

(1991. 9. 24)

構造化ビューオブジェクトによるマルチメディアデータベースの検討

安田 直樹 服部 進実

金沢工業大学

あらまし 次世代データベースとして注目を集めているオブジェクト指向データベースは、いくつかの克服すべき問題点とその研究とともに明らかになってきており、その内の1つにビュー機能の弱さが挙げられている。これを解決する一案として、抽象データ型、ハイパーリンクの概念に基づいてビュー機能を提供する構造化ビューオブジェクトという概念を導入する。データベースに格納されているオブジェクトに対して直接アクセスを行うのではなく、目的とするオブジェクトの機能、データ構造を隠蔽しバーチャル化する構造化ビューオブジェクトを介在させることにより、オブジェクトの多様化、融通性を実現する方法について考察を行った。

Multimedia Database System  
with Structured View Objects

Naoki Yasuda Shimmi hattori

Kanazawa Institute of Technology

7-1 Ohgigaoka, Nonoichi-machi, Ishikawa 921, Japan

abstract The object oriented database system which is now considered as the next generation database is noted to have some problems to be cleared, for example, the poor view structure for retrieval access. In this paper, we are proposing concept of Structured View Objects (SVO) based upon abstract data type and hyper link. By using this new concept of SVO which virtualyzes data structure in the database system, variety and flexibility of objects structure can be realized for user access to the database system.

## 1. はじめに

近年、CAD/CAMやCASEに代表されるエンジニアリングデータベース、あるいは、画像や音声なども扱えるマルチメディアデータベースなどのような、関係データベースではサポートしきれないような複雑なデータを効率的に扱う新しいデータベース技術が求められるようになってきた。そこで、関係データベースにかわる次世代データベースとしてオブジェクト指向データベースが注目を集めている。オブジェクト指向データベースでは、複雑な対象の直接的な表現能力、データと手続きとのカプセル化による対象の動的なモデリングなど、関係データベースでは実現できなかった機能に充分対応できるようになっている。だがその反面、まだ新しい技術であるため、幾つかの解決すべき問題点や、データベースをより高度化するために必要な機能が指摘されている(文献1、2、3、4、5、6)。我々は、その中で特にオブジェクト指向データベースのビュー機能に注目した。ビュー機能とは、データをユーザの求めるデータ構造(ビュー)に変換して提供する機能である。データベースを単なる情報蓄積の道具と区別しているのは、データについての多様な管理・操作能力であり、ビュー機能は、ユーザがデータベースにアクセスするための重要な操作能力のひとつであると考えられる。そこで、オブジェクトの仮想化、多様化を行い、より柔軟で拡張性に富んだシステムの構築をするための検討を行った。本稿では、その結果得られた一つの方法について報告する。

## 2. データベースのビュー機能について

この章では、関係データベースとオブジェクト指向データベースのビュー機能について簡単に説明するとともに、表題に掲げた構造化ビューオブジェクトについて説明する。

### 2.1 関係データベースのビュー機能

関係データベースでは、一つ概念モデルを複数の独立した関係に分割し、格納するのが普通である。この分割はデータベースへの無秩序なアクセスを防ぐため、定められた規約に基づいて操作されるものであり、それを正規形と呼ぶ。正規形に直され、格納されたデータから、ユーザが必要な情報を取り出す際には、SQLのような問い合わせ言語を用いて、必要なデータ集合を定義する。この定義は、関係の名前と、そこから取り出す属性名、そして、値に適用する条件とから構成される。この定義によって求められる新しい関係は、実際にデータベース上に格納されている複数の関係をもとにデータベース上で関係代数演算を行って得られるもので、見かけ上は1つの関係として提供されるものである。

関係データベースにおけるビュー機能は、問い合わせ機能そのものであるといえる。

### 2.2 オブジェクト指向データベースのビュー機能

関係データベースでは、関係同士は互いに独立し

た存在であった。そのため、複数の表から一つの関係を導き出す時は、関係代数演算時にそれぞれの関係のキーを基にしてそれぞれの関係を結びつけていた。だが、オブジェクト指向データベースの場合では、オブジェクト同士が構造を持つことによってデータ構造を表現するため、構造による制限が強くなってしまふ。言い替えれば、どのようなデータをアクセスするときにも、ユーザは目的とするデータのデータ構造を意識している必要が生じてしまふ。その点で、オブジェクト指向データベースは関係データベースに比べてビュー機能は弱いといえる。また、オブジェクトというデータの単位は、それ自体が1つの意味をもった存在として扱われるが、任意のオブジェクトはどのようなユーザにとっても同一のオブジェクトとして提供される(図1)。関係データベースでは、同じ関係からであっても、ユーザは自分に必要な関係を導きだして利用していたことを考えると、ユーザが独自のオブジェクトを作成することができるようになる機能が必要となる。

前者について言えば、オブジェクト指向データベースがオブジェクト同士の構造の上に成り立っているからこそ、複雑なデータ構造も直接的に表現でき、柔軟性の高いスキーマも実現できるのであるともいえる。そこで、オブジェクト間の構造を動的なものとし、ユーザごとに異なったスキーマとして提供しようという研究もある(文献7)。

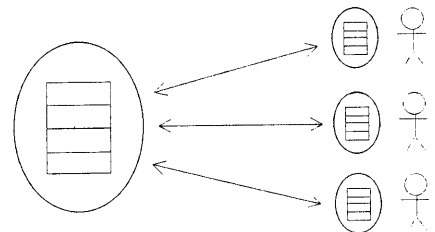


図1 通常のOODB

### 2.3 SVOによるビュー機能

前節で述べたように、現在のオブジェクト指向データベースのビュー機能の問題点の1つは、データベースからユーザ(またはアプリケーションプログラム)へのオブジェクトの提供の仕方にあると考えられる。それはつまり、あるオブジェクトはだれにとっても同じ構造をもち、同じクラスのインスタンスであるということである。言葉を変えれば、現在のオブジェクト指向データベースでは、オブジェクトとビューが同じものであり、1つのオブジェクトには1つのビューしか許されない。それを解決するためには、オブジェクトとビューとを分離することが必要であると思われる。つまり、オブジェクト指向データベースにおけるビューとは単なるデータの集合体ではなく、性質(データ、手続き)をもった対象(オブジェクト)について、別の視点から再構成したオブジェクトであると考えられる。

言い替えると基本的な考え方は、次のようなものである。オブジェクトとはデータとメソッドとをカプセル化したものである。このデータの部分に別のオブジェクトを格納し、メソッドには格納したオブジェクトへのメッセージを定義しておく(図2)。

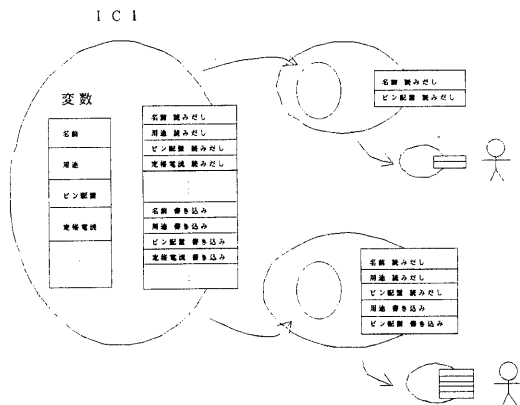


図2 SVOの働き

このようなオブジェクトをデータベースの基本機能として提供し、かつ、ユーザによる拡張性を保証し、アクセス権の下に秩序だった構造として利用できるようにする。

ここで、そのようなオブジェクトのことを、ビューのデータ構造を定義するオブジェクトとして、「構造化ビューオブジェクト」(以下単に「SVO」と呼ぶこととし、SVOを適用するオブジェクトを「ターゲットオブジェクト」と呼ぶことにする。ターゲットオブジェクトはビューを作り出すための対象であり、SVOはその対象についてのビューとして存在する。

SVOを用いることにより以下のような利点が考えられる。

- ・ 1つのターゲットオブジェクトに対して複数のSVOを定義すれば、そのオブジェクトは定義した各SVOのインスタンスとして扱うことができる。

- ・ SVO中でターゲットオブジェクト以外の変数を設定することで、もとのデータ(ターゲットオブジェクト)に手を加えることなく独自の情報(オーナーデータ)を付加できる(図3)。

- ・ 同じオブジェクトであっても、アクセスできるSVOを限定することによってクラス毎のアクセス権を設定できる。

### 3. SVOに基づくデータベース構築に必要な基本要素

ここではSVOを用いたデータベースを構築する

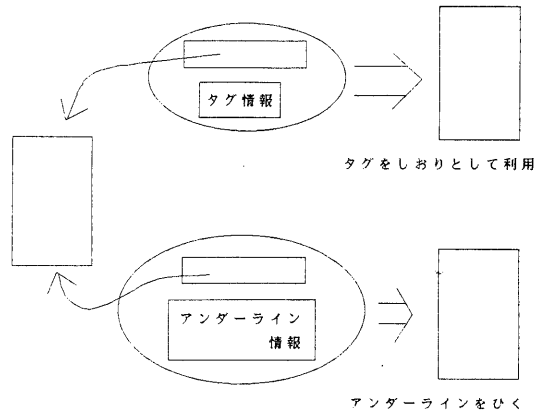


図3 SVOによるユーザ独自の情報の付加

ために必要なものについて、以下に述べる。

#### 3. 1 クラス階層(構造化ビュークラス階層)

SVOには大きく2つの種類が存在する。それは、

a) 他のSVOのインスタンスを値として含まない、一般のOODBの「クラス」と同じもの

b) 他のSVOのインスタンスを、値として持つもの

である。後者はさらに

b 1) 他のSVOのインスタンスのみを値に持つもの

b 2) 他のSVOのインスタンスとともに、それ自身が独自の値を持つもの  
の2つに分けることができる(図4、5)。

a) の場合、SVOにはインスタンス変数の値として取り得る値の型と、それらに対するメソッドが定義されることになる。

b) の場合、基本的にはa) の場合と同じであるが、インスタンス変数の値がSVOのインスタンスとなる場合、SVOを型として扱う必要があり、SVO型を導入することにする。SVO型とは、変数のドメインとしてSVO名をとるもので、実際の値としては指定したSVOインスタンスのオブジェクトIDが格納される。値にオブジェクトが代入されるという意味では、通常のOODBのクラスにおける、入れ子構造のデータ構造を定義する場合と基本的にはおなじである。a) と b) の違いは、定義中にSVO型が含まれるかどうかの違いである。

また、b 1) の場合、図4に見られるように抽象クラスのような存在であり、抽象データ型としての性質が強い。実際のアクセスに際しては、アクセスの度にインスタンスが生成され、そこへの変数値としてターゲットオブジェクトのオブジェクトIDが代入されることになる。

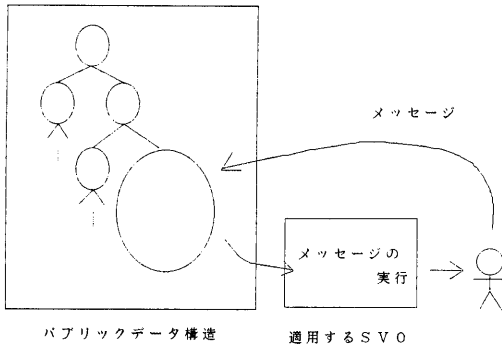


図4 ユーザ独自の情報が無い場合

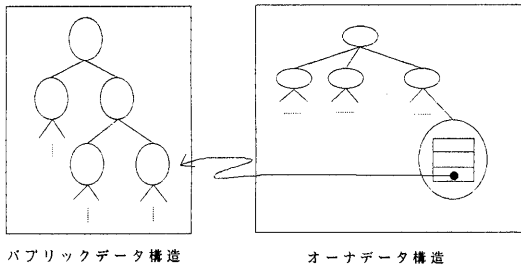


図5 ユーザ独自の情報がある場合

b 2) の場合では、図5のように、ハイパーリンクを含んだオブジェクトといった性質のものになっている。実際のアクセスの時はb 1) の場合と異なり、オーナーデータ構造 (3. 3参照) として管理されている局所的なオブジェクト構造中にインスタンスが格納され、そこにインスタンス変数をもった実体として管理される。このときSVO型の変数の中身はターゲットオブジェクトのオブジェクトIDとなる。

構造化ビュークラスは階層構造を持った拡張可能な管理が行われることを想定している。ただし、このとき、任意の構造化ビュークラスと、そのターゲットオブジェクトのクラスとがあった場合、ターゲットオブジェクトのクラスが構造化ビュークラスのスーパークラスとなっている必要は全くない。なぜなら、ターゲットオブジェクトはインスタンス変数の値として用いるにすぎないからである。また、構造化ビュークラスの定義について、定義を行うこと自体は特に制限設けるものではないが、それをアクセス辞書に登録する場合には許可権による制約が必要である。これは、アクセス辞書に登録しないSVOはアドホックなものとして利用可能であり、その場合には、アクセス辞書への登録はかえって面倒であることと、SVOがいたずらに増大するのを防ぎ、アクセスに関する秩序を守るための措置である。

### 3. 2 メタデータ

メタデータはユーザごとに用意されるもので、アクセス辞書と、オーナーデータの情報、そしてアクセス辞書の操作のためのアクセス権といった情報が格納されている。

まず、アクセス辞書には、データアクセスに関するSVO変換の情報が格納される。

アクセス辞書は図6に示すような構造をしている。図において、

- ・SVO1のインスタンスは、SVO15のインスタンスとしてアクセスされる。

- ・SVO2のインスタンスは、そのままSVO2のインスタンスとして扱われる。

- ・SVO3のインスタンスは、SVO7、SVO8、SVO11のいずれかのインスタンスとして扱うことができる。

ということを示している。

アクセス辞書は、ユーザがデータベースを使用する際に呼び出され、オブジェクトへのアクセスが行われる毎にDBMSに参照され、必要な場合にはSVOへの変換が行われる。データ構造自体を仮想化せず、このような形にしたのは、1つのターゲットオブジェクトが複数のSVOに変換されるからである。また、一つのアクセス辞書はパブリックなアクセス辞書を基に、その差分を記述したユーザ独自のアクセス辞書を重ね併せて完成する。この動作は、ユーザがデータベースにログインした時点で行われる。

オーナーデータの情報は、ユーザごとに登録してあるSVOのオーナーデータと呼び出すためのもので、データベース中に登録してあるものが指定される。

アクセス辞書操作のためのアクセス権は、アクセス辞書が無秩序に変更されることによって、ターゲットオブジェクトの操作が無秩序に行われることを防ぐためのものである。アクセス権はパブリックなアクセス辞書のものと、ユーザ各自のものと2つがある。

```

svo1 as: #svo15.
svo3 as: #(svo7,
           svo8,
           svo11).

```

図6 アクセス辞書の例

### 3. 3 パブリックデータとオーナーデータ

データベースとして管理するデータには、パブリ

ックデータとオーナーデータの2種類が用意される。パブリックデータは通常のオブジェクト指向データベースにみられるような階層構造を成している。オーナーデータはユーザ毎に用意されるもので、3.2のb2)で述べたようなインスタンスのローカルな値を保持したSVOのインスタンスの集合体である。オーナーデータも階層構造を成すが、パブリックデータのそれとは異なる(図7)。まず、階層構造の頂点にくるのはユーザのIDである。その下には、そのユーザにアクセスが許されたSVOのうち、ローカルな値を持つもののSVO識別子が並ぶ。さらにその下にそのSVOのインスタンスが並ぶことになる。ローカルな値を持たないSVOはこの中には含まれない。このオーナーデータの構造は、データのアクセスを前提としたものである(4.2参照)。

パブリックデータは全てのユーザからアクセスできる、いわば基本となるデータ構造であり、オーナーデータは、それに重ねあわせて個々のユーザが自分のデータベースにカスタマイズするためのものである。

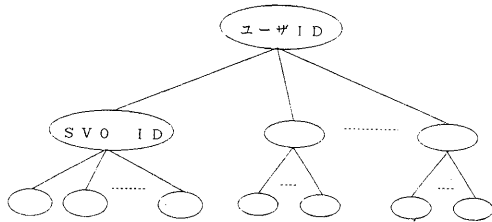


図7 オーナーデータ構造

### 3.4 データベース操作の手順

いままで述べた各要素の実際の操作時の動作について簡単に説明する。

まず操作にあたっては、ユーザがデータベースにログインする必要がある。データベースはログインしたユーザのメタデータからオーナーデータ構造と辞書データの情報を取り出す。次に、そのユーザのオーナーデータを取り出して(図8)。

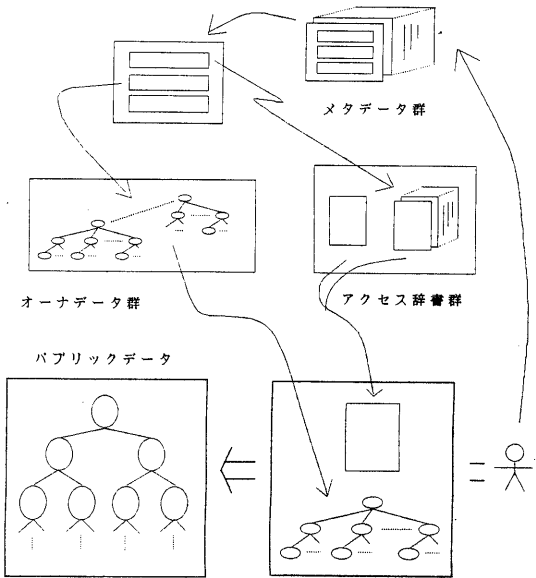


図8 ユーザのデータベースへのログイン

### 3.4 DBMSとDML

DBMSは次のような機能を備えている必要がある。

- ・ユーザの登録、変更、削除機能
- ・アクセス辞書の内容について、追加、変更、削除
- ・アクセス権について、変更、削除
- ・オーナーデータ構造について、データ構造の生成、変更、削除

以上のものについては、それら进行操作するための操作言語が必要であるが、必ずしもデータベース上で扱う必要はないものと思われる。これらはデータ構造自体に直接的に関わるものではなく、ある程度ユーザから隠れているほうが都合がよいのである。そのため、DBMSでは操作言語を解釈、実行できることが求められるが、必ずしも質問言語と統合される必要はない。

オブジェクトへのアクセス動作は次のようなものである。

まず、ユーザからオブジェクトとそれに対するメッセージが送られてくる。すると、まず指定されたオブジェクトがパブリックデータ構造から取り出される。

次に辞書データ上にそのオブジェクトのSVOが登録されているかを確認する。登録されていなければ、そのままメッセージを実行することになる。もし、他のSVOのインスタンスとして変換が行われる必要があった場合、オーナーデータからそのオブジェクトとつながりをもつものが検索される。検索は、まず、オーナーデータ構造の中から変換するSVOが含まれているかどうかを調べる。含まれていなければ、ローカルデータを持たないSVOとして、そのSVOのインスタンスが生成され、そのインスタンス変数にオブジェクトが代入され、メッセージが実行される。SVOが含まれていた場合には、そのSVOのインスタンス群からターゲットオブジェクトのオブジェクトIDをインスタンス変数として保持しているものを探し出し、そのインスタンスにメッセージを実行させる(図9)。

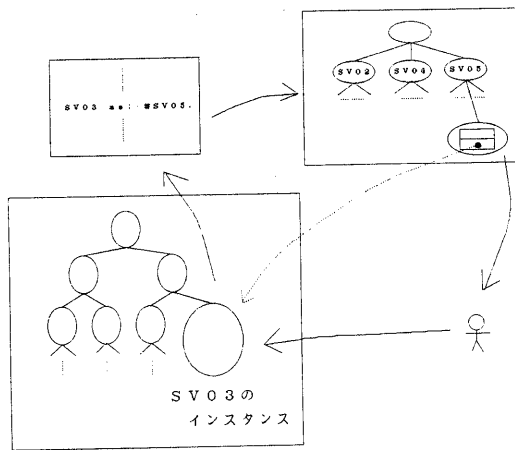


図9 データアクセス時の処理の流れ

また、メッセージとともにSVOが指定されているときには、パブリックデータから取り出したオブジェクトが指定されたSVOに変換可能であるかどうか調べる。変換可能であれば上で説明した手順でメッセージが実行されることになる。

#### 4. 今後の課題

本方式の最大の問題点は、一貫性の維持についてである。SVOにローカルなデータを持たせたことによって、データ構造はハイパーリンクをもったシステム、ハイパーテキストやハイパーメディアなどに近い性質をもつことと等価になる。リンクしているターゲットオブジェクトと意味的に強いつながりをもたせているような場合に、ターゲットオブジェクトが変更、あるいは削除されてしまった場合、どのように対処するべきかという、従来のオブジェクト指向データベースの一貫性問題に加えてハイパーリンクの一貫性問題もからんでくる。

また、ビュー機能を与えるために、ターゲットオブジェクトを他のオブジェクトの入れ子構造として表現しており、パフォーマンスの劣化につながる。また、アクセスの方法についても現時点では最善のものとは言えず、改良の余地があるものと思われる。

オーナーデータ構造、メタデータを改良し、マルチメディアデータの時系列データの管理、バージョンデータの柔軟な管理などについて、今後の検討課題と考えている。

#### 文献

- 1) 大須賀節夫：データベースと知識ベース、オーム社、1989
- 2) 増永良文、田中克己：次世代データベースの展望、情報処理、Vol 32、No. 5
- 4) Won Kim：オブジェクト指向データベース 一定義と研究の方向、日経AI別冊1991冬号、pp 154-175
- 5) 西尾章示朗、田中克己：オブジェクト指向データベースシステム宣言とその意義、bit、vol.

22、No. 8

6) 西川正俊、下條真司、田中克己：オブジェクト指向データベースシステムの最近の研究・開発動向、情報データベース・システム 90-30、1990

7) 宇都宮、遠藤他：知識表現機構をもつハイパーメディア文書モデルとその試作、情報マルチメディア通信と分散処理 45-15、1990

8) K. Parsaye, M. Chignell, S. Khoshafian and H. Wong: Intelligent Databases, John Wiley & Sons, Inc., 1989