データベース・システム 86－5
（1991. 11. 8）

オブジェクト管理用データベースシステム Zodiac-1 の概要

田沼 均

電子技術総合研究所

本論文では現在開発中のオブジェクト管理用データベースシステム Zodiac-1 の概要について述べる。

Zodiac-1 の特徴は 3 つある。一つはカーネル化したデータベースシステムであること。そのためシステムは非常にコンパクトに作られている。二番目は開放型システムであること。分散 OS の外部ファイルシステム機能を利用してあらゆるシステムと協調可能である。三番目は拡張可能なシステムであること。システムを拡張することによって様々な種類の情報を管理することができる。

A Database System for Objects Management: Zodiac-1 its Overview

Hitoshi Tanuma

Electrotechnical Laboratory

We are developing a database system for objects management. We call it Zodiac-1 and this paper reports overview of this system.

Zodiac-1 has three features. First, it is a kernelized database system and very compect. Second, it is a open system. It uses the external pager functin of recent distributed operating system and co-operate with other systems. Third, it is a extensible system. It can manage many kind of information to extent system.

1 はじめに

最近の計算機システムの顕著な変化の一つはネットワーク化である。それまでは大小を問わず計算機システムは相互に独立して作られ運用されてきた。最近では卓上にあるパーソナルコンピュータから共用の大型コンピュータ、スーパーコンピュータにいたるまで各種の計算機がLANに接続している。今や日本でもそのLANを組織や国を越えて接続するインターネットの構築がすすめられている。接続の初期は通信速度も遅くネットワークを通じて行なえる機能もかなり限られたものではあったが、徐々に通信速度も速くなりネットワークを通じて行なわれる機能も高度化してきた。

計算機システムがネットワークによって相互に接続されると利用可能なデータの量は飛躍的に増大する。ネットワーク化の一つの目的がお互いに蓄積したデータを相互に利用することにあるから、相互に接続した計算機システムに蓄えられているデータの総和だけデータが利用可能になるのは当然である。さらにお互い参照可能なデータを整理して統合することにより今まで蓄積してきた以上のデータを提供できるようになる。

特にネットワーク化することによってシステム内に存在するオブジェクトを管理するためのデータの増加量は顕著である。個々の計算機の中だけで閉じていた場合は、管理の対象となるものはそれほど多くはなく、管理の構造も比較的単純で管理情報は少なくて済んだ。しかしネットワーク化により管理の対象は飛躍的に増えた。ネットワークが単純なうちは構造も単純であるが、ネットワークが広がり例えば組織や国の境を越えたりする場合、その様々な制約や要請により管理の構造も非常に複雑化し、様々な管理のレベルや管理の手法が共存することとなる。その結果十分な管理を行なうための管理情報の必要量は通常のデータの増加量以上の増え方となり、情報の質も複雑となる。

さらにシステム内のオブジェクトを管理する情報は通常のデータの比較して共有されている機会が多い。システム管理情報は計算機システム内でシステムが起動されてからそのシステムの運用のために様々な機会にアクセスされる。またシステムの状態が変わるごとに頻繁に更新される必要がある。オブジェクト管理用の情報は共有される場合が多いという事実はシステム管理情報のデータ量の増大、構造の複雑さと共に管理システムを作る上で十分注意する必要がある。

ところで現在は多くの場合システム管理用のデータの管理には専用のデータサーバが使用される。例えばTCP/IPの世界では、ネットワークアドレスを調べるためのネームサーバであるDomain Name Serverやネットワーク管理に使用するMIB(Management Information Base)、OSIの世界のThe Directory、UNIXのワークステーションで構成されるLANで良く使用されるSun Microsystemsが開発したNISなどである。

これらのデータサーバはある目的のために専用に作られている。そのためデータの種類ごとにそれぞれ別のデータサーバが存在し、情報を管理されているサーバに応じて異なるサーバにアクセスする必要があると共に、同一のものを表す場合でも表現方法やアクセス方法が異なり全体として繁雑な管理となる。また多くの場合、通信関係の使用が主であるため、データサーバの設計がアクセスのためのプロトコルに中心がおかれている。サーバのモデル記述能力にはそれほど配慮が払われず、使用されるデータモデルはそれほど強力なものではない。

システムを管理する情報はそれぞれ目的別に個別に管理するよりは全体を統合して管理する方が良い。情報は統合することによりさらに情報が増え、より詳細で便利な管理を行なうことが可能となる。従ってシステムを管理している情報も含めて全ての情報を、管理能力が強力なデータベースシステムで管理する必要がある。

2 既存データベース管理システムの問題点

システムのオブジェクトを管理するために既存のデータベース管理システムを使おうと

すると次のような問題が生じる。

2.1 システムの肥大化

現在の汎用データベースシステムは、様々なアプリケーションに対応するために多くの機能を備えている。データベースシステムが単独で多種多様な機能を提供するため、システム内にそれら全ての機能を実現し、システムが膨張してゆき巨大なシステムが出来上がる。

システムが膨張するとそれにつれて記憶領域やCPUといった計算機資源を多く消費することとなる。管理システムはアプリケーションシステムの背後にあって、システムの運用を円滑にするために奉仕する存在でなければならない。この管理システムが多くの資源が必要になると本来の目的であるアプリケーションシステムに十分な資源の割当を割けず、実行の妨げになる。

実際に存在するデータベースシステムの多くには、ある応用では不要である機能、実装してあるほどの強力さを必要としない機能、また既に他のシステムで実現されている機能などがあり、必ずしも実装されている機能全てが常に必要とは考えられない。例えば多くのデータベースシステムではアクセスのために専用の言語を備えている。ユーザが直接データベースにアクセスするか、データベース設計の過程で試作のために直接データベースを操作する場合は強力なデータベース操作言語が必要とする。しかし計算機内でシステムの管理のためなどに他のシステムからデータベース機能を利用する場合、必ずしも強力な言語は必要ではない。データベース言語にしても言語の処理のためのコストは無視できず、強力な言語は強力であるほど障害になることがある。またトランザクション制御機能はデータベースシステムにおいて重要な機能の一つである。トランザクションの制御には固有の幾つかの機能が必要ではあるが、効率的にトランザクションの管理を行なうにはシステムやデータの構造や動作といった管理情報が不可欠である。その意味でトランザクション制御機能は共通的な最小セットの機能とデータモデルのに合わせた制御を行なう部分とに分けて考えられ、後者は具体的なデータベースを設計する際にデータのモデルを使って実現された方が効率の良いものができる。

オブジェクトを管理するデータベースとしてはシステムの肥大化は不都合であり、できるだけ機能を絞ったカーネル化されたシステムが望ましい。

2.2 閉鎖性

管理システムには管理する対象がある。この管理される対象には管理のされ方とは別に本来果たすべき機能を持っている。これらの機能は多くの場合、データベースシステム以外のところで実現される。ユーザが直接データベースにアクセスして検索する以外は多くの場合、検索などにより得られた情報はすぐにデータベースシステムの外部にある別のシステムで利用される。特に管理情報の場合では得られた情報をデータベースシステムまたは外部のシステムがそれぞれの行なう管理のために利用する。

ところが多くのデータベースシステムでは内部に蓄えてあるデータを外部の別のシステムに渡すために非常なコストがかかる。外部システムは直接データベースシステム内のデータにはさわることができず、データベースアクセス用言語やデータベースの用意したアクセスインターフェースを使わざろうえないことが多い。多くの場合これらの方法はデータをまとめて取り出すことはできない。従って大きな量の情報を取り出そうとすると非常なコストがかかってしまう。

2.3 拡張性の不足

現在のデータベースシステムは拡張性に乏しい。これはシステムを閉鎖的に作っていることとも関係がある。多くのシステムではシステムの能力が不足していても、システムを拡張することは難しい。外部からのデータベースシステムのアクセス機能の不足と共に、データベースシステムから外部システムにアクセス機能も不足している。

しかしデータベースシステムにあらゆる機能を装備することは不可能であるから何らかの拡張機能が必要である。特にシステムの管理を行なう場合、扱う可能性のある情報は非常に多様性に富んでいる。このため様々なタイプの様々な構造を持つデータに対応するには強力な拡張機能が不可欠である。

2.4 オブジェクトに対するアクセス

システム管理用の情報は、その管理の対象になっているオブジェクトと密接に結びついている。システム管理用の情報への情報の検索や更新などの操作は関連するオブジェクトへの操作を伴うことが多い。管理の対象とするオブジェクトの情報とともにそのオブジェクトへのアクセスなどの操作もともに管理されている方が便利である。

3 Zodiac-1 の位置付け

Zodiac-1 は計算機システム中のオブジェクトを管理するためのデータベースシステムの中核となるシステムである。Zodiac-1 を使ったシステムは Zodiac-1 を単独で使用するのではなく、常に他のシステムと組み合わせて使用される。システムはサブシステムの組合せである複合システムとなる。

システムの設計やプログラミングの方法論として、特定の機能を持った小規模な単位のモジュールを組み合わせて全体のシステムを作り上げることは一般的である。複合システムはこの手法と同様にそれぞれ固有の機能を持つシステムを組み合わせて、目的の機能を果たすシステムを構築する。

複合システムを構成するサブシステムは二つのタイプに分けられる。汎用的な機能を持ちほとんど全てのシステムに使用されるカーネルシステムと、アプリケーションの目的に合わせてカーネルシステムを拡張する拡張システムである。例えば OS はカーネルシステムに入り、アプリケーションシステムとして直接ユーザが使用する部分は拡張システムの部分である。

Zodiac-1 は複合システムの中で永続的なオブジェクトの管理を行なうためのカーネルシステムである。Zodiac-1 はデータベースの機能が使われる場合に必ず使われる機能をサポートするシステムである。データベース機能はシステムの管理から実際のアプリケーションまであらゆる情報システムで使われる機能である。

OS、特に最近開発された分散システム OS のカーネルも複合システムのカーネルの一つである。OS のカーネルは主にシステムの動的な側面、つまり処理の単位となるプロセスの資源と相互作用を管理するシステムである。Zodiac-1 も実行はこの上で行なわれる。Zodiac-1 は幾つかのプロセスによって実行され、プロセス相互の通信は OS カーネルの機能を通じて行なわれる。しかし OS がシステムの管理などのために蓄積されているオブジェクトや情報を利用しようとした場合、今度は Zodiac-1 の機能を利用することとなる。従来はこの機能はファイルシステムが受け持ってきた。その意味では OS から見た時、Zodiac-1 はファイルシステムとなる。

4 Zodiac-1 のデータモデル

Zodiac-1 のデータモデルの目標は、拡張性と開放性を備えた単純で柔軟なモデルである。システムの基本構造の上から外部システムの存在が仮定されている。具体的に Zodiac-1 を使ってシステムを構成する時はデータモデルの上でも、Zodiac-1 の基本機能として管理する部分とモデルを拡張しその拡張部分が扱う部分および基本機能と拡張機能との間のインターフェースの部分が存在し、それぞれが明確に分離可能である。

関係データベースは現在もっとも一般的なデータベースである。理論的にも整理されており体系としても強力である。また関係データベースについては過去二十年近くにわたって多くの研究がなされ、理論的にも技術的にも多くの成果があげられている。もちろん関係データベースも幾つかの欠点は持っているが、データベースシステムとして関係データベースを全く捨て去ってしまうことは得策ではないと考える。Zodiac-1 は関係データベースの技術を継承できるように作ってある。適

当な拡張法をとればZodiac-1で関係データベースシステムを構成することができる。

システムは常に全部の情報を見せたり全ての操作を許したりしているわけではなく、状況に応じて情報へのアクセスや操作を制限している。また状況によっては同じ情報でも異なった表現方法がとられる。このような現在の情報に対する見え方を制御しているものをアスペクトと呼ぶ。アスペクトは現時点でどの情報がどのように見えるかといった、情報の見え方に関する環境である。アスペクトには次の二つの機能がある。

- 情報のアクセスや操作の制限。ある時点でどの情報にどの程度アクセスできるかどの程度操作できるかを決める。この目的は二つある。一つは安全性の問題である。データベースの目的がデータの共有であることから、全てのものに無制限のアクセスや操作を許すのではなく、適切な状況に応じたアクセスや操作の許可を与える必要がある。このコントロールのためにアスペクトを使う。もう一つは情報のワーキングセット化のためである。膨大な量の情報を一時期に扱うのは人間の識別性の点からも計算機の効率からも好ましくない。実際にある時点で扱っている情報はその近傍の限られた範囲であることが多い。この限られた範囲を制御するのもアスペクトの役割である。

- 情報の提示の制御。情報はその処理の目的や方法によって異なる見方が要求されることがある。同じ情報でも表現の形式が異なる方法をとらなければならないことがある。これらの環境を制御するためにアスペクトが使われる。

Zodiac-1が扱うオブジェクトは次のようにモデル化される。

- オブジェクトは個々に識別が可能であり、個々に何らかのまとまりを持ったものである。

- オブジェクトは必ず一つ以上のクラスに属する。

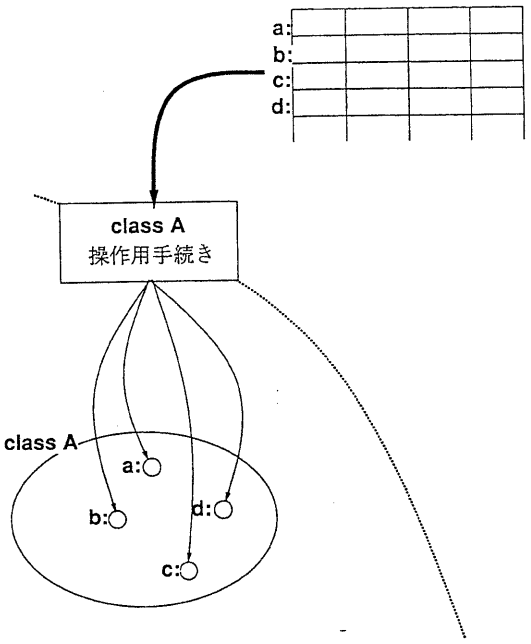

図 1: Zodiac-1 のデータモデル

- 同じクラスに属するオブジェクトは属するクラスで定義された同じ手続き群によりアクセス、操作される。

- 同じクラスに属するオブジェクトは、同一種類の属性を持つ。属性値はオブジェクトの性質を示す。

属性値は文字列または数値かこれらのサブタイプなどの単純値であり、複雑な構造はとらない。

オブジェクトは以上の条件を満たす限り内部において、どのような構造を持ちどのような動作を行なうかは自由である。つまりここで規定したオブジェクトはあるクラスに属し、操作手続きが定義されており、オブジェクトの性質が正確に属性に反映されている限り、オブジェクトは外部に存在するかシステムが実際の動作を制御していてもよい。

これらオブジェクトの集合であるクラスに対して、オブジェクトの属性を項目とした表を作れる。表の各タプルはそれぞれオブジェクトに対応する。またオブジェクトを操作する手続きは、このクラスに対応する表と一緒に管理する。

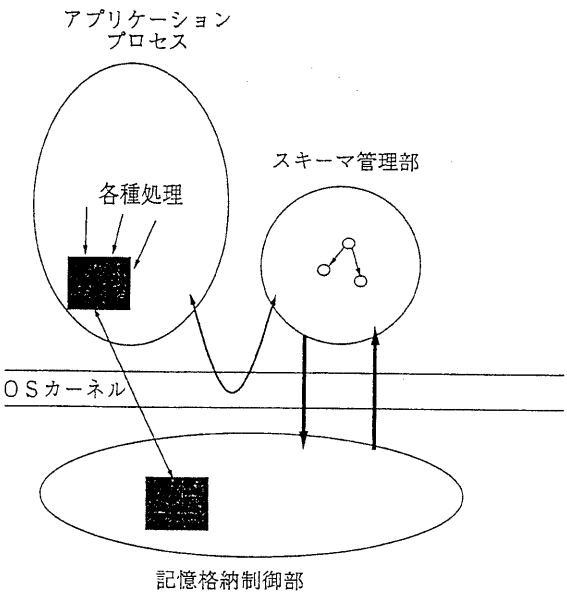

図 2: システム構成

Zodiac-1 のデータモデル上で提供される操作は三種類ある。一つはオブジェクトのクラスに対応する表への操作である。この操作は関係データベースの操作系と同じものとなる。もう一つは表の各タプルから実際のオブジェクトへの対応である。個々のタプルが示す実際のオブジェクトにアクセスするためのオブジェクト識別子をとってくる。最後はこの識別子と表と共に管理されている操作手続き群を使っての実際のオブジェクトに対する操作である。

5 システム構成

Zodiac-1 の中心はオブジェクト格納制御部とスキーマ管理部より構成される。各部は以下の機能を持つ。

5.1 オブジェクト格納制御部

オブジェクト格納制御部はオブジェクトそのものを格納する。オブジェクトは計算機システムの中で Zodiac-1 の管理の対象となるものである。オブジェクトの中には実体一部または全部が Zodiac-1 の内部には存在しないものがある。例えばネットワークを管理する場合の他の計算機システムや他の計算機上に有るリソースなどである。このようなオブジェクトはそのオブジェクトを管理およびアクセスをするための情報や手続きの一部もしくは全部をオブジェクトの代わりに格納する。その他の Zodiac-1 の中に格納されるオブジェクトは全てオブジェクト格納制御部の中に格納する。これは Zodiac-1 が自分自身を管理する場合も例外ではなく、Zodiac-1 管理用のメタデータの全てもオブジェクト格納制御部に格納される。

このオブジェクト格納制御部において、オブジェクトはそれぞれ一つのまとまりとしてのみ管理される。オブジェクトは内部構造などの情報の性質によらず、オブジェクト格納制御部ではオブジェクトがそれぞれ識別されるのみである。オブジェクト内部の構造やオブジェクトの性質を考慮した操作はスキーマ管理部またはアプリケーションシステムなどの外部システムが行なう。この意味ではオブジェクト格納制御部に格納されているオブジェクトは、ファイルシステムのファイルに相当するものである。通常のファイルシステムではファイルアクセスのためにディレクトリなどの構造を持つが、オブジェクト格納制御部はアクセスのためのスキーマは持たず、オブジェクトは識別されるだけでお互いの関係もフラットである。しかしオブジェクトの性質や相互の依存関係などを全く無視してしまうとオブジェクトのアクセスの効率を損なってしまう。そこで将来はオブジェクトの配置戦略などの情報を与え、これらの情報を考慮して制御できるようにする予定である。

オブジェクト格納制御部の格納以外の主要な機能としては、オブジェクトアクセス制御を行なう。オブジェクトを識別し、オブジェクトの内容を読み、オブジェクトの中身の更新を行なう。オブジェクトのアクセスの際に生じる並行制御問題に対処する。具体的にはオブジェクトを管理するプロセスをオブジェクトと共に登録する。オブジェクトは読み書きの各操作に対して、無条件に操作を許す、操作は許すが管理プロセスに通知する、管理プロセスに問い合わせて操作を行なうかどう

か決める、無条件に拒否する、の4つの選択がある。管理プロセスはオブジェクトがこの内のどれを選択するか、事前にオブジェクトにセットしておく。

実際にはこのオブジェクト格納制御部は、OSから見ると外部二時記憶制御の一部、つまり外部ページャとして働く。アプリケーションプロセスは必要とするオブジェクトを自分のプロセス空間にマップすることができ、場合によっては直接オブジェクトに対し操作を行なうこともできる。この時、メモリ空間上にマップされたオブジェクトのバックストレージとしてオブジェクト格納制御部が使われる。

現在はオブジェクト格納制御部はB+treeを中心として構成される。B+treeを用いてオブジェクトの識別管理およびオブジェクト内部のページの管理を行なっている。将来的は抽象化したインターフェースを考え、様々な記憶管理方式が使用できるようにするために、現在検討中である。

5.2 スキーマ管理部

スキーマ管理部はオブジェクト格納制御部に格納されているオブジェクトまたはオブジェクトについての情報をデータモデルに従って実際に管理する部分である。管理のために使用されるデータモデルは4節で述べたデータモデルである。管理の対象となるのはオブジェクトそのものに相当するデータのまとまり、オブジェクトの性質を記述する属性値、オブジェクトが属するクラスに相当するテーブル、およびオブジェクトのタイプに応じたオブジェクトへの操作の集合である。スキーマ管理部はこのデータモデルに基づいたインターフェースを提供する。

Zodiac-1のデータモデルは完結したものではない。Zodiac-1のスキーマ管理部が提供する機能は、オブジェクトの管理において最低限必要と考えられるものである。従って実際に管理を行なう場合は管理する対象や管理の目的に応じて適切なデータモデルを設計し、その管理対象に特有の性質に応じた特殊機能を作り、Zodiac-1が提供する基本機能と合わせて各オブジェクトの管理に当たる。

機能の拡張には二通りの方法がある。

一つはスキーマ管理部の提供するインターフェースを用いる方法である。スキーマ管理部は4節で述べたデータモデルをベースとしたインターフェースを提供している。このモデルは一面、関係データベースに非常に近い形をとっている。スキーマ管理部が提供するインターフェースを用いてテーブルおよび属性値に関して関係データベースの機能相当のものを実現することができる。またZodiac-1では、テーブル相互の関係については規定されていない。テーブルは同一タイプのオブジェクトの集合であるから、それらの間には何らかの関係が定義できることがある。このテーブル間の関係の定義機能もインタフェースの一部であるのでこれを用いてモデルの拡張を行なうことが可能である。

もう一つの方法はオブジェクト格納制御部の外部ページャとしての機能を利用して、スキーマ管理部の外部にオブジェクトの内部構造を司るシステムを別途構築する方法である。Zodiac-1では個々のオブジェクトについて管理している点はオブジェクトがどのクラス（テーブル）に属しているか、オブジェクトの性質を示す属性値、およびテーブル中のタプルを特定した時のオブジェクトへのアクセスなどの操作、の3つである。外部システムの構築によってモデルを拡張するには以下の制約を守らなければならない。

- オブジェクトの生成、消滅およびそのオブジェクトがどのクラスに所属しているかについてZodiac-1の管理と整合性をとらなければならない。Zodiac-1の管理下に入るオブジェクトは生成消滅に対しZodiac-1に要求を出す。この要求に答えてスキーマ管理部はオブジェクトの領域の制御、登録や抹消、初期化処理や消滅後の後処理などを行なう。

- あるクラスが新たに作られた場合、そのクラスのオブジェクトを操作するための手続き群を定義する必要がある。同じクラスに属するオブジェクトは操作のインターフェースは同じであるが、個々の手

続きの実現は異なることがある。このためオブジェクトのクラスを新たに登録する時は操作手続きのインターフェースとデフォルト手続きを登録し、個々のオブジェクトで具体的実現が違う場合にはオブジェクトの登録の時にそれぞれの手続きを登録する。

- オブジェクトの性質はオブジェクトに相当するタプルの属性値に反映していなければならない。このためオブジェクトなどの状態が変化した場合、該当する属性値を正しい値に変化させる必要がある。この操作は基本的には外部システムの作成者が行なう必要がある。ただしZodiac-1ではオブジェクトの登録の時に指定すれば、オブジェクトの内部変化に対し手続きを起動する機能がある。この機能を利用してオブジェクトの内部変化に対してZodiac-1と外部システムとの整合とることは可能である。

その他のデータベース管理システムの機能として重要な機能としてデータの並行制御、トランザクション制御があげられる。現段階でZodiac-1ではオブジェクト、テーブル、タプルに対しての2 phase lockをサポートしているのみである。並行制御やトランザクション制御に対しては、デッドロックの問題、処理効率の問題などから様々な方式が提案されている。多くの場合、データ相互の構造や処理の形式などシステム管理情報を活用して効率良く実行する場合が多い。そこでZodiac-1では並行制御やトランザクション制御のきめ細かい制御についてはZodiac-1のモデル上に専用処理システムとして構築してもらうこととし、本体ではlockなどの基本機能のみのサポートとなる。

6 現状と将来計画

現在、Zodiac-1は基本部分をCMUで開発された分散OS Mach上で作成中である。Zodiac-1に対するインターフェース言語としてはC、C++、Prologを使えるようにする予定である。またこれら三つの言語が同時に相互の機能を生かして使えるよう検討している。またZodiac-1の有効性を検証するために実際のオブジェクトを管理するアプリケーションシステムの作成を検討している。

謝辞

本研究の機会を与えて下さった 弓場 敏嗣 情報アーキテクチャ部長に感謝いします。また日頃から有益な御議論をいただく岡田 義邦 室長をはじめとする情報ベース研究室の皆様ならびに塚本 亨治氏に感謝いたします。

参考文献

(1) P. Mockapetris: Domain Names - Concept and Facilities, RFC1034,Nov. 1987

(2) K. McCloghrie, M. Rose: Management Information Base for Network Management of TCP/IP-Based internets,RFC1066,May. 1990

(3) C. J. Date: An Introduction to Database Systems 2nd Edition,Addison-Wesley,1977

(4) 田沼, 小島, 佐藤, 植村: LAN上のマルチメディアデータベースのための分散記憶システム, 情処研究会報告 DBS-66-4,Jul. 1988

(5) 田沼, 岡田: 第43回情処全体 4M-12

(6) R. V. Baron, et al, MACH Kernel Interface Manual, Aug. 1990