データベースシステム 94−22
（1993. 7. 23）

# オブジェクト指向データベースシステムにおける画像オブジェクトの構成と実装

脇山 俊一郎 † 大津 浩二 †† 福田 紀彦 †† 金森 吉成 †† 増永 良文 †††

† 仙台電波高専 †† 群馬大学 ††† 図書館情報大学

異機種分散環境で、特定のアプリケーション、動作環境にとらわれず、共有資源として有効利用が可能なマルチメディアデータベースを構築するための方策を、画像オブジェクトを例に検討した。メソッドについて、共通であるもの、アプリケーションに依存するもの、環境に依存するもの、など、それぞれの責任範囲と役割を徹底的に分析、分割することで、データベースに格納するオブジェクトの汎用性を高めることができた。また、データベース上のオブジェクトにアプリケーションに対する多態性を付加するための仕組みとして、メソッドの階層構造を導入し、メソッドを実行時に動的に結合する方法を提案した。

# An Organization and Implementation of Image Object on Object-Oriented Database System

Shunichiro Wakiyama † Kouji Ootsu †† Norihiko Fukuda ††
Yoshinari Kanamori †† Yoshifumi Masunaga †††

† Sendai National College of Technology
†† Gunma University
††† University of Library and Information Science

We have examined how to organize reusable and generic image objects, which are stored in object-oriented database system under heterogeneous distributed computer environments. Generally, the organization of image objects tends to depend on applications and their working environments. We have analyzed the methods of image object from the viewpoint of roles and responsibilities in the applications, and divided them into generic and specific methods. These methods make a hierarchical structure, and are dinamically linked to the object in the database at the run time.

# 1 はじめに

オブジェクト指向データベースシステムは、関係型などの従来のデータベースシステムに比べ、マルチメディアデータの取扱いに適しているといわれており [1]、オブジェクト指向データベースシステムを利用したマルチメディアデータベースの構築が多数試みられている。

我々は、画像を主とする医療分野向けデータベースシステムを例に、オブジェクト指向データベースシステムによるマルチメディアデータベース構築の研究を行っている [2] [3] [4] [5]。これまでの研究から、オブジェクト指向データベースシステムやマルチメディアデータベースは、ともすると特定のアプリケーションや動作環境に依存した実装になってしまうことがわかった。これは、共有資源たるデータベースとしての利点を損なうことにほかならない。

データベースは、従来それぞれのプログラムで独自に利用、管理していたデータをプログラムから分離し、そのデータを一括管理するとともに、共有資源として多数のユーザ、アプリケーションに供する仕組みを提供した。これによりデータの有効利用が図られ、またアプリケーション毎にデータ管理部分を構築する手間を省くことが可能になった。オブジェクト指向のマルチメディアデータベースシステムにおいても同様に、データの有効利用やアプリケーション作成時の負担軽減を図る機能が提供されるべきである。

現代の計算機環境は、オープンシステム化によって、様々な種類の計算機がネットワークで接続された異機種分散環境へと移行してきている。したがって、データベースが提供すべき諸機能は、異機種分散環境においても有効なものでなければならない。

本論文では、マルチメディアの一例として静止画像を取り上げ、アプリケーションや動作環境に依存しないデータベースの構築法を、画像オブジェクトの構成、アプリケーションの構造などの観点から議論する。

# 2 分散環境におけるマルチメディアデータベースの利用

分散環境では、それぞれの計算機に機能を割り振り、それらを組み合わせてタスクを処理するような、分散コンピューティングが可能になる。したがって、データベースシステムも分散環境の中の一つのサブシステムとして、サービスを提供できるようにしなければならない。

異機種分散環境では、計算機のハードウエア、オペレーティングシステムをはじめとするソフトウエアは多種多様で、それぞれの計算機上に実現されているサービスを提供し合うには、仕様の差異を吸収し、インタフェースを共通化する仕組みが必要になる。

OMG (Object Management Group) が提案している CORBA (Common Object Request Broker Architecture)[6] では、オブジェクト指向の概念をクライアントサーバモデルに適用し、クライアントがサーバのオブジェクトにサービスを要求するという形態をとることで、異機種分散環境での相互運用性を実現している [7]。すなわち、オブジェクトのカプセル化の概念で、ハードウエア、ソフトウエアの機種依存部分を隠蔽し、メッセージによってインタフェースを共通化している。

ここでは CORBA を念頭に、異機種分散環境でのマルチメディアデータベースのあるべき姿を、(1) アプリケーション独立、(2) 動作環境独立、の 2 つの視点から明らかにする。

## 2.1 マルチメディアオブジェクトのアプリケーション独立

データベースは様々な目的を持ったユーザが、各自の目的にあったアプリケーションを利用して、自由にアクセスできるものでなければならない (図 1)。

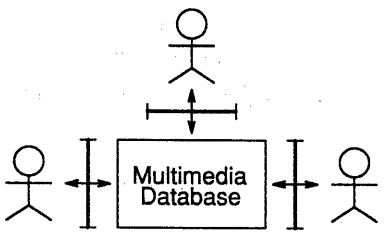


図 1: 共有資源としてのマルチメディアデータベース

オブジェクト指向データベースシステムでは、データとそれを操作するメソッドをカプセル化し、オブジェクトという形態で格納する。このとき、オブジェクトのメソッドとしてどのような処理機能を付加するかが、オブジェクトの汎用性を大きく左右する。

共有資源としてのデータベースを保持するためには、アプリケーション固有の処理をオブジェクトのメソッドとして登録することを避け、オブジェクトのアプリケーション独立を確保しなければならない。

一方、マルチメディア固有の基本的な処理は、オブジェクトのメソッドとして登録することが好ましい。画像オブジェクトを例にとると、画像の拡大·縮小、色の量子化 (減色処理) といった処理は、どのようなアプリケーションを作成する際にも不可欠である。これらをオブジェクトのメソッドとして採用することにより、アプリケーションのマルチメディア処理部分のコーディングが簡素化され、アプリケーションプログラマの負担が軽減される。

このような方針でオブジェクトを設計することにより、特定のアプリケーションに依存せず、しかもアプリケーション作成が容易なマルチメディアデータベースが実現できる。

## 2.2 マルチメディアオブジェクトの動作環境独立

異機種分散環境では、マルチメディアデータベースを利用するアプリケーションの動作環境を特定することができない。したがって、マルチメディアオブジェクトを設計する際には、動作環境に依存する部分について検討する必要がある。

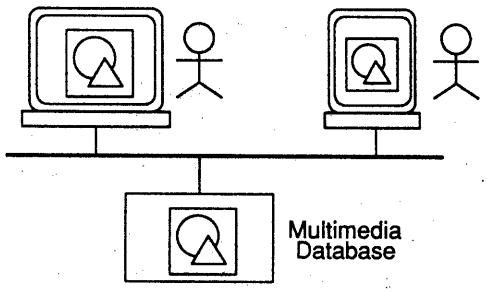


図 2: 異機種分散環境でのマルチメディアデータベース

### 2.2.1 ユーザインタフェースの分離

動作環境依存の一つは、GUI に代表されるユーザインタフェースである。ワークステーションでは X-Window システムが最も一般的であるが、OPEN LOOK や Motif などツールキットには様々なバリエーションがある。また、パーソナルコンピュータでは、PC 系の Microsoft Windows や Apple の Macintosh の環境がある。

ユーザインタフェース部分のコーディングはアプリケーションプログラマにとっては大きな負担であるため、共通のコードとしてマルチメディアオブジェクトの中に組み入れ、マルチメディアとユーザインタフェースを一体化してしまうことも考えられる。しかし、異機種分散環境ではユーザインタフェースが多種多様であるため、これは不可能である。

また、仮に単一環境を想定したとしても、GUIのようなイベント駆動型のプログラムスタイルは、かなり下位レベルでの制御となってしまうため[8]、オブジェクト間のメッセージ交換のような上位レベルでの制御には必ずしも適していない。結局、オブジェクトへのユーザインタフェースの組み込みは、得策とは言えない。

### 2.2.2 デバイスとの整合

ユーザインタフェースの問題はアプリケーションソフトウエア一般に言えることであるが、マルチメディアでは特有の動作環境依存の問題がある。

マルチメディアはデータ型であるメディアと、それを具象化するハードウエアであるデバイスからなる[9]。この両者間の整合が取れてはじめて、マルチメディアがユーザに提示される。

多くのマルチメディアシステムは、動作環境を予め特定しているため、メディアとデバイスの整合について特に考慮する必要はなかった。ところが、異機種分散環境では、データベースに格納されているマルチメディアデータが、そのままアプリケーションの動作環境のもとで具象化できる保証はない。そこで、メディアとデバイスの整合をとるためのマルチメディア処理が必要になる[5]。

例として画像を挙げると、1画素24bitのフルカラーのデータを256色疑似カラーや白黒のディスプレイに表示するための整合処理は、画像オブジェクトのメソッドとして持たせた色の量子化処理で行なうことができる。

## 3 画像オブジェクトの構成

マルチメディアの一例としてカラー静止画像を取り上げ、異機種分散環境での画像オブジェクトの構成法について議論する。

### 3.1 画像オブジェクトのユーザモデル

アプリケーションユーザにとって、画像オブジェクトは画面に表示されている画像そのものである(図3)。このオブジェクトをユーザが直接操作[10]できる環境をシステムが提供しなければならない。

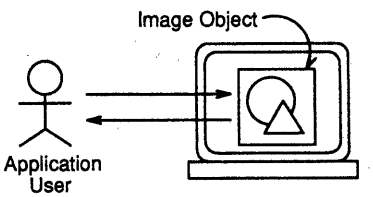


図3: 画像オブジェクトのユーザモデル

アプリケーションユーザが画像オブジェクトを直接操作する際には、次の過程がある。

(1) ユーザのリクエストを受け付ける

(2) 画像処理を行う

(3) 処理結果を画面に描画する

ここで、(1)と(3)はユーザインタフェースが担当すべき役割であり、本来、画像オ

ブジェクトが成すべき役割は (2) のみである。しかも (2) の内容はアプリケーションの内容によって異なってくる。

つまり、アプリケーションユーザが認識している画像オブジェクトは、ユーザインタフェースと融合し、しかもアプリケーション、動作環境に強く依存した仮想的なものであると言える。

## 3.2 ユーザモデルの整理・分析

2章での議論で、データベースに格納されるマルチメディアオブジェクトは、アプリケーションや動作環境に依存しないものでなければならないことを示した。また、ユーザインタフェース部分は分離すべきであることも述べた。

これに沿って、ユーザモデルを整理・分析すると、ユーザレベルでの仮想的な画像オブジェクトが、ユーザインタフェースオブジェクトと画像オブジェクトの2つのオブジェクトに分離される。

ユーザレベルでの仮想的な画像オブジェクトの振舞いは、

(1) ユーザインタフェースオブジェクトがユーザのリクエストを受け付け、それを画像オブジェクトに伝える。

(2) 画像オブジェクトは要求された画像処理を行う。

(3) ユーザインタフェースオブジェクトは画像オブジェクトの処理結果を画面に描画する。

と書き換えられ、それぞれのオブジェクトの責任範囲も明確になる。

## 3.3 画像オブジェクトの多態性

実際にオブジェクトとして存在するのは、動作環境依存のユーザインタフェースオブジェクトとデータベース上の画像オブジェクトの2つである。データベース上の画像オブジェクトは基本的なメソッドしか持たないが、3.2節での画像オブジェクトは、アプリケーションとしての処理も行なっており、その点でなお、仮想的な存在に留まっている。

アプリケーションの処理を、それがあたかもデータベース上の画像オブジェクトのメソッドでおこなわれているかのごとく見せることにより、画像オブジェクトのアプリケーションに対する多態性が生まれる。

データベースが共有資源として多くのアプリケーションから利用できるようにするため、データベースのオブジェクトからアプリケーション依存の部分を取り去った。今度は取り去ったアプリケーション依存部をどのようにしてデータベース上のオブジェクトに付加するかを検討しなければならない。これについては、3.5節で議論する。

## 3.4 メソッドの階層構造

一般のユーザやアプリケーションプログラマは、マルチメディア固有の複雑なデータ処理に不案内であることが多い。マルチメディア処理についての詳しい知識がなくとも、簡単にアプリケーションを作ることができる仕組みを提供することが、共有資源としてのマルチメディアデータベースの有効利用につながる。ここではその仕組みとして、メソッドの階層構造と一般的な画像処理を行なう画像処理サーバの導入を考える。

アプリケーションユーザに見えている画像オブジェクトは、図4のような四層の階層構造によって仮想的に実現される。

Database Object Layerでは、データベース上の画像オブジェクトにより画像の拡大・縮小、色の量子化などの基本的な機能が提

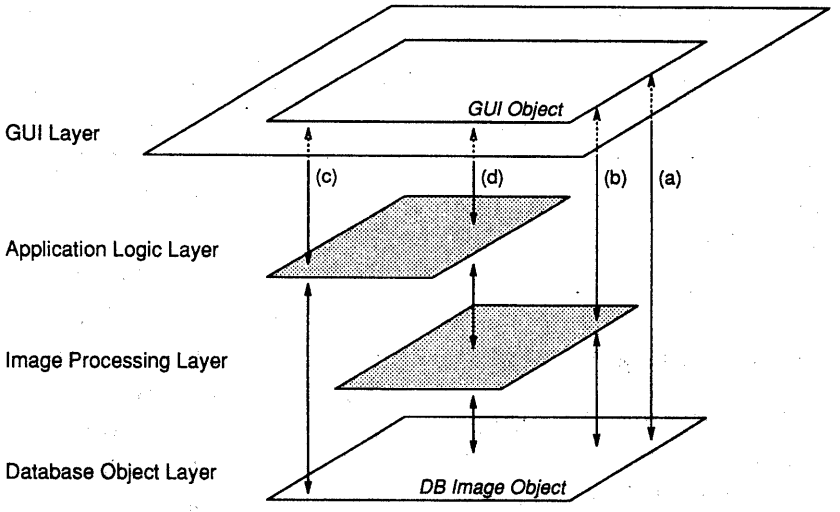


図 4: メソッドの階層構造

供される。また、GUI Layer では、画像表示用ウインドウやオブジェクトへのリクエストを発行するユーザインタフェースが提供される。

Application Logic Layer と Image Processing Layer は、オブジェクトの形式は取らず、処理機能だけが提供される。

Application Logic Layer では、それぞれのアプリケーション固有の処理機能が提供され、一方、Image Processing Layer では、エッジ検出、$\gamma$ 補正、各種フィルタリングなどの一般的な画像処理機能が画像処理サーバによって提供される。

各階層での機能を組み合わせることにより、図 4(a)〜(d) の 4 つの処理形態でアプリケーションを作ることができる。これらを我々が研究している歯科矯正学向けの医療画像データベースを例に、具体的に説明する。

(a) 画像を単に表示する
データベース上の画像オブジェクトを動作環境下の画像表示ウインドウに描画する。画像表示ウインドウのサイズや使用可能な色数は、画像オブジェクトの持つデータとは必ずしも一致しないので、画像オブジェクトの基本メソッドにより、拡大・縮小、色の量子化処理を行なって整合を取る。

(b) 一般的な画像処理を施して表示する
データベースに格納されている画像の中には、撮影時の条件不良などで他の画像に比べ色調が良くないものもある。そのような画像を表示するときは、Image Processing Layer で $\gamma$ 補正を行ない、色調を整えたうえで画像表示ウインドウに描画する。

(c) アプリケーション固有の処理を施して表示する
我々の医療画像データベースでは、患者の横顔の写真と X 線写真からトレースした線図を重ね合わせるという、アプリケーション固有の処理がある。横顔の写真をユーザに提示し、重ね合わ

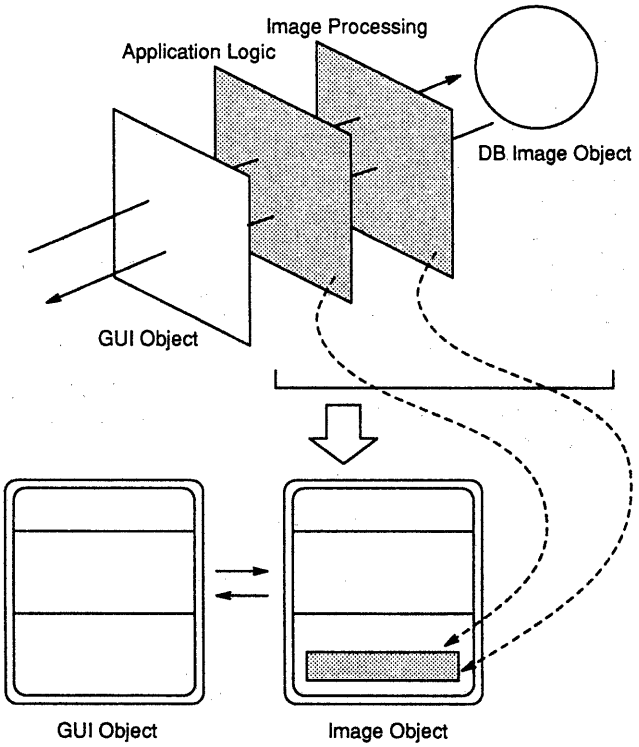


図 5: メソッドの動的結合

せの基準点をマウスで入力してもらい、それに合わせて線図オブジェクトの基本メソッドにより拡大・縮小・回転処理を施し、横顔の写真に重ねて線図を描画する。

(d) 一般的な画像処理を必要とするアプリケーション固有の処理を施して表示する
(1) 治療履歴に沿って患者の正面顔写真を並べて表示する際、目の高さで揃えたい、(2) 患者のプライバシー保護のため、第三者に正面顔写真を見せる時は目線を塗りつぶしたい、(3) 顔の輪郭線を抽出し、それらを重ね合わせたい、などのアプリケーション固有の処理がある。これらは、エッジ検出などの一般的な画像処理を行ない、そこで得られた情報をもとに、さらにアプリケーションレベルで処理を重ねることになる。

## 3.5 メソッドの動的結合

メソッドの階層構造によって、データベース上の画像オブジェクトにアプリケーション依存の処理や一般的な画像処理機能を付加し、アプリケーションに特化した振舞いをもつ画像オブジェクトを仮想的に作り出すことが可能になる。

仮想的な画像オブジェクトは、図 5上のようにGUIオブジェクトとデータベース上

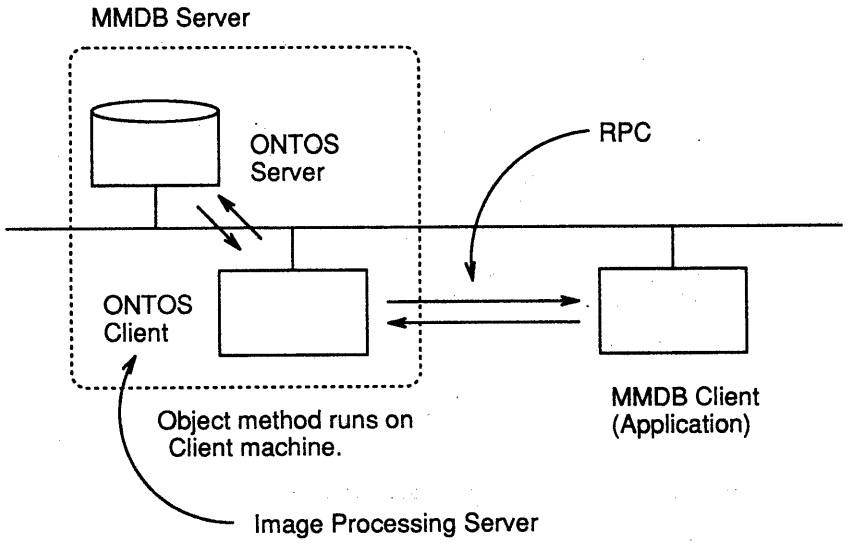


図 6: ONTOS による実装

の画像オブジェクトの間にフィルタとして Image Processing Layer、Application Logic Layer の機能を挿入するという実現法も考えられるが、オブジェクト指向プログラミングを貫くためには、図 5下[1]のようにそれぞれの追加機能が必要に応じて画像オブジェクトのメソッドとして実行時に動的にリンクされる方が好ましい。

3.4 節で例示した (a)～(d) は、

(1) GUI オブジェクトがユーザのリクエストを受け付け、それを画像オブジェクトに伝える。

(2) 画像オブジェクトは要求された処理を行うために、必要なメソッドを動的に結合し実行する。

(3) GUI オブジェクトは画像オブジェクトの処理結果を画面に描画する。

という形で実現される。

[1]Coad/Yourdon 法 [11] の記法による

# 4 実装

我々はオブジェクト指向データベース管理システムとして米国 Ontologic 社の ONTOS を利用してきた。今回も ONTOS での実装を試みる。

## 4.1 CORBA 的環境の構築

ONTOS はクライアント・サーバモデルに基づいて動作する。ONTOS ではオブジェクトのメソッドを実行形式のメモリイメージで持ち、それを動的に結合させ、クライアント側で実行する。このため、ONTOS のクライアント・サーバは同じアーキテクチャのマシンでなければ動かないという問題がある。これでは異機種分散環境への対応が困難である。

異機種分散環境に対応するために、図 6 のように ONTOS のクライアント・サーバをひとまとめにして MMDB (MultiMedia DataBase) サーバと位置づける。MMDB サ

ーバ上には、データベースに格納された汎用の画像オブジェクトがアプリケーションに必要なメソッドが結合された形で活性化される。一方、MMDB クライアントはアプリケーション実行のためのユーザインタフェースオブジェクトを自分の動作環境の上に生成する。双方のオブジェクトがネットワークを介してメッセージを交換することによってアプリケーションの実行がなされる。

CORBA では、リモート手続き呼び出し (Remote Procedure Call - RPC) により、オブジェクト相互のメソッド起動を実現している。同様のメカニズムを MMDB サーバとクライアントに実装することで、実験システムの構築が可能である。

CORBA 環境が利用できるようになれば、Image Processing Layer の一般的な画像処理は、画像処理サーバという共通ファシリティの一つとして実装するのが好ましい [5]。実験システムでは Image Processing Layer のメソッドも ONTOS クライアント上で動的に結合されるため、MMDB サーバが画像処理サーバも兼ねることになる。

## 4.2 ONTOS でのメソッドの動的結合

ONTOS ではメソッドの動的結合のための仕組みをいくつか用意している [12]。

Programmatic function invocation は、例えばメニューで処理項目を選び、その項目名を使って関数を起動するような、イベント駆動型のプログラミングに便利な機能である。

また、method table を用いた関数起動では、同じ名前に割り当てられた複数の関数を選択的に起動することが可能である。

これらの機能を活用して、Image Processing Layer の一般的な画像処理機能や Application Logic Layer の処理を、データベース上の画像オブジェクトに動的に結合、実行することができる。

# 5 むすび

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、いずれも問題向きに構築される例が多く、汎用性を論じたものはあまりない。

今回、我々は異機種分散環境で、特定のアプリケーション、動作環境にとらわれず、共有資源として有効利用が可能なマルチメディアデータベースを構築するための方策を、画像オブジェクトを例に検討した。

メソッドについて、共通であるもの、アプリケーションに依存するもの、環境に依存するもの、など、それぞれの責任範囲と役割を徹底的に分析、分割することで、データベースに格納するオブジェクトの汎用性を高めることができた。

また、データベース上のオブジェクトにアプリケーションに対する多態性を付加するための仕組みとして、メソッドの階層構造を導入し、メソッドを実行時に動的に結合する方法を提案した。

今後は、4 章に示した実験システムを ONTOS をベースに構築し、今回提案した手法の有効性を実地に検証していきたい。

# 参考文献


[1] 増永良文：次世代データベースシステムとしてのオブジェクト指向データベースシステム，情報処理,Vol.32,No.5,pp.490-499 (1991).

[2] 金森吉成、脇山俊一郎、増永良文：オブジェクト指向データベースの医療への応用，情報処理学会データベースシステム研究会資料,86-1(1991).

[3] 脇山俊一郎、金森吉成、増永良文：オブジェクト指向データベースシステムによる医療画像データベースの実装，情報処理学会データベースシステム研究会資料,87-10(1992).

[4] 金森吉成、脇山俊一郎、増永良文：マルチメディアデータベースのオブジェクト構成とそのユーザインタフェースの設計，情報処理学会データベースシステム研究会資料,89-4(1992).

[5] 脇山俊一郎、金森吉成、増永良文，布川博士：ORB プラットフォーム上でのマルチメディアオブジェクトの構成法，電子情報通信学会 信学技報 DE92-35(1992).

[6] Object Management Group: The Common Object Request Broker: Architecture and Specification, OMG Document Number 91.12.1 Revision 1.1, Draft 10 December 1991

[7] 大野邦夫：異機種分散環境とオブジェクト指向 -CORBA の概要と今後の展望-，アドバンスト・データベースシステム・シンポジウム '92 論文集，pp.121-135 (1992).

[8] 白田由香利、飯沢篤志：ウインドーベースのデータベース操作用アプリケーションの開発システム、情報処理学会データベースシステム研究会資料、92-10 (1993)

[9] 小坂一也、梶谷浩一、何千山、森本康彦、福田剛志：オブジェクトサーバとその応用、情報処理学会データベースシステム研究会資料、89-3 (1992)

[10] Ben Shneiderman : Designing the User Interface, Addison-Wesley,Menlo Park (1987).

[11] Peter Coad, Edward Yourdon : Object-Oriented Analysis (Second Edition), Prentice-Hall(1991).

[12] ONTOS DB 2.2 Developer's Guide, ONTOS(1992).