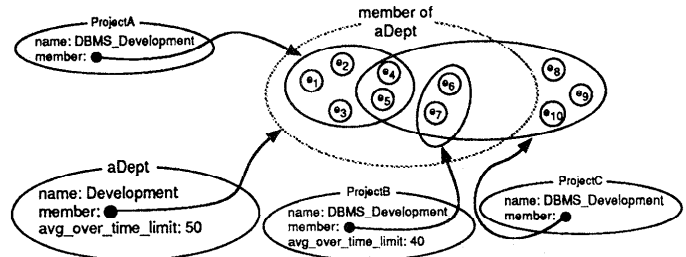


図 4 集合の包含問題に基づく継承
Fig. 4 Inheritance based on set inclusion relationship.

の集合一部分集合の関係に基づいてなされるべきである。

本節では、その定義域中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、このような半順序関係に基づいた情報の伝播機構として、伝播ビューを提案する。

2.2 伝播ビュー定義

以下に形式的な議論を進める上で、次の互いに素な三つの集合の存在を仮定する。

- すべての文字列および数値の集合 \mathcal{D} 。
- すべてのオブジェクト識別子の集合 \mathcal{SD} 。
- データベースに現れるすべての属性名から成る集合 \mathcal{A} 。

本論文においてクラスとは、伝播ビュー機構と直交する概念であり、実世界の実体に対応するオブジェクトの属性構造を定義するものである^{*}。クラス階層は有向木 $G=(V, E)$ で表現され^{**}、 V はクラスに対応する節点の有限集合、 E は IS-A 有向枝の集合である。ここでの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とそれらの継承関係を定義するものとする。

ここでクラス階層 G 中の各クラスを“実クラス”と呼び、実クラス C に対し、クラス集合 $\{C\} \cup \{C' | C \text{ から } C' \text{ への IS-A 有向枝のパスが } G \text{ 中に存在する}\}$ を C^* と表現する。直観的には、 C^* はクラス C のすべての直接および間接の上位クラス（スーパークラス）の集合である。各クラス C に対し、 C において定義される全属性名の集合を $attr(C)$ と表す。また、 $C^* = \{C_1, \dots, C_n\}$ とした時、 $attr(C_1) \cup \dots \cup attr(C_n)$ を $attr(C^*)$ と表す。さらに、クラス C に属する全インスタンス（実インスタンス）の集合を $instance(C)$ と表記し、これを C の外延 (extent) と呼ぶ。

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。すなわち $i \in \mathcal{SD}$ 。
- v は n -組 $[a_1 : v_1, \dots, a_n : v_n]$ である^{***}。各ラベル a_1, \dots, a_n は属性名であり、すなわち $a_1, \dots, a_n \in \mathcal{A}$

^{*} 通常、クラス定義中にはメソッド定義も含まれるが、ここでは議論の簡単化のためオブジェクトが有するメソッドに関しては考えない。

^{**} 議論の簡単化のため、クラス階層を木構造に限定するが、本質的にこのような制限が必要なのではない。

^{***} 本論文では、議論を簡単化するために v を組値に限定する。

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値である。

1. 記号 nil は値である。空値を表す。
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^{*}。
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. 属性と属性値の 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 の有する全属性名の集合 $\{a_1, \dots, a_n\}$ を $attr(o)$ と表記する。この時、オブジェクト o がクラス C に属するインスタンスであるとする、 $attr(o) = attr(C^*)$ となる。各属性値はドット記法を用いて $o.a_i = v_i$ と表す。さらに、すべ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。

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 \mathcal{O} 上の半順序関係を以下のように定義する。

属性 a のとり得る値の集合（定義域）上に、ある半順序関係 \leq が定義されているとする。オブジェクト o_1, o_2 と属性 a に対して、 $o_1.a \leq o_2.a$ ならば、 $o_1 \leq o_2$ と表記する。□

例1

図1の例において、“領域”属性に次のような半順序関係 \leq を定義する。

$$v_1 = [\text{Left} : x_1, \text{Top} : y_1, \text{Right} : z_1, \text{Bottom} : w_1],$$

$$v_2 = [\text{Left} : x_2, \text{Top} : y_2, \text{Right} : z_2, \text{Bottom} : w_2]$$

に対して、

$$x_2 \leq x_1 \text{ and } y_2 \leq y_1 \text{ and } z_1 \leq z_2 \text{ and } w_1 \leq w_2 \text{ ならば, } v_1 \leq v_2.$$

この時、 o_2 領域 $\leq o_1$ 領域、すなわち

$$[\text{Left} : 60, \text{Top} : 30, \text{Right} : 90, \text{Bottom} : 45] \leq$$

$$[\text{Left} : 10, \text{Top} : 15, \text{Right} : 100, \text{Bottom} : 50]$$

が成り立つので、オブジェクト o_1, o_2 には $o_2 \leq$ 領域 o_1 なる関係が存在する。この場合、 \leq 領域 という半順序関係は、矩形オブジェクト間の二次元的な包含関係を表している。

次に、本論文では、このような包含関係に基づく継承を一般化した、オブジェクトの属性値の半順序関係に基づく継承機構として、伝播ビュー (Propagation-

^{*} O_2 のデータモデルでは OID 自身を値としているが¹⁾、本論文では、第2章後半で述べる種々の操作の定義の簡単化のため、このような定義としている。

View)と呼ぶ概念を導入する。これは、2.1節に示したような二次元的な包含関係に基づく継承機構を一般化したものである。

定義3 伝播ビュー定義

クラス階層 $G=(V, E)$ において、伝播ビュー定義 P は以下の四つ組

$$P=(T, A, S, \leq_a)$$

で与えられる。ここで、 T は任意のクラスまたは他の伝播ビュー定義であり、これをターゲットクラスと呼ぶ。 S は G 中の任意のクラス又は伝播ビュー定義の集合 $\{C_1, \dots, C_n\}$ であり、これをソースクラス集合と呼ぶ。 A は $\text{attr}(C_1^*) \cup \dots \cup \text{attr}(C_n^*)$ の部分集合であり、 A 内の属性を継承可能属性と呼ぶ。 \leq_a は $\text{attr}(T^*) \cap \text{attr}(C_1^*) \cap \dots \cap \text{attr}(C_n^*) \ni a$ なる属性 a の定義域上に定義された半順序関係であり、 a を比較属性と呼ぶ。

□

伝播ビュー定義とは、直観的には、あるクラスのインスタンスが他のインスタンスから、属性とその属性値を継続する様式を定義するものである。ターゲットクラスとは、“継承する属性を受け取る側”のインスタンスの属するクラスであり、ソースクラスとは“属性を提供する側”のインスタンスの属するクラスである。ソースクラス集合内のクラス C_i に属するインスタンスをソースインスタンスと呼び、ターゲットクラス T に属するインスタンスをターゲットインスタンスと呼ぶ。また、データベース中に定義されるすべての伝播ビュー定義の集合を pall と表記する。

例えば、伝播ビュー定義 $P=(T, A, S, \leq_a)$ において、あるソースインスタンス o_s と、あるターゲットインスタンス o_t が、それぞれ属性 a を有し、その属性値間に $o_s.a \leq o_t.a$ なる半順序関係が存在すると仮定する。この時、 o_s が有する属性のうち継承可能であるもの、すなわち属性集合 A に含まれる属性について、ソースインスタンスの有する属性と属性値がターゲットインスタンス o_t に継承（伝播）される。その結果、その継承された属性と属性値を、ターゲットインスタンスが有しているかのごとく見せかける機構が、本論文で提案する伝播ビュー機構である。

2.3 伝播ビューインスタンス

定義4 伝播ビューインスタンス

伝播ビュー定義 $P=(T, A, S, \leq_a)$ のターゲットクラス T の任意のインスタンス（ターゲットインスタンス）を $o=(i, v)$ 、 $v=[a_1: v_1, \dots, a_n: v_n]$ とする。伝播ビュー定義 P におけるターゲットインスタンス o に

対する伝播ビューインスタンス o' は以下のように表される。

$$o'=(i, v')$$

$$v'=[a_1: v_1, \dots, a_n: v_n, b_1: u_1, \dots, b_m: u_m]$$

o' は以下の性質を満たす o と同一のオブジェクト識別子を有するオブジェクトである。

1. 属性 b_1, \dots, b_m ($m \geq 0$) は継承可能属性である。すなわち、

$$b_1, \dots, b_m \in A (m \geq 0).$$

2. 属性 b_1, \dots, b_m は o の値 v には含まれない。すなわち、

$$b_1, \dots, b_m \notin \text{attr}(o).$$

3. 属性値 u_j は、各ソースインスタンスの属性 b_j の属性値のうち、極大なものである。すなわち、

$$u_j = \max_{\leq_{b_j}} (\{ \dots, o_s.b_j, \dots \}).$$

ここで $\max_{\leq_{b_j}}()$ は、属性 b_j の定義域の任意の部分集合から半順序関係 \leq_{b_j} に関して極大な要素を取り出す操作である。また o_s は次のような性質を満たす。

- o_s は P におけるソースインスタンスである。すなわち、

$$o_s \in \text{instance}(C), C \in S,$$

また o_s は o と相異なるオブジェクト識別子を有し、さらに o_s は属性 b_j を有する。すなわち、

$$b_j \in \text{attr}(o_s).$$

- o_s は伝播ビュー定義 P におけるすべてのソースインスタンスのうち、 o に対し、半順序関係 \leq_a に関して極小なオブジェクトである。すなわち、 $o \leq_a o_s$ かつ、 $o_s \neq o'_s$ なる任意の $o'_s \in \text{instance}(C)$ 、 $C \in S$ に対して、 $o \leq_a o'_s \leq_a o_s$ は成立しない。 □

ここで、伝播ビュー定義 $P=(T, A, S, \leq_a)$ に対し、 $\text{attr}(P^*)$ とは $\text{attr}(T^*)$ 中のすべての要素と P 上の伝播ビューインスタンスがソースインスタンスから継承しているすべての属性を含む属性集合であるとす。また、 $\text{instance}(P)$ とは、伝播ビュー定義 P に基づいて得られるすべての伝播ビューインスタンスの集合であるとす。

例2

再び、図1の例を仮定する。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に包含される区クラスの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るべきでない。この場合、区クラスをターゲットクラスとする伝播ビュー定義 P_1 は、例えば次のように書ける。

$$P_1=(\text{区}, \{\text{市外局番}\}, \{\text{市}\}, \leq_{\text{領域}})$$

ここで領域とは、“領域”属性の定義域上に定義された半順序関係 \leq に基づいて継承が起きることを表している。この例では \leq は矩形領域の包含関係である。ターゲットクラスとして“区クラス”を指定し、ソースクラスが“市クラス”であることから、区クラスのインスタンスに属性と属性値が渡され、市クラスのインスタンスから、継承させるべき属性と値が提供される。また、すべての属性が継承されるわけではない。こ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無意味である。継承することが妥当である属性、すなわち継承可能属性として、ここでは属性“市外局番”を指定している。ソースクラス、ターゲットクラス、継承可能属性の三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うことが可能になる。

またインスタンス自身が独自に属性と属性値を有すれば継承は起こらず、それらが優先される。すなわち属性値の再定義が可能である。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”という値は灘区オブジェクトには継承されず、独自の属性値である“灘区”という属性値が優先される。

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る場合に、どちらのインスタンスから実際に継承させるかという問題である。インスタンス o に対して、二つのインスタンス o_1, o_2 から、属性 a とその属性値が継承される場合を考える。図5のように o_2 が o_1 に完全に包含される、すなわち包含関係という半順序関係がある場合、我々の伝播ビューインスタンスの定義では、 o を含む極小なインスタンス、すなわち o_2 が選ばれ、 o_2 から属性値 $o_2.a$ が継承されるとしている。また o_1, o_2 が互いに包含関係がない、すなわち半順序関係が成立しない場合(図6)、我々の定義では、属性値の集合 $s = \{o_1.a, o_2.a\}$ 中の、半順序関係 \leq における極大な値をとるとしている。ただし、この部分の定義はアプリケーションに依存す

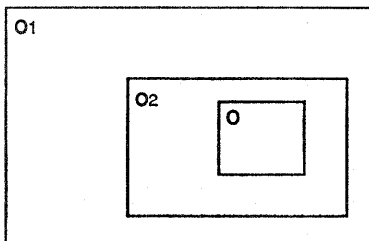


図5 包含関係にある空間的オブジェクトの例
Fig. 5 Inclusion-related spatial objects.

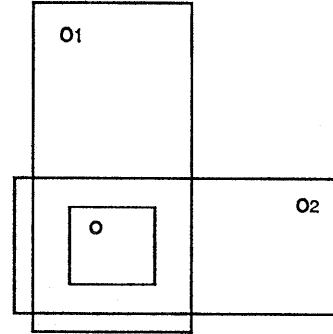


図6 包含関係にない空間的オブジェクトの例
Fig. 6 Non-inclusion-related spatial objects.

る要因が大きいと考えられ、実際には、インスタンス o の属性 a の値は、 $o.a = inherit(s)$ などのように $inherit()$ をユーザ定義の関数にする(例えば、 $inherit(s) = nil$)ことも考えられる。

伝播ビュー定義はクラス階層とは独立に定義され、直観的にはターゲットクラスのインスタンスを仮想化する機構である。しかし、いわゆる仮想クラス^{13), 14)}とは、以下の点で異なる効果を有することに注意を要する。仮想クラスの外延において、各インスタンスは、一般に、仮想クラスの定義に応じた同一の属性構造を有する。それに対し、伝播ビュー定義の外延においては、継承属性として指定された属性を実際に継承するかどうかは、実行時にソース・ターゲット各インスタンスの比較属性の値によって動的に決定され、その結果として、一般にそれぞれのインスタンスごとの継承状態が異なる。つまり属性構造は同一になるとは限らない。

ここで、現在の伝播ビューの定義においては、属性値としてオブジェクトが入っていた場合、その扱いに問題が残されている。例えば次のような例を考える。オブジェクト $o_1 = (i_1, v_1)$, $v_1 = [A_1 : o_1, B_1 : b_1]$ が伝播ビュー機構により、 $o_1' = (i_1, v_1')$ に仮想化されるとする。この時、伝播ビューインスタンスの定義により、 $o_1'.A_1 = o_1$ となり、仮想化されないオブジェクトが属性値となる。しかし、これは $o_1'.A_1 = o_1'$ として、仮想化された状態のオブジェクトを属性値とするのが、より妥当と考えられる。また別の例として、他のオブジェクト $o_3 = (i_3, [B_3 : o_2])$ から属性 B_3 を伝播ビュー機構により継承することにより、オブジェクト $o_2 = (i_2, [A_2 : a_2])$ が o_2' に仮想化されると仮定する。この時、現在の定義では、 $o_2'.B_3 = o_2$ となってしまうが、 $o_2'.B_3 = o_2'$ であるほうが望ましいと考えられる。以上

の問題は単純な問題ではなく、今後の検討課題の一つである。

3. 伝播ビュー機構の実現例

一次元空間的な広がりを持つデータとして音声やビデオ映像等があり、二次元空間的な広がりを持つデータとして地図データベースシステムにおける領域オブジェクトなどがある。これらの空間的オブジェクトを扱うデータベースシステムにおいては、本論文で提案する伝播ビュー機構は、特に有効である。本節では、その具体例として、実際に我々が開発したビデオデータベースシステムおよび配置図面データベースシステムについて述べ、伝播ビュー機構の有効性を示す。

3.1 ビデオデータベースシステム OVID

我々の開発したプロトタイプビデオデータベースシステム OVID^{(3),(4),(5),(6)}では、ビデオディスク上に格納されたビデオ映像の任意の部分に属性・属性値を付与して、ビデオオブジェクトとして記述する。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およびその意味を記述する組値の三つ組から成る。例えば、ビデオオブジェクトの例として、図7のような例が挙げられる。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、三本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シーンであり、 i_{15} は三木首相の散歩シーンであるとする。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。

OVID のビデオオブジェクトモデルでは、ビデオオ

ブジェクトの操作として、Interval Projection, Merge, Overlap の三つの操作を定義している。Interval Projection とは、あるビデオオブジェクトの一部を切り出し、異なるオブジェクトとする操作で、Merge は二つのオブジェクトを結合する操作である。また、Overlap は、二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。これらは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。

また OVID は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型操作を行うためのチャート型視覚的ユーザインターフェイス VideoChart や、ビデオオブジェクト検索のための検索言語 VideoSQL を基本機能として提供している。

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が予想される。基本的にユーザが膨大な量のビデオシーンを実際に見ることにより、意味的にまとまりのある単位ごとにオブジェクトとして認識・記述してゆくことで、データベース構築がなされると考えられるからである。この際、オブジェクト間で共有できる情報は自動的に共有・継承して、差分的に記述するだけで個々のオブジェクトの詳細な記述が得られる機構があれば、その記述量削減に有効である。そこで OVID システムでは、ビデオフレーム区間集合の包含関係に基づく伝播ビュー機構を導入した。

OVID 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している。ここでは、日本の首相に関するビデオ映像データベースを例として取り上げる。継承可能な属性の集合は {Prime_Minister, Location, Event_Type} であり、VideoObject クラスに対する伝播ビュー定義の例として、 P_{video} を次のように定義する。

$$P_{\text{video}} = (\text{VideoObject}, \{ \text{Prime_Minister}, \text{Location} \}, \{ \text{VideoObject} \}, \leq_{\text{videoFrame}})$$

伝播ビュー定義 P_{video} による、ビデオオブジェクトのインスタンス間継承の例を図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_2 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

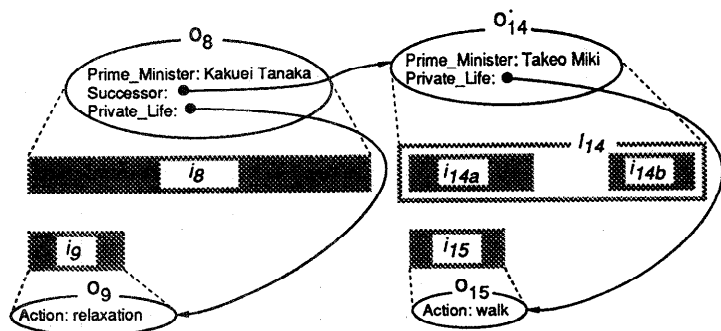


図7 ビデオオブジェクト
Fig. 7 Video objects.

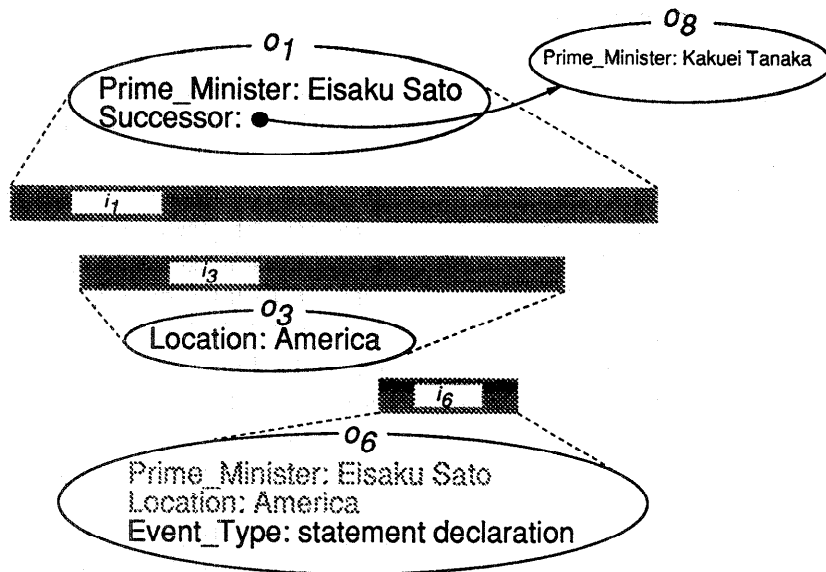


図 8 インターバル包含インヘリタンス
Fig. 8 Interval inclusion inheritance.

が声明発表を行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、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。P_{video}の定義において、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 o₆ に継承され、o₆ の属性を評価した際にあたかもそれらの属性を o₆ が有しているかのように見え、Prime_Minister 属性の値が“Eisaku Sato”、Location 属性の値は“America”であるかのように仮想化される。

伝播ビュー機構をビデオデータベースに取り入れることにより、データベース中の各インスタンスの情報記述量を削減することが可能である。例えば、上記の例で o₁, o₃, o₆ それぞれが、異なるユーザによって記述されたビデオ映像に対するビデオオブジェクト記述であると仮定し、ビデオデータベースに対して「佐藤栄作氏が、アメリカにおいて、演説を行っているビデオオブジェクトを検索せよ。」という条件検索がなされたとする*。この場合、この条件で指定された属性・属性値を、すべて物理的に付与されたビデオオブジェクトは存在しない。しかし、伝播ビュー機構により、インスタンス間で属性・属性値の継承・伝播が行われる結果、それぞれ独立に記述されたオブジェクト

* ただし、どのような検索言語系でどのように表現するかは、簡単のため、ここでは考えない。

群 o₁, o₃, o₆ から、検索結果として o₆ を返すことが可能である。すなわち、差分的にオブジェクト o₁, o₃ および o₆ を記述することにより、全体として、徐々に詳細なビデオデータベース記述を得ることが可能である。

また、データベースに対する検索を行う場合を考えると、ユーザはデータベーススキーマや検索対象となるオブジェクトの属性構造を、一般には、厳密に把握・理解したうえでデータベースに対する各種操作を行う必要がある。それに対し、本論文で提案した伝播ビュー機構を導入することにより、さまざまな属性とその属性値が各オブジェクト間で伝播・継承される結果スキーマの認識はそれほど重要でなくなり、クラス定義やオブジェクトの属性構造をさほど厳密に知らずとも検索を行える利点が生まれる。これはビデオデータベースなどの、あらかじめクラス階層等のスキーマを与えにくいデータベースシステムでは、非常に有効と考えられる。

3.2 配置図面データベースへの適用

ここでは、伝播ビューの配置図面データベースへの適用を述べる。これは、半順序関係として特に二次元領域間の包含関係に着目し、これによってインスタンス間で動的に属性・属性値を伝播させるもので、実際にオブジェクト指向 DBMS Versant¹⁹⁾を用いて実現したものである⁹⁾。

このシステムでは、部屋、研究室、講座などをそれぞれオブジェクトと考え、各オブジェクトは二次元領域の座標のデータを Region 属性の値として持つとともに、その他の必要な属性を有している。そのクラス定義の階層を図 9 に示す（一例として大学学舎の見取り図を考えている）。そして、実際のユーザインターフェースと検索結果を図 10 に示す。この例では、伝

$P=(Study, \{supervisor\}, \{Chair\}, \leq_{Region})$ によって、Chair (講座) クラスの supervisor (総責任者) 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。

また、実際の検索操作としては、overlap (検索領域に含まれる領域オブジェクトすべてを検索)、intersect (検索領域と交差する領域オブジェクトすべてを検索)、contain (検索領域を含む領域オブジェクトすべて

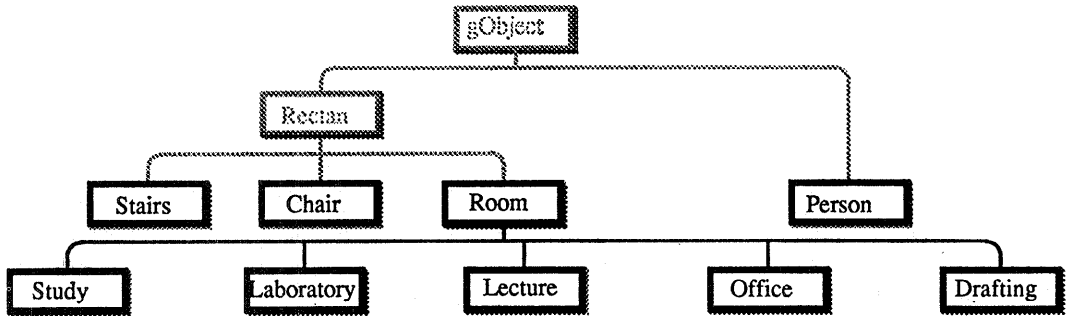


図 9 配置図面データベースシステムのクラス階層
Fig. 9 The class hierarchy of the layout database.

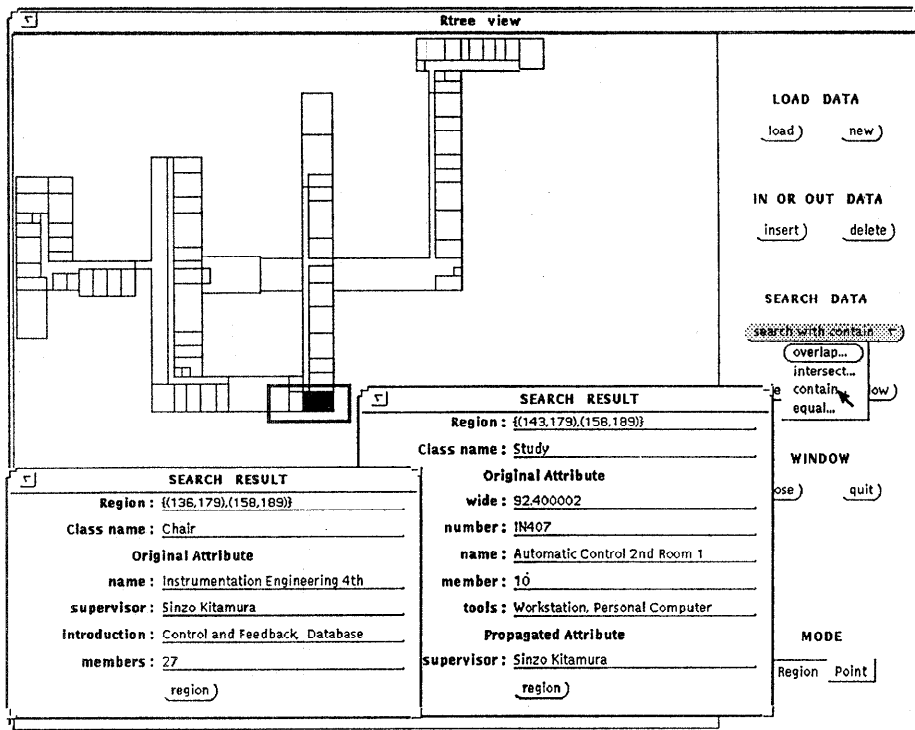


図 10 配置図面データベースの検索結果
Fig. 10 An example retrieval of the layout database.

てを検索), equal (検索領域に一致する領域オブジェクトを検索)がある。

この伝播ビューの配置図面データベースへの適用に対する利点は、以下のようにまとめられる。

●オブジェクト記述作業量の削減

上の例では、“supervisor”という属性を Study クラスと、Chair クラスの両方に持たせようとした場合、両方のクラスでその属性を独自に定義しなければならないが、この伝播ビュー機構の導入によって、Study クラスの属性 supervisor は定義する必要がない*

●データベーススキーマの構造の強い認識が不必要

通常のオブジェクト指向データベースでは、検索質問の生成など各種操作において、クラスの属性やクラス階層情報など、データベースのスキーマに関する知識がある程度必要である。しかし、伝播ビュー機構を導入したシステムでは、そのようなことをあまり考慮しなくても検索が可能である。例えば上の例で、“supervisor”がどのクラスの属性なのか、不明な場合も十分考えられる。しかし例のような伝播ビューが存在することによって、ユーザが、Study クラスのインスタンスに“supervisor”は誰かという質問を問い合わせても、目的の情報を得ることができる。

上で述べた例は比較的単純なものであったが、大規模かつ複雑な地図データベースの場合、各オブジェクトのレイヤー構成に基づく継承属性の多重化も考慮しなければならない。

例えばある地図データベースにおいて、工業地帯や商業地帯などの情報を扱う利用区分レイヤーと、県、市や町などの情報を扱う行政区分レイヤーがあったとする(もちろん両レイヤーは、意味的に互いに独立である)。さらに、地図上の各オブジェクトの領域は、“領域”属性の値として表されるとする。このとき、

1. ある工場地帯内の工場は、その工業地帯の名前を継承する。
2. ある市内の工場は、その市の市外局番を継承する。

という条件があったとき、このようなレイヤー構造に基づく継承は二つの伝播ビュー定義

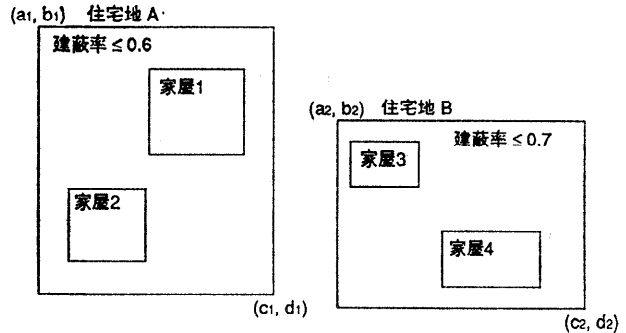


図 11 建蔽率制約
Fig. 11 An example of a coverage constraint.

$$P_1 = (\text{工場}, \{\text{工業地帯名}\}, \{\text{工業地帯}\}, \leq \text{領域})$$

$$P_2 = (P_1, \{\text{市外局番}\}, \{\text{市}\}, \leq \text{領域})$$

として表現することができる。このようにして、一つのインスタンスが、異なるレイヤーから複数の情報を得ている場合にも、この伝播ビューは適用可能である。

伝播ビュー機構は、属性構造や属性値の継承以外に制約(の継承)問題にも適用可能である。都市地図データベースにおいて地域Aおよび地域Bが存在するとする。各地域内の家屋の建蔽率はそれぞれ60%以下および70%以下でなければならないと仮定する(図11)。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12のような家屋クラスを考え、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、それぞれの住宅地もオブジェクトとして、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捉えることができる。住宅地オブジェクトと家屋オブ

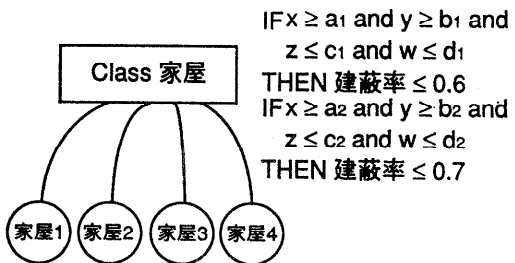


図 12 クラス定義による制約表現
Fig. 12 A representation of a coverage constraint by a class definition.

* ただし、処理効率とのトレードオフであり、高速化のための何らかの方法が必要となる。

ジェッ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、このような制約も伝播ビュー機構により表現できると考えられる。

4. 伝播ビューインスタンスの導出手順

本節では与えられた伝播ビュー定義の外延を求める基本的手順を示す。

4.1 伝播ビューインスタンスの導出手順

与えられた任意の伝播ビュー定義 P の外延を求める手順は、以下のように与えられる。

導出手順 **GetExtension**

入力：伝播ビュー定義 $P=(T, A, S, \leq_a)$

出力： P の外延 *Extent*

step 1: P が実クラスなら、その外延を *Extent* として終了。

step 2: P 中のターゲットクラス T に対し、手順 **GetExtension** を適用した結果を集合 O_i とし、各ソースクラスに対して **GetExtension** を適用した結果の和集合を O_i とする。 O_i が空集合ならば空集合を *Extent* として終了。 O_i が空集合ならば O_i を *Extent* として終了。

step 3: O_i 中の各インスタンス o に対し、以下のサブステップを適用する。

substep 3.1: O_i の部分集合

$$Source = \{s | o \leq_a s, s \in O_i\}$$

を生成し、*Source* が空集合ならば、 o を *Extent* の要素に追加し、 o に対するサブステップの適用を終了。

substep 3.2: 伝播ビューインスタンスの定義に基づき、 o に対して、 P で指定された継承可能属性とその属性値を、*Source* 中の各インスタンスから継承させた結果である伝播ビューインスタンス o' を生成し、それを *Extent* の要素として追加する。

step 4: *Extent* を出力として終了。 □

ただし、この手順は必ずしも停止しないことがある。

4.2 伝播ビュー定義に関する今後の課題

与えられた伝播ビュー定義から、そのすべての伝播ビューインスタンスを求めるには、伝播ビュー定義のターゲットクラスの各インスタンスから、第2章に述べた定義に従って、個々の伝播ビューインスタンスを生成してやればよい。伝播ビュー定義は、通常のクラ

スが外延を持つのと同様の意味で、その伝播ビューインスタンス集合を外延として持つクラスと見なすことができる。したがって、伝播ビュー定義中のソースクラスやターゲットクラスとして、別の伝播ビュー定義を指定してもよい。ただし、伝播ビュー定義の与え方によっては、不都合が生じることがあることに注意を要する。例えば、

$$P_1 = (P_2, A, S_1, \leq_{a1})$$

$$P_2 = (P_1, A, S_2, \leq_{a2})$$

のように、二つの伝播ビュー定義が、互いに他方のターゲットクラスに指定されている場合、それらの伝播ビューインスタンスと同一のオブジェクト識別子を有する実クラスのインスタンスは存在しないため、無意味である。

手順 **GetExtension** が停止するための条件に関する検討は重要な問題であるが、本論文の主題から離れるため、ここではこれ以上の議論を行わない。直観的には、与えられたすべての伝播ビュー定義中のソースクラスとターゲットクラスの参照関係、すなわち、ある伝播ビュー定義ではソースクラスとなっているクラスが、別の定義中ではターゲットクラスとして指定されるような関係がサイクルを形成しなければ、それらのクラスに対して **GetExtension** は停止する。また半順序関係を決定する属性について、ここまでの議論はすべての実クラスに定義された属性であると限定して行ってきた。これを継承属性も許すように拡張すると半順序継承した属性による半順序継承の連鎖が起きる可能性があり、この問題についても検討が必要である。

5. 関連する研究と本研究の位置づけ

従来における方法および代表的オブジェクト指向言語である SmallTalk-80 と、本論文におけるアプローチをまとめると表1のように整理される。SmallTalk-80 においては、基本的にインスタンス間の継承機能は提供されず、あるクラスの全インスタンス間で属性値を継承するための機構としてクラス変数があるだけである。

Banerjee らは、文献 8) においてオブジェクト指向 DBMS ORION におけるデータモデルの基本的概略を述べており、その特徴的概念として、スキーマ進化、コンポジットオブジェクト (Composite Object)、および版管理 (バージョン) 等がある。特に複合オブジェクトに関してコンポジットオブジェクト階層に基

表 1 種々のインスタンス間継承機構
Table 1 Instance-based inheritances.

	SmallTalk-80	ORIONにおける 値伝播	増永の方法	Wilkesのis-a継承	Scoreの方法	本論文のアプローチ
継承する 方向	クラスから 全インスタンス	コンポジットオブ ジェクトを構成する 親インスタンスから 子インスタンス	複合オブジェクトの 子から親インスタ ンス及び親から子イ ンスタンス (双方向)	親インスタンスから 子インスタンス	任意のインスタ ンスから任意のイ ンスタンス	任意のインスタ ンスからインスタ ンス
オブジェ クト間の 関係	instance-of関係	composite (part_of)関係	part_of関係	所属クラスの is-a関係	ユーザの指定した subobject- superobject関係	インスタンスの属 性値間の半順序関 係
継承され る対象	属性とその属性値	属性値	述語 (属性と属性値)	属性値	属性とその属性値 およびメソッド	属性とその属性値
実現手法	クラスオブジェクト への参照共有	composite object hierarchy概念に基づ く	クラスに付加され たルール	インスタンス生成 時の親オブジェク ト指定による属性 値共有機構	オブジェクト階 層概念に基づく	半順序関係評価に よる動的属性値評 価
主な用途 と目的	あるクラスの全 インスタンス間 での属性値共有	CADにおける部品 オブジェクト間の 属性値共有	複合オブジェクトの 各部品オブジェク トからの情報導出	設計情報の階層化 バージョン管理	実世界の実体の多 様な側面の表現	マルチメディア (時間的・空間的) オブジェクト管理
仮想的 可変性	・非仮想的 ・機構は固定的	・非仮想的 ・クラス定義更新に より変更可	・仮想的 ・ルール変更による 可変性あり	・非仮想的 ・継承機構は 固定的	・仮想的 ・継承様式の 変更可能	・仮想的 ・継承様式の 変更可能

づく値伝播(value propagation)と呼ばれる機構を導入している。これはある属性値が、コンポジットオブジェクト*から、その構成要素である子オブジェクトへ伝播するというものである。このとき継承がなされるオブジェクト間の関係は IS-PART-OF 関係に限定されている。ただし本論文では IS-PART-OF 関係とは、あるオブジェクトとその属性値として参照されるオブジェクトの関係を意味しており、コンポジットオブジェクトはその特別な場合である。

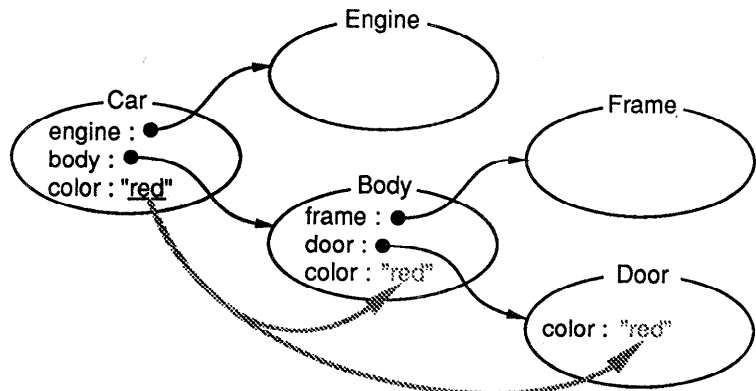


図 13 属性値の値伝播
Fig. 13 Value propagation.

例えば、Car オブジェクトが Body オブジェクトと Engine オブジェクトから構成され、Body オブジェクトが Frame と Door オブジェクトから成るとすると、Body と Door の “color” 属性の値は、親オブジェクトである Car オブジェクトの “color” 属性の値を用いるように指定で

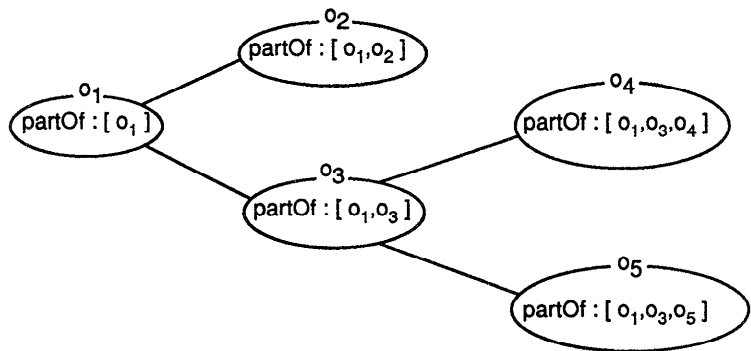


図 14 属性値を用いた IS-PART-OF 関係の表現
Fig. 14 A representation of IS-PART-OF relationships by attribute values.

* 一般にコンポジットオブジェクトを構成する上位オブジェクト。

きる (図 13 参照). このアプローチでは, “属性値” が継承の対象であるが, 伝播ビュー機構では属性名も含めて継承対象となっている. また継承基盤はオブジェクト間の IS-PART-OF 関係に限定されているのに対し, 伝播ビュー機構では属性値間の任意の半順序関係である.

また, この値伝播機構においては, 親および子供オブジェクトに属性が “物理的に” 作成されるが, 伝播ビュー機構では同一属性を “有しない” 場合に, 属性と属性値が仮想的に継承されるとしている. 同一属性を物理的に有する場合は, 値の伝播は起こらず本来の値が属性値として優先される. 伝播ビュー機構を, 継承対象がオブジェクト間で共通な属性の属性値となるように修正し, IS-PART-OF 関係を属性値の半順序関係として表現することにより, 上記のアプローチを模倣することができると考えられる. 例えば, 各部品オブジェクトに partOf 属性を持たせ, 複合オブジェクトの根オブジェクトから部品オブジェクトまでのパスを, リスト表現したものを属性値とすることで, そのリスト間の prefix (包含) 関係を用いて表現可能である (図 14).

増永は, オブジェクト間の IS-PART-OF 関係を表現する枠組みとして, クラス定義中にクラス間の IS-PART-OF 関係を表現する項目を持たせることにより, 各オブジェクトの構造記述を行うことを提案している⁹⁾. さらにクラスやオブジェクトに対するアクセス処理のための機構として, クラステンプレートに陽に記述された IS-PART-OF 関係にかかわる知識を表現する “ルール” を導入している. 例えば, 電気回路クラスを考え, ある電気回路インスタンスは機密であるかどうかという質問がメッセージとして発せられるとする. 電気回路が機密であるのは, その構成部品が機密であるときであるというルール R_1 が存在するとすれば, そのルールをクラステンプレートに記述する. R_1 を PROLOG 風に記述すると以下のようになる.

R_1 : 機密 (X) ←
電気回路 (X), IS-PART-OF (Y, X), 機密 (Y).

ここでのアプローチの特徴は, IS-PART-OF 関係 (およびクラス間の IS-A 関係) に基づくオブジェクト間の満たすべき条件をルールとして表現し, ある一方のオブジェクトの情報とルールから, 他方のオブジェクト*の情報を導出する機構である. これに対し,

* 親から子, 子から親, いずれの継承も可能.

本論文で提案した伝播ビュー機構では, オブジェクト間の関係は属性値間の半順序関係である.

増永の方法では継承されるオブジェクトの特性 (情報) を述語で記述し, その継承をクラスに付与されたルールで表現する*. このルール機構により複合オブジェクトの各部品オブジェクトからの情報導出を可能としている. またルールを用いているため, このルールの変更により継承の振る舞いをユーザが変更できると考えられる.

本論文の伝播ビュー機構において, 半順序関係の特殊な場合として IS-PART-OF 関係を考え, 属性とその属性値としてオブジェクトの情報 (色や機密性) を表現して, それらの継承という形に限定すれば, 増永のアプローチを表現することが可能であると考えられる.

Wilkes は汎化関係によるクラス間の属性構造の継承を IS-A 継承と呼び, それに対しインスタンス間での属性のみならず属性値の継承も必要であるとして, is-a 継承と呼ばれる継承機構を導入している⁹⁾. is-a 継承とは次のような継承機構である. クラス C および C' に対して, C' is-a C と宣言された場合, 以下の条件が満足されねばならない.

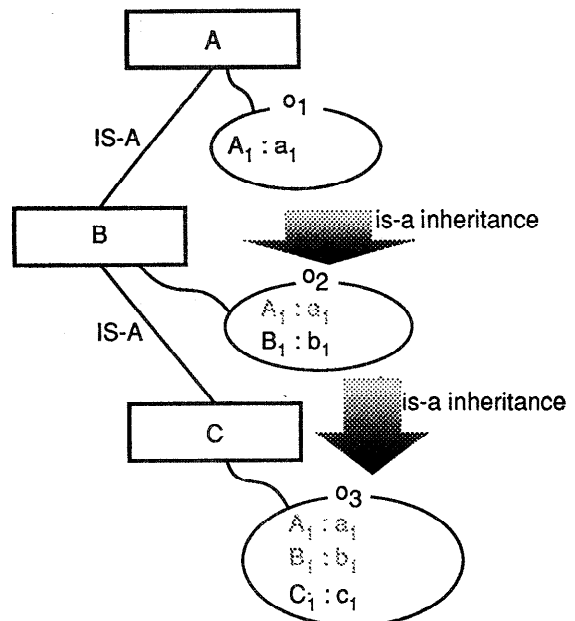


図 15 is-a 継承
Fig. 15 An example of is-a inheritance.

* ただし, それら述語の詳細については述べられておらず, 概要を示唆するに留まっている.

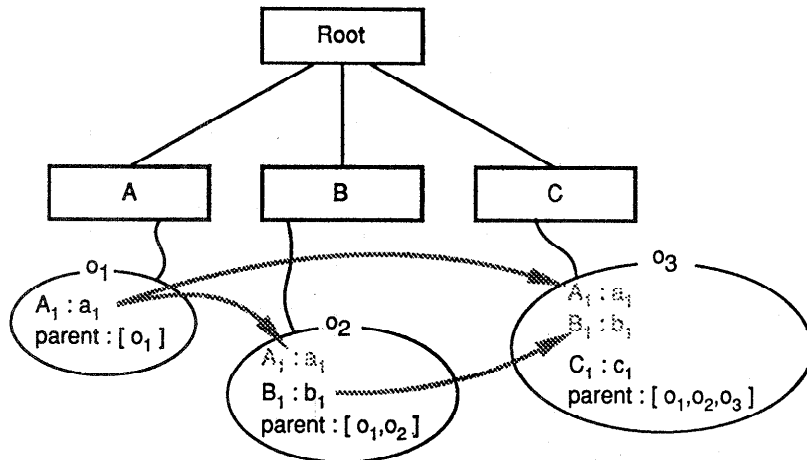


図 16 is-a 継承のエミュレーション
Fig. 16 An emulation of is-a inheritance.

1. C' は C のサブクラスであり、 C' は C で定義された属性構造を継承する。
2. C' のインスタンスは、 C のあるインスタンスから属性値を継承する。

サブクラスの各インスタンスは、スーパークラスで定義された属性構造のみならず、そのスーパークラスに属する、ある一つのインスタンスの属性値も継承する。すなわち、is-a 継承において継承基盤となるオブジェクトの関係は所属クラスの IS-A 関連であり、継承される対象はあくまでも属性値である。このアプローチは、クラス階層に基づいて属性構造のみならず属性値も、ある一つのインスタンスからサブクラスのインスタンスへ波及させることをユーザが陽に指定するものであり、CAD/CAM データベース応用における設計情報の階層的な管理において有効であると考えられる。しかし、これは従来のクラス階層に基づく継承機構の強化であり、汎化関係に基づくクラス階層概念に非常に強く拘束されていると言える。また、継承すべきオブジェクトは静的に指定され、実行時に動的に変化するような機能や仮想的に継承させる機能も提供されていない。

ここで、Wilkes のアプローチを伝播ビュー機構で模擬することは可能であると考えられる。図 15 のようなクラス階層を仮定し、クラス B のインスタンス o_2 は属性値をインスタンス o_1 から継承し、 o_3 は o_2 から継承するとする。これを伝播ビュー機構で実現するならば、図 16 のようなクラス構成として、継承すべき属性を有する親オブジェクトの継承パスを表現するリストを、ある適当な属性（例えば“parent”属性と

する）に保持させることで、“parent”属性の値となっているリストの prefix 関係すなわち包含関係を半順序関係とする伝播ビューを構成できる。

Banerjee らおよび Wilkes のアプローチは、実際にインスタンス自身に属性が作成される点で伝播ビュー機構とは異なっている。伝播ビュー機構では継承される属性は物理的にオブジェクトに付与されないで、ユーザが任意に伝播ビュー定義を変更してもオブジェクトの物理的更新は発生しない。このため、例えば試行的に伝播ビュー定義を追加することや、各ユーザごとに異なる伝播ビュー定義を持つことも問題なく行える。我々は、クラス階層や IS-PART-OF 階層など特定の意味の関連に依存した継承機構だけでは不十分であると考えている。例えば地理データベースにおいては、オブジェクト間の包含関係に基づく何らかの情報伝播・共有機構があれば有用な場合がある（例：神戸市内の各区の市外局番は、原則的に神戸市の市外局番と同一である等）。一般に地理オブジェクトの包含関係とクラス階層とは無関係であり、汎化以外のオブジェクト間の関連に基づいても、ある種の継承機構が実現できる必要がある。

Sciore は、実世界の実体を複数のオブジェクトから構成されるオブジェクト階層という概念を用いて表現し、これにより実体の有する様々な側面を表現することを提案している¹⁵⁾。オブジェクト階層を構成するオブジェクト r_1 , r_2 において、半順序関係 $r_1 < r_2$ が成立するとき、 r_2 を r_1 の superobject と呼び、 r_1 を r_2 の subobject と呼ぶ。subobject r_1 は r_2 より属性とメソッドを継承する。図 17 にオブジェクト階層の

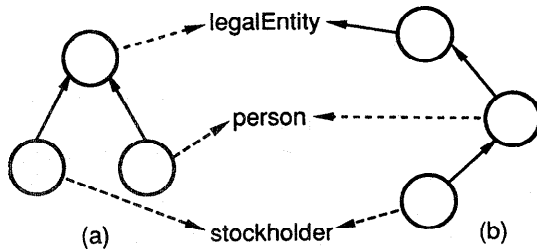


図 17 オブジェクト階層
Fig. 17 Object hierarchy.

例を示す。図 17 (a)のオブジェクト階層の stockholder は、legalEntity の属性やメソッドを継承するが、person の属性は継承しない。それに対し、図 17 (b)の stockholder は、legalEntity および person 両方の属性およびメソッドを継承する。つまり、図 17 (a)の stockholder は、単に legalEntity の特性を有しているのに対し、(b)の stockholder は person としての特性も有することになる。

この方法では、ユーザがオブジェクト階層と呼ばれる、複数のオブジェクトを陽に指定して構成した、実世界における一つの実体を表現するオブジェクト間の上位下位階層を構成し、それにおける上位オブジェクトから下位オブジェクトへ、属性、属性値やメソッドが継承される^{*}。この機構の基盤は実体を表現するオブジェクト階層にあり、これに基づくインスタンス間継承により実体の多様な側面を表現することを目指している。また継承は仮想的であり変更もオブジェクト階層に更新により可能であるが、実行時にオブジェクト階層が動的に変化して継承が変化するような機能はあまり考えられていない。

これに対し本論文の伝播ビュー機構では、オブジェクト間の関係は属性値間の任意の半順序関係であり、さらに伝播ビュー定義という形で、データベース中のオブジェクトとは、独立かつ宣言的に定義できるところが異なっている。

Sciore の方法についても、属性値の継承に関しては Banerjee のアプローチの場合と同様の方法で、伝播ビューとして実現することが可能であると考えられる^{**}。

^{*} しかし、Sciore は実現に関する具体的な示唆は与えていない。

^{**} ただし、Sciore のスーパーオブジェクトからサブオブジェクトへのメソッドの継承に対しては、現在の伝播ビュー機構ではメソッドに関して扱っていないので、何らかの拡張を要する。

6. おわりに

本論文では、伝播ビューと呼ぶ、属性値の半順序関係に基づく、インスタンス間の動的・仮想的な継承機構を提案し、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とは異なる伝播ビューについて基本的な定義を行い、その応用、伝播ビュー機構の処理の基本手順について論じた。本論文で提案した伝播ビュー機構の主要な特徴は以下のとおりである。

- 任意の属性定義域上の半順序関係に基づいて、属性と属性値をインスタンス間で継承する。
- 継承の基盤となる半順序関係が固定でなく、ユーザが任意に定義可能。
- 継承の対象となる属性やインスタンス群を、ユーザが伝播ビュー定義という形で宣言的に指定でき、また、これを自由に変更できる。
- 継承が実行時になされ、属性やその属性値が、継承されたインスタンスに保持されているかのように仮想的に見せかける。

また、伝播ビュー機構の実装例として、我々の開発したビデオデータベースシステム OVID および配置図面データベースシステムにおける伝播ビュー概念の適用例を示した。

今後の課題として、以下のようなものが重要である。

- 本論文で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られるが、その性質など、さらに詳しい検討が必要である。特に継承機構が矛盾なく機能するためには、継承の巡回などが起きてはならない。そこで伝播ビュー機構による属性継承処理の詳細な停止性に関する議論が重要である。
- 伝播ビュー導入に伴う実行効率が低下することは明らかであり、何らかの工夫が必要となる。我々はすでに伝播ビュー機構の効率的処理のための物理的記憶構造についても検討してきており⁹⁾、それに関する議論を進めてシステムの実装に反映させる必要がある。
- 現在の定義では、伝播ビューインスタンスを属性値とするオブジェクトの扱いに関して不十分である。すなわち、オブジェクトそのものが属性値になっている場合、属性値であるオブジェクトが伝播ビュー機構により、仮想化されるべきか否かという問題である。
- メソッドの継承は議論の簡単化のため、本論文では

言及しておらず、これについても拡張が必要である。例えば、都市地図データベースにおいて、ある地域中の建物に課せられる建蔽率が守られているかどうかチェックするメソッドを、地域内部の建物オブジェクトに継承させることができれば、各地域ごとに異なる建蔽率制約の判定基準(メソッド)をそれぞれの地域オブジェクトごとに付与するだけで、個々の建物オブジェクトに建蔽率制約判定のメソッドを付与する必要がなくなる。ただし、このようなメソッドの継承は次のような点で複雑な議論を要することは明らかである。まず継承されるメソッドが、内部で参照する属性はすでにオブジェクトに存在しているかまたは伝播ビューによって継承されていないかならなければならない。さらに継承されるメソッド中で参照される属性についても同様である。

- オブジェクトの属性評価が実行時のデータベース中のインスタンスの状態に応じて動的に決定されるため、オブジェクトの属性値更新や生成・削除により、オブジェクトの属性評価決定に影響があると考えられる。つまり、並列制御について議論が重要となる。

謝辞 本研究は、一部、科学技術庁技術振興調整費総合研究(自己組織型情報ベース)による。本研究にあたり、有益な議論をしていただいた神戸大学工学部北村研究室のデータベースグループ諸氏に感謝いたします。また、数多くの貴重なコメント・示唆をいただきました Obase コンソーシアムのメンバーの方々に深謝いたします。

参 考 文 献

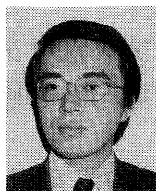
- 1) Atkinson, M. et al.: The Object-Oriented Database System Manifesto, *Deductive and Object-Oriented Databases*, pp. 223-240, North-Holland (1990).
- 2) Next-Generation Database Systems (Special Section), *Comm. ACM*, Vol. 34, No. 10, pp. 30-120 (Oct. 1991).
- 3) 大本英徹, 渥美 純, 田中克己: ビデオデータベースの概念モデリングとビジュアルインターフェイスについて, 電子情報通信学会技術研究報告, DE-89-50, pp. 17-24 (1990).
- 4) 大本英徹, 田中克己: A Video Object Model and Its Application to a Video Database System, 電子情報通信学会技術研究報告, DE-90-28, pp. 33-40 (1990).
- 5) 大本英徹, 奥田太郎, 田中克己: 空間的・時間的オブジェクトの包含関係に基づく継続機構とその物理的記憶構造, 情報処理学会研究報告, 91-DBS-84-26, pp. 219-228 (1991).
- 6) Oomoto, E. and Tanaka, K.: OVID: Design and Implementation of a Video-Object Database System, to appear in *Trans. on Knowledge and Data Engineering* (Aug. 1993).
- 7) 増永良文: 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).
- 8) Banerjee, J., Chou, H.-T., Garza, J.F., Kim W., Woelk, D., Ballou, N. and Kim, H.-J.: Data Model Issues for Object-Oriented Applications, *ACM Trans. on Office Information Systems*, Vol. 5, No. 1, pp. 3-26 (Jan. 1987).
- 9) Wolfgang Wilkes: Instance Mechanisms for Object Oriented Databases, *Proceedings of the 2nd International Workshop on Object-Oriented Database Systems* (Dittrich, K. R. Eds.), pp. 274-279 (Sept. 1988).
- 10) Stroustrup B. 著, 齊藤信男訳: プログラミング言語 C++, トッパン, (1988).
- 11) Goldberg, A., Robson, D. 著, 相磯秀男監訳: SMALLTALK-80 一言語詳細一, オーム社, (1988).
- 12) Deux, O. et al.: The Story of O₂, *Trans. on Knowledge and Data Engineering*, Vol. 2, No. 1, pp. 91-108 (Mar. 1990).
- 13) Tanaka, K., Yoshikawa, M., and Ishihara, K.: Schema Virtualization in Object-Oriented Databases, *Proc. Int. Conf. on Data Engineering*, pp. 23-30 (1988).
- 14) Kimura, Y. and Tsuruoka, K.: A View Class Mechanism for Object-Oriented Database Systems, *Proc. of 2nd Int. Symposium on Database Systems for Advanced Applications*, pp. 269-273 (Apr. 1991).
- 15) Sciore, E.: Object Specialization, *ACM Trans. on Information Systems*, Vol. 7, No. 2, pp. 103-122 (Apr. 1989).
- 16) Guttman, A.: R-Trees: A Dynamic Index Structure for Spatial Searching, *Proc. ACM SIGMOD*, pp. 47-57 (June 1984).
- 17) Timos, S., Nick, R., and Christos F.: The R⁺-Tree: A Dynamic Index for Multi-Dimensional Objects, *Proceedings of The International Conference on Very Large Databases*, Brighton, England (1987).
- 18) Freeston, M. W.: A Well-behaved File Structure for the Storage of Spatial Objects, *Design and Implementation of Large Spatial Databases* (Buchmann et al. Eds.), pp. 287-300 (July 1989).
- 19) Versant Object Technology Corp.: *VERSANT System Manual* (Sept. 1990).

(平成4年9月10日受付)
(平成5年6月17日採録)

**大本 英徹 (正会員)**

1963年生。1988年神戸大学工学部計測工学科卒業。1990年同大学院工学研究科修士課程計測工学専攻修了。1993年同大学院自然科学研究科博士課程知能科学専攻単位取得後中

退。同年より京都産業大学工学部情報通信工学科講師。マルチメディアデータベース、オブジェクト指向データベース、ユーザインターフェースなどの研究に従事。

**田中 克己 (正会員)**

昭和26年生。昭和49年、京都大学工学部情報工学科卒業。昭和51年、同大学院修士課程修了。京都大学工学博士。昭和54年、神戸大学教養学部助手。昭和61年、神戸大

学工学部助教授（情報知能工学科）、現在に至る。主にデータベース（RDB、履歴DB、OODB）の研究に従事。昭和60年、情報処理学会創立25周年記念論文採択、平成3年度情報処理学会 Best Author 賞受賞。現在、IEEE Computer Society, ACM, 日本ソフトウェア科学会、人工知能学会、システム制御情報学会各会員。