データベース・システム 83－6
（1991. 5. 24）

# 「OODBによるマルチメディアDB実現法の検討」

桑澤嘉宏*, 鈴木幸市†, 本橋征行‡

NTTデータ通信株式会社 開発本部

様々な記憶媒体上に存在する数値、テキスト、静止画情報を扱うマルチメディアDBをOODBを用いてDB利用者側で実現する方法を検討した。これは、DBMS管理下のオブジェクトとして管理オブジェクトを作成し、この管理オブジェクトと付加した記憶媒体上のデータの間をメソッドによって一対一の関係を作ることで実現している。本稿では、まず既存のオブジェクト指向DBMSが提供してきた機能を用いてマルチメディアデータベースを実現する方法を述べ、そこから、マルチメディアDBを実現するためにOODBMSが備えるべき機能を検討している。また、検討した方法に基づく検証システムについて述べる。

# A Multi-Media Database using Object-Oriented Database

Yoshihiro Kuwazawa, Kouichi Suzuki, Masayuki Motohashi

NTT DATA COMMUNICATIONS SYSTEMS CORPORATION

Kowa Kawasaki Nishiguchi Bldg., 66-2 Horikawa-cho, Saiwai-ku, Kawasaki-shi Kanagawa 210 Japan

We have investigated how to implement a multi-media database which can handle numerical, text and still-image data on various kinds of storage media which is out of the control of DBMS itself. We realized it by maintaining one to one relation between management object which is under DBMS control and data on external storage media. In this paper, this approach using available Object-Oriented DBMS will be described, and desirable functions of OODBMS to realize multi-media database will be proposed. A prototype system based on this approach will also be mentioned briefly.

*atom@rd.nttdata.jp
†kouichi@rd.nttdata.jp
‡motohasi@rd.nttdata.jp

## 1 はじめに

データベースをマルチメディア化するためには、使用可能な記憶媒体を拡張し、そこにさまざまな種類の情報を記録できるようにしなくてはならない。記憶媒体の拡張に関しては、近年 LD, CD 等のさまざまな媒体が次々に現われている。また、さまざまな情報を扱うことに関しては、例えば CD-DA, CD-ROM, CD-I, DVI といったように多くの記録フォーマットによって動画、音声といった多様な情報が記録されるようになってきている。従って多くの媒体上のこれらの情報をオブジェクトとして管理する機能を DBMS にあらかじめ組み込むことを要求することは、実際的ではない。

だが、現時点でもマルチメディア DB システムを構築したいというニーズは大きい。このような状況下、磁気ディスク以外の媒体を用いたマルチメディアシステムをユーザが構築する場合には、データベース側に管理情報を保持し、その情報を元に光ディスク等の媒体にユーザーアプリケーションがアクセスを行う仕組みが一般的である [Kuwa 90][Haya 90]。しかし、この方法では、磁気ディスク以外に存在する情報は DBMS の管理下にないため、データベースシステムの利点を十分に生かすことができない。なぜならば、DB 上の管理情報と付加した媒体に存在する情報との一貫性保持機構をアプリケーションプログラム側で作成しなくてはならないからである。従って、付加媒体上のデータについても DBMS の管理を及ぼすことが望ましい。これによってアプリケーションプログラマは、拡張媒体上のデータをもともと DBMS がサポートする媒体上のデータと同様に一貫性保持を気にすることなく扱うことができる。

この要求を満たすため、本研究では著者らが [Moto 90a] で提案したオブジェクト指向に基づくデータモデルを採用した。

以上より目標は、「ユーザ側で既存 OODBMS を拡張する形態でデータベースの媒体／情報双方のマルチメディア化を行うこと」とし、その第一ステップとして、以下を目指した。

> ◎ 従来データベースが提供していた数値、テキスト情報に加え、静止画情報をさまざまな記憶媒体上で扱うマルチメディア情報データベースを既存 OODBMS を用いて実現すると共に、その際に DBMS が備えて欲しい機能を明確にする。

## 2 従来の研究状況

マルチメディア DB 実現に関係する研究には、幾つかの流れがある。第一は、非正規データを扱うことができる拡張 RDB によるマルチメディア DB の実現である。これは一部の商用 RDBMS で実現されている。しかしこの方法では例えば動画情報の出力を考えた場合、DBMS を通して情報を出力するためそのオーバーヘッドは避けられず、動画情報のようなリアルタイム性の高い情報では問題がある。

第二は、拡張可能 DBMS によるマルチメディア DB 実現の研究である [Bato 88][Care 87]。また、OODBMS を用いたマルチメディア DB の実現もこの流れに属しているといえる [Woel 87][Turu 89]。本研究もこの OODBMS を用いたマルチメディア DB 実現のアプローチをとっている。このアプローチをとれば、先に述べた RDB での問題は以下のようにして回避できる。すなわち、[Suzu 90] で述べているように OODBMS が管理するメソッドによって、動画情報を読み出すプロセスを直接起動し、DBMS を経由せずに動画情報を必要とするプロセスに渡すようにすればよい。しかし、これらの研究において、ユーザ側でマルチメディア拡張を可能とし、さらに拡張媒体上のデータに対しても一貫性制御を及ぼす方法は明確に述べられていない。本研究では、このユーザ側でのマルチメディア拡張、拡張媒体上のデータに対する一貫性制御を行う方法を明確にしようとするものである。

## 3 定義

以降の記述の正確を期すために幾つかの用語の定義を行う。

**(1) DBA(Database Administrator) と利用者**
DBA は、利用する DBMS を アプリケーションのニーズに応じてマルチメディア拡張する人とする。また利用者とは、媒体拡張された DBMS を利用するアプリケーションプログラマとする。

**(2) 基準空間と拡張空間**
DBMS が標準でサポートする、データを記録する媒体空間を基準空間、DBA が拡張を行なった媒体空間を拡張空間と呼ぶ。

**(3) 時間同期情報と時間非同期情報**
時間同期情報は、動画、音声情報のように時間軸上に展開される情報のこととする。例えば、音声情報は、一

定の情報速度で出力しなければならない。これは、動画も同様である。さらにビデオの情報などで、音声情報と、画像情報が別オブジェクトとして保存されている場合には、その出力は同期してなされなければならない。

時間非同期情報は、数値、テキスト、静止画情報のように時間軸上には展開されない情報とする。従って、同期等の考慮をしなくて済む。

(4) **データオブジェクト**
利用者がオブジェクトとして捉えることのできるデータのことをこう呼ぶ。例えば拡張空間中の静止画一画面分の情報などがこれに当たる。

## 4 条件

ここで、以降の考察を進めていく上での条件を整理する。

- **媒体に関する条件**
  媒体の種類は特に限定しない。
- **情報に関する条件**
  対象とする情報は、時間非同期情報とする。

## 5 実現のアウトラインと実現のための課題

これまでの我々の研究によって、OODBMS でマルチメディア拡張を行うためのアウトラインは示されている[Moto 90b]。それは、以下のようになる。図 1に示すように拡張空間中のデータを DBMS の管理下に置くために基準空間中にオブジェクトを作り、そのオブジェクトと拡張空間中のデータオブジェクトとを一対一対応させる。この基準空間中のオブジェクトを管理オブジェクトと呼ぶ。ここで一対一対応のリンクメカニズムが必要になる。管理オブジェクトは、図 2に示すように、利用者に対して拡張空間中のデータ・オブジェクトがあたかも基準空間中に存在するように見せる機能をもつ。

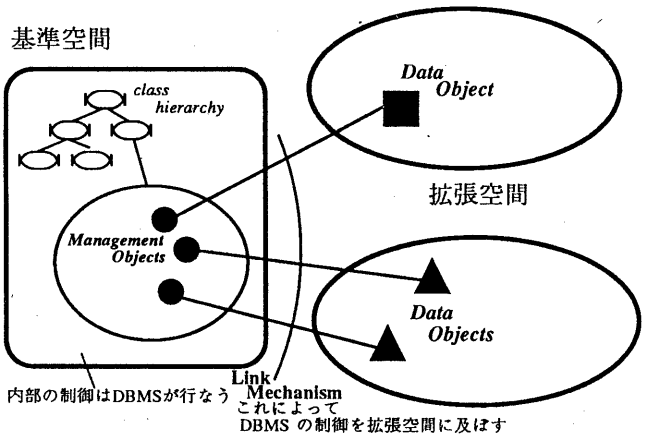


図1: OODBMSでマルチメディア拡張を行なうためのアウトライン

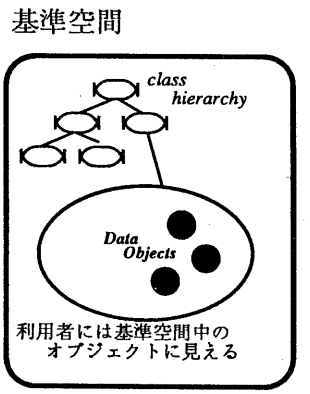


図2: 利用者から見た管理オブジェクト

このアウトラインを現実化するための課題は大別すると 3 つになる。

(1) **管理オブジェクトのクラス階層中での位置付け**
拡張媒体中のデータと結び付いた管理オブジェクトをこの OODB のクラス階層中のどのクラス配下におき、継承機能をどのように利用するかという点である。この点は、利用者がプログラミングする時、また DBA が使用媒体や使用情報フォーマット等を拡張する時にこの部分の設計の善し悪しが影響するため、重要である。

(2) **リンクメカニズムの機能**
管理オブジェクトと拡張媒体中のデータオブジェクトを結び付けているリンクメカニズムにはどのような機能が必要か、明確にする必要がある。

(3) リンクメカニズムの実現

上記で明確になったリンクメカニズムの機能について、その実現方法を明確にする必要がある。

## 6 既存 OODBMS による課題の解決

上記の課題について、以下のように既存 OODBMS の利用を前提として解決を行なった。

### 6.1 管理オブジェクトのクラス階層中での位置付け

クラスは、自動車会社クラスとか、社員録クラスというように本来、情報の型によって定義される概念であり、媒体の指定は、基本的にはクラス定義とは直交していると考えることができる。この考えに基づけば、あるクラスに属するオブジェクトを管理オブジェクトとする場合、その生成時に管理オブジェクトが管理するデータオブジェクトが存在する媒体の指定をし、その情報をインスタンス変数 (アトリビュート) に記録すれば良い [Kato 90]。すなわち、smalltalk 風の記述をすれば、

```
aClass newOn: 'OpticalDisk'.
```

とすることで、aClass 配下の、その対応するデータオブジェクトが光ディスクに存在する管理オブジェクトを生成することができる。

しかし、この考え方には問題が存在する。それは、媒体の指定は本来クラス定義とは直交しているはずなのだが、媒体によっては、ある種のメソッドを受けつけることができない場合があるために生じる。

具体例を示す。aClass のインスタンスとしての管理オブジェクトを考える。光ディスク、レーザーディスク (読み込み専用) に対応した管理オブジェクト、anOdkObject, anLdObject があるとする。また、aClass には、データ書き込みのためのメソッド "dataWrite:" が、定義されているものとする。このとき、利用者のアプリケーション中で anOdkObject, anLdObject にこのメソッド "dataWrite:" が送られたとする。

```
anOdkObject dataWrite: aWriteData.
anLdObject dataWrite: aWriteData.
```

光ディスクに対しては、書き込み可能なデバイスのため、そのメソッドは、少なくとも原理的には有効である (もちろん、DBA がそのためのプログラムを記述しなくてはならない)。しかし、読み込み専用のレーザーディスクでは、このような操作は、原理的に不可能である。すなわち、同じクラスのオブジェクトにもかかわらず、適用可能なメソッドと不可能なメソッドが生じる可能性がある。従ってマルチメディア DB では、記録媒体別にクラスを作成する必要がある。

オブジェクト指向計算における継承システムには、single inheritance と multiple inheritance がある。現在の OODBMS では、そのデータ定義・操作言語 (DDL: Data Definition Language / DML: Data Manipulation Language) が single inheritance を採用している OODBMS と、multiple inheritance を採用しているものがある。このそれぞれの継承システムについて、上述のように記録媒体別にクラスを作成することを前提とし、クラス階層中の管理オブジェクトの位置付け方を検討する必要がある。しかし、ある方法が最適であると決めることは困難であり、ここでは一例を示すにとどめる。

(1) multiple inheritance を採用する OODBMS

図 3 に示すように媒体駆動のためのメソッドを記述するクラス "MultiMedia" と、その媒体上の情報のフォーマットのためのメソッドを記述するクラス "MultiInformation" をつくり、それらの multiple inheritance によって、媒体拡張のためのクラス階層を実現する。

この方法の利点は、媒体拡張と、扱う情報の種類の拡張を全く別に行なうことができる点である。これによって柔軟なマルチメディア化を実現できる。

(2) single inheritance を採用する OODBMS

この場合には、図 4 のようにある機能を果たすクラスの配下に媒体対応のサブクラスを付加することで実現する。この方法の欠点は、あるクラスを拡張空間に拡張しようとすると、そのつど、媒体対応のメソッドを記述した媒体対応サブクラスを作らねばならず、冗長なことである。

逆に、媒体対応のクラスを作ってからそのサブクラスとして、MultiInformation のクラス、Magnitude クラス等さまざまなクラスを付加することもできるが、やはり冗長であることにはかわりない。

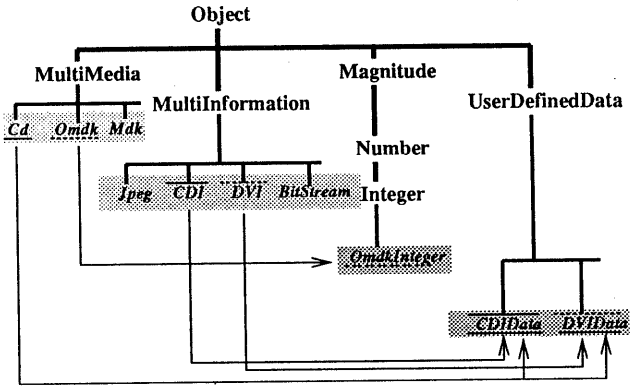



図3: multiple inheritance を利用した
媒体拡張のためのクラス階層

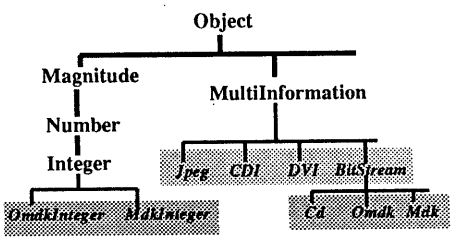



図4: single inheritance を利用した
媒体拡張のためのクラス階層

## 6.2 リンクメカニズムの機能

リンクメカニズムの基本的な機能は、基準空間中の管理オブジェクトと拡張空間中のデータオブジェクトとの一貫性を保持することにある。この一貫性保持のために下記の機能が必要になる。

- 管理オブジェクトの生成・削除とそれに伴う拡張空間中のデータオブジェクトの生成・削除機能
- トランザクション制御のための commit/abort 時の機能

## 6.3 リンクメカニズムの実現法

上記の 2 つの機能について述べる。

(1) 管理オブジェクトの生成・削除とそれに伴う拡張空間中のデータオブジェクトの生成・削除機能

i. 拡張空間中データの生成
管理オブジェクトおよび拡張空間内データオブジェクトの生成処理に関しては問題はない。管理オブジェクトが所属するクラスにおいてインスタンス生成のクラスメソッドに拡張空間中のデータオブジェクトを作る手続きを記述しておけばよい。

ii. 拡張空間中データの削除
管理オブジェクトが削除された時、拡張空間内のデータオブジェクトもリンクして削除するようにしなくてはならない。もし、利用者が管理オブジェクトに対して陽に削除メッセージを送った場合には問題はない。その削除メッセージであるメソッドに拡張空間内のデータオブジェクト削除[1]のための操作を付け加えればよい。問題となるのは、管理オブジェクトがガーベッジ・コレクタの対象となる場合である。

これは、宣言的な検索を行なう際の検索スコープに関係する。OODBMS では、今のところ 2 つの考え方がある。一つは、検索スコープをクラスないしはクラス階層とする考え方 [Kim 89]、もう一つは、Set, Bag, Array といった Collection のオブジェクトを検索スコープとする考え方である [Bret 89]。この 2 つの検索方法の違いとガーベッジ・コレクションとの関係を以下に述べる。

a. 検索スコープがクラスないしクラス階層の場合
この場合には、あるクラスの下に他のオブジェクトと全くリンクが張られていないオブジェクトが存在しても、そのオブジェクトは検索対象である。この場合、クラスは、RDB の表と同様に考えることができる。すなわち、表中のタプルは、それぞれ独立であるようにクラス中のオブジェクトは独立に存在しうる。従ってこのようなシステムではガーベッジ・コレクションは、永続性オブジェクト

[1] abort 処理が行なわれる可能性があるため、本当に削除するわけではないが、説明を簡単にするためこう書く

に関しては必要ないし、行なってはならない。従って、マルチメディア拡張に関しても特に問題はない。

b. 検索スコープが Collection の場合

こちらで RDB の表に対応するものは、Collection のインスタンスである。この場合、このインスタンスへの参照がなくなった段階でガーベッジ・コレクションが必要となる。GemStone はこの方式であり、自身が管理する基準空間内のオブジェクトのためのガーベッジ・コレクタを持っている。しかし、このガーベッジ・コレクタは拡張空間中のデータオブジェクトに対しては無効なため、拡張空間用のガーベッジ・コレクタを作成しなくてはならない。

拡張空間用のガーベッジ・コレクタの実現は、図5に示すように拡張空間のデータオブジェクトにオブジェクト ID を持たせておき、基準空間内に対応するオブジェクト ID を持つオブジェクトが存在するかどうかでチェックできる。一度使用したオブジェクト ID をガーベッジ・コレクション後再利用するシステムでは、同じオブジェクト ID を持つ基準空間内のオブジェクトと拡張空間中のデータとが本当に対応したものであるかを調べるための二次的なオブジェクト ID が必要である。

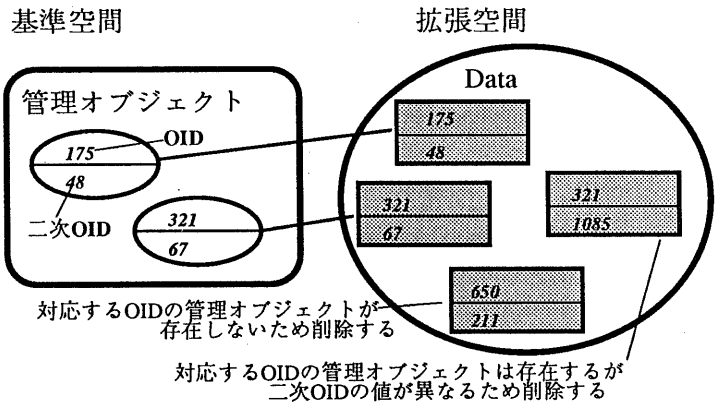



図5: 拡張空間用ガーベッジ・コレクタ

(2) トランザクション制御のための commit/abort 時の機能

OODBMS が、楽観的一貫性制御 (Optimistic Concurrency Control) を利用者に対して許すかどうかが、commit 機能の実現に影響を与える。abort 機能については、どちらのシステムでもまず拡張空間内のデータオブジェクトに対して abort 処理を行なってから OODBMS に abort 命令を発行すればよい。ここでは、以下、許すシステムと、許さないシステム双方の commit 機能について考察する。

i. 楽観的一貫性制御を許すシステム

このようなシステムでは、利用者が並行制御戦略として楽観的一貫性制御を選択した場合、トランザクションの commit 時に直列化可能性が判定される。そして、場合によっては、トランザクションを abort しなくてはならない。この場合、一旦 DBMS に対して commit 命令を発行した後でなければ、本当にオブジェクトが commit されるのかわからない。従って、媒体拡張を行なう時には、あるトランザクション内に生成、変更を行なった管理オブジェクトを Collection オブジェクト内に保存する。トランザクション終了時には、図6 に示すように、まずいったん DBMS に commit 命令を発行し、実際に commit されたことを確認してから Collection 内に記録された管理オブジェクトのデータを基に拡張空間中のデータオブジェクトの commit 処理を行ない、それが終了した後に Collection オブジェクトの中身を削除してもう一度 commit 処理を行なうことで対処している。

ii. 楽観的一貫性制御を許さないシステム

このようなシステムでは、利用者がアクセスしようとするデータに対して、自動的にロックがかかる[2]。従って、ハード障害時以外利用者が陽に行なう commit 命令が失敗することはない。そのため、媒体拡張においては、楽観的一貫性制御を許すシステムと同様、トランザクション内に生成、変更を行なった管

[2]例えば、ORACLE 等の商用 RDB など

理オブジェクトを保存する Collection オブジェクトを生成する。トランザクション終了時には、図 7に示すように まず、commit 命令の発行前にこの Collection オブジェクト内の管理オブジェクトに対応する拡張空間内データオブジェクトの commit 処理を行なう。そして、Collection オブジェクトの中身を削除した後、DBMS に commit 命令を送る。つまり、OODBMS に対して発行する commit 命令の回数が一回で済む。

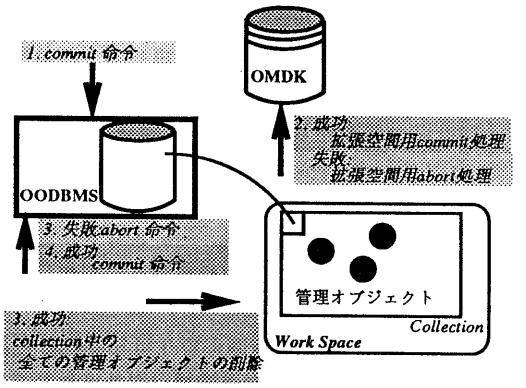



図6: 楽観的一貫性制御を許す
システムにおけるトランザクション制御

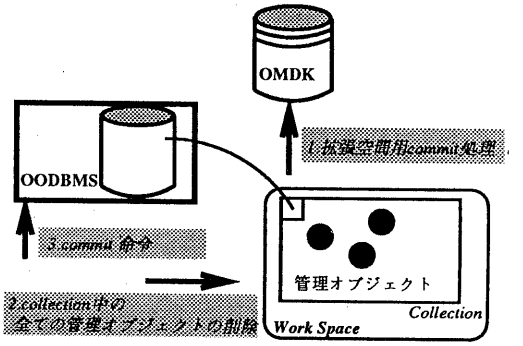



図7: 楽観的一貫性制御を許さない
システムにおけるトランザクション制御

## 7 マルチメディア拡張の際の OODBMS への要求事項

これまでで、既存 OODBMS を用いたマルチメディア DB 実現の方法を述べた。しかし、ユーザのニーズに十分に答えるマルチメディア DB を実現するためには、OODBMS が備えていて欲しい機能がある。その一つとして、分散環境における利用では、ユーザ定義のメソッドをどのマシンで動作させるか指定できる機能の必要性が指摘されている [Suzu 90]。

ここでは、5章で述べたアウトラインの方法を用いる時、これ以外に OODBMS が備えて欲しい機能について述べる。

### 7.1 インスタンス変数、メソッド に対する authorization

マルチメディア拡張を行なう場合、一部のインスタンス変数は、DBA のみが使用できるように制限する必要がある。従来の OODBMS ではこの機能をサポートしていないものが多い。この機能の必要性を具体例で説明する。

光磁気ディスク上のデータオブジェクトに対応する管理オブジェクトのためのクラス、OmdkClass を考える。この OmdkClass には、下記の定義に示すように利用者が利用するインスタンス変数 userData と、DBA が拡張空間中のデータオブジェクトとのリンクのために利用するインスタンス変数 control があるとする。

```
Class Definition {
     OmdkClass [:attributes
        control    " for DBA, to control data object in extended space "
        userData   " for user use "]
}
```

インスタンス変数 control に対しては、利用者はアクセスする必要はないし、間違ってアクセスした場合、拡張空間中のデータオブジェクトとこの管理オブジェクトとのリンク関係が崩れ、致命的な障害が発生する恐れがある。従って、このインスタンス変数に対しても authorization を行なって、特定のユーザのみがアクセスできるようにする機構が望まれる。

また OmdkClass には、光磁気ディスクにデータを書き込むメソッド "writeData:" が定義されているとする。OmdkClass のインスタンスを anOmdkObject としたとき、

```
anOmdkObject writeData: aData.
```

を実行すれば、このメソッド中では、anOmdkObject のインスタンス変数である control にアクセスし、その値を書き換える必要がある。しかし、上述のインスタンス変数に対する authorization がなされている場合には、メソッドの実行者が DBA でなければ、control にアクセスできない。したがって、このメソッドが実行される際にはその実行者を一時的にそのメソッド作成者にするなどの UNIX の実行モード指定のような機能が必要になる。

### 7.2 メッセージ送出機能

さまざまな操作に伴うメッセージ送出機能の存在によってマルチメディア拡張が容易になる。以下にそれを示す。

(1) commit, abort 前のメッセージ送出機能
OODBMS ではないが、永続性オブジェクトをサポートしたオブジェクト指向言語には、永続性オブジェクトを commit する際に commit 対象となるオブジェクト全てに対してメッセージを送る機構を有するものがある [Mitu 90]。この機構は、マルチメディア拡張に有益である。

つまり、commit 前のメッセージ送出機能がサポートされている場合、6.3の 2で述べたような collection による管理オブジェクトの管理が不要となる。すなわち、commit 時にシステムから管理オブジェクトにメッセージが送られた段階で、拡張空間中のデータオブジェクトの commit 処理を行えばよいわけである。

abort 時に関しても同様である。

(2) ガーベッジ・コレクタのメッセージ送出機能
6.3の 1 で述べたようにガーベッジ・コレクションを行なうシステムの場合、拡張空間中のデータに対するガーベッジ・コレクションは面倒な操作が必要である。この操作は、OODBMS に付随するガーベッジ・コレクタが、削除しようとするオブジェクトに対してメッセージを送ることで不要になる。すなわち、削除される管理オブジェクトがメッセージを受けとったら拡張空間のデータに対して削除処理を行なうようにメソッドを記述するだけで良くなる。

(3) 宣言的操作に伴うメッセージ送出機能
OODB に SQL のような宣言的操作を行なう必要性が指摘されている [Kim 90]。OODB にこの機能がサポートされた場合、INSERT や DELETE が、メソッド起動なしに実行されると、基準空間オブジェクトの生成、削除に伴う拡張空間上のデータに対する処理を行なうことができない。従って、オプションで良いから INSERT の際にはオブジェクトを生成するクラスに対して new メッセージを送り、DELETE の際には削除するオブジェクトに対して delete メッセージを送るような機能が望まれる。

## 8 検証システム

6章で述べた方法を検証するためにデモシステム (MULDIGRAS 〈マルディグラ〉:Multimedia Database Interactive Graphics System) を作成した。このデモシステムは、磁気ディスクまたは光磁気ディスク上にある静止画またはテキストからなる技術情報を検索・表示するシステムである。用いた OODBMS は GemStone である。これは、継承システムとして single inheritance であり、不要なオブジェクトは、ガーベッジ・コレクタによって削除を行ない、トランザクション制御としては、楽観的一貫性制御を許す。従って、管理オブジェクトのためのクラス階層は、図 4に準じたものであり、図 6に示す方法でトランザクション制御を行なった。

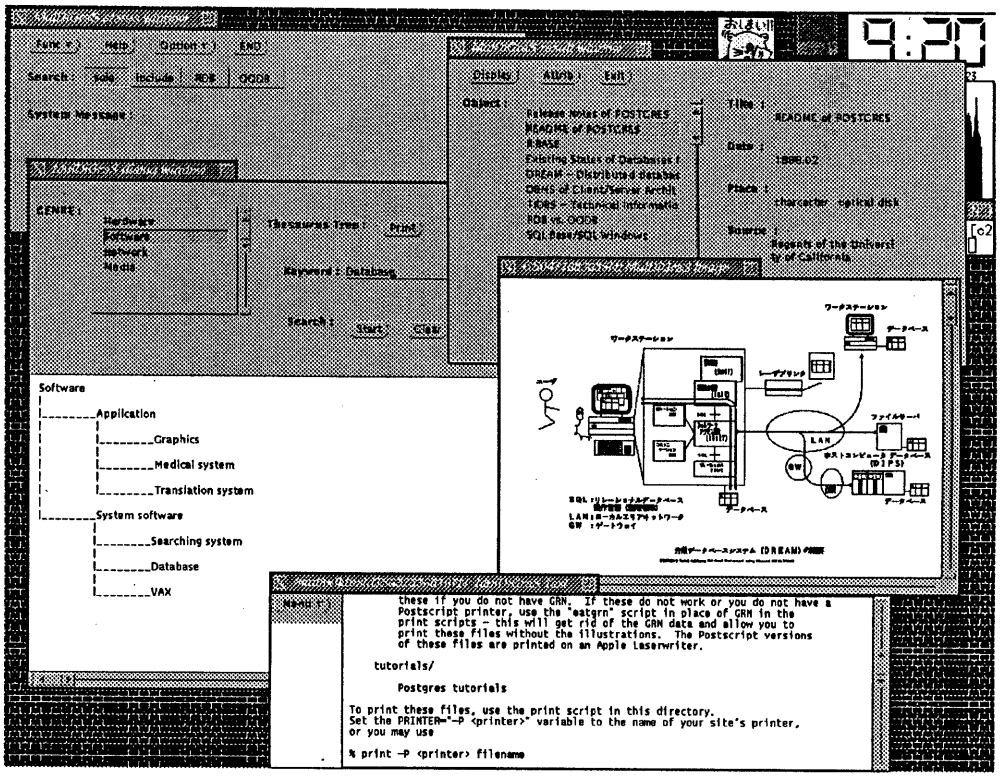
このシステムによって本稿で述べた方法の有効性を検証した。ただし、拡張空間用のガーベッジコレクタに関しては、まだ未検証である。このデモシステムの操作イメージを図 8に示す。

## 9 今後の課題

multiple inheritance を用いた管理オブジェクトの位置付け等、今回のデモシステムで未検証な点の検証、拡張空間データに対する障害回復機能の検討、時間同期情報が扱えるマルチメディアデータベース実現法の検討は、今後の課題である。

## 参考文献

[Bato 88] D.S.Batory, J.R.Barnett, J.F.Garza, K.P.Smith, K.Tsukuda, B.C.Twichell, and T.E.Wise. "GENESIS: An Extensible Database Management System," *IEEE Trans. onSoftware Engineering.* Vol.14, No.11, Nov.1988




[Bret 89] R.Bretl, D.Maier, A.Otis, J.Penney, B.Schuchardt, J.Stein, E.H.Williams, M.Williams,"Features of the ORION Object-Oriented Database System," *Object-Orented Concepts, Databases, and Applications,* Chapter 11, Addison-Wesley publishing co. 1989.

[Care 87] M.J.Carey and D.J.DeWitt. "An Overview of the EXODUS Project," *IEEE database Engineering,* Vol.6, No.2, June 1987

[Haya 90] 早川, "イメージ情報検索システムにおける光ディスク制御法の検討", 1990 年信学会秋季全国大会 D63

[Kato 90] 加藤、本橋、鈴木, "オブジェクト指向ＤＢにおける物理媒体制御法の検討," 情処学会第 41 回全国大会

[Kim 89] W.Kim, N.Ballou, H.Chou, J.F.Garza, D.Woelk, "Features of the ORION Object-Oriented Database System," *Object-Orented Concepts, Databases, and Applications,* Chapter 11, Addison-Wesley publishing co. 1989.

[Kim 90] W.Kim "Object-Oriented Databases: Definition and Research Directions," *IEEE trans. on knowledge and data engineering,* Vol.2, No.3, Sept. 1990

[Kuwa 90] 桑潴, "イメージ情報検索システムにおけるＲＤＢ利用方式", 1990 年信学会秋季全国大会 D62

[Moto 90a] 本橋、加藤、鈴木、"オブジェクト指向データベースのマルチメディアへの適用法の検討" 情処学会第 40 回全国大会

[Moto 90b] 本橋、加藤、鈴木, "拡張物理媒体の排他制御に関する検討", 1990 年信学会秋季全国大会 D61

[Mitu 90] 三ッ井 "COB における持続性オブジェクトの設計と実現," 情処研報 90-DBS-79

[Suzu 90] 鈴木 "マルチメディア応用から見たオブジェクト指向データベースへの期待," *Proceedings of Advanced Database System Symposium'90* Tokyo, Dec.5-6

[Turu 89] 鶴岡 木村 “オブジェクト指向データベース管理システムの物理仮想化方式,” 情処研報 DB システム 73-8, 1989,9

[Woel 87] D.Woelk and W.Kim, “Multimedia Information Management in an Object-Oriented Database System,” *Proceedings of the International Conference on Very Large Data Bases,* Brighton,England, Sept. 1987.