データベース・システム 57 −11
（１９８７．１．１９）

機能分散型データベースサーバシステムについて

滝沢　誠、鈴木隆益、舘 祐治、富岡 誠
東京電機大学理工学部

本論文では、分散型データベースシステムの二つの基本機能、基本的なデータ操作機能と利用者インタフェースを各々データベースサーバとワークステーションに分散したシステムについて述べる。ローカルエリア網(LAN)に結合された種々のデータベースサーバに対して、これらを統合的に利用させるクライアントをワークステーションに実現する問題について論じる。データベースサーバの種類を、サーバが提供するサービスのレベルの差とし、利用者に共通の論理型言語Prologを提供する。また、複数のデータベースサーバとワークステーションから構成される分散システムでは、データベースサーバから導出されたデータがワークステーションを経由して他の利用権のないワークステーションに流出してしまう情報フローを制御する必要がある。本論文では、情報フローモデルに基づいた安全性制御方法を示す。

Distributed Database Server System

Makoto TAKIZAWA, Takamasu SUZUKI, Hiroharu TACHI, and Makoto TOMIOKA
Tokyo Denki University, Faculty of Science and Engineering,
Ishizaka, Hatoyama, Saitama 350-03, Japan.

Information systems are composed of various database servers and workstations which are connected y communication networks. This paper presents how to provide users with unified access to various database servers distributed on local area networks (LANs). In our approach, common Prolog intreface to these various database servers are implemented in workstations. Users can access various database servers through their workstations without being conscious of heterogeneity and distribution of the database servers. Furthermore, incorrect information flow among workstations have to be controlled in the distributed systems. In our approach, by assigning security class to database servers and workstations, information flow are controlled.

1．はじめに

現在の情報システムは、通信網で相互結合された種々のサーバとワークステーションから構成されてきている。サーバとは、利用者に対して、ある計算サービスを提供する計算機システムである。超高速計算機、データベースサーバ、高品質印刷サーバ等は、サーバの例である。利用者から見たとき、サーバは、自分自身のローカルなデータ構造と、これに対する操作演算とから構成されるオブジェクトと考えることが出来る。オブジェクトは、メッセージを受信することにより起動される受動的な情報システムの構成要素である。メッセージは、サーバに行わせたい操作演算を運び、サーバは、メッセージを受信すると、メッセージが運んできた操作演算を実行する。　こうしたサーバの中で、データベースシステムのサービスを提供するデータベースサーバは、最も重要なサーバであり、利用者に複数の種々のデータベースサーバを容易に利用させることが求められている。

こうしたサーバ群に対して、利用者は、通信網に接続された高機能、高性能なワークステーションを持ち、このワークステーションを通して、　種々のサーバを利用する。ワークテーションは、データベースサーバに対して、能動的であり、サーバに対する操作演算をメッセージとしてサーバに送信出来る。

複数のデータベースサーバとワークステーションから構成される情報システムにおいて、利用者に情報システム内のサーバを容易に利用させるためには、次の目標が達成される必要がある。

1) データベースサーバの種類を利用者に意識させず、
2) 通信網上にどのようなデータベースサーバが分散されているかを利用者に意識させず、
3) 利用者の必要とする多様なデータベース応用を容易に構築出来る。

こうした目標を達成する一つの方法は、各データベースサーバが標準的な利用者インタフェースを提供することである。しかし、各データベースサーバが標準的なインタフェースを提供するには、多様なインタフェースを容易しなければならず、そのためのコストと運用オーバヘッドが問題となる。他の方法は、利用者のワークステーションに種々のデータベースサーバに対するインタフェースを設けることである。最近のワークステーションは、数MIPSの演算速度を持ち、インタフェースとしての充分な性能を有している。従って、本論文では、ワークステーションに種々のデータベースサーバに対する種々のインタフェースを持たせる問題について考える。

複数のデータベースサーバとワークステーションから構成される情報システムでは、安全性の問題が、従来の集中型のシステムより重要になってくる。特に、ワークステーションを使用してデータベースサーバをアクセスするときの安全性として、　ワークステーション間での情報の流れ(フロー)を制御する必要がある。　即ち、データベースサーバからワークステーションに導出されたデータを、他のワークステーションが通信網を経由して利用してしまうことにより、利用権のないワークステーションがデータベースサーバ内のデータを利用出来てしまう問題である。この問題を解決するために、本論文では、情報流モデル[DENN]に基づいた方式を示す。

まず、第２章では、データベースサーバの異種性とは何かについて考える。第３章では、情報システムのアーキテクチャを示す。第４章では、ワークステーション内に実現されるクライアントについて述べる。第５章では、情報流の制御について述べる。最後に、第６章で、我々のシステムの実現について述べる。

2．データベースサーバの異種性と応用の多様性

データベースサーバの異種性とは何かを考える。まず、データベースサーバとは、あるデータ構造とこれに対するデータ操作演算を利用者に提供するシステムとする。　現在、データベースサーバといわれるものには、ローカルエリア網(LAN)のバーチャルサーバ、 CORBUSサーバといったファイルサーバ、 SunのNFS[NFS]、汎用大型計算機の関係型、 網型データベースシステム、更にJICST等の商用情報サービス会社の検索システム等がある。これらのデータベースサーバの提供するサービスには、いくつかのレベルがあり、この差をデータベースサーバの異種性と考えることにする。データベースサーバの提供するサービスには、図１に示すレベルがある。

非手続き的操作レベル	SQL
巡航的操作レベル	網型モデル
ファイル操作レベル	物理ファイル
仮想空間操作レベル	ページ

図１．データベースサーバのサービスレベル

仮想空間操作レベルでは、　ページ単位の入出力が行われる。仮想空間は、ページの連続領域であり、各ページはページ番号により識別される。このレベルでは、指定したページ番号を持ったページがアクセスされる。例としては、Apollo DomainのAGEISオペレーティングシステム[APOL]では、通信網で接続されたワークステーション全体に対して一つの仮想空間がある。

ファイル操作レベルでは、物理レコード単位の入出力が行われる。同一形式の物理レコードの集合を物理ファイルという。各物理レコードは物理レコード識別子により識別される。このレベルでは、指定した物理レコード識別子を持った物理レコードがアクセスされる。各物理ファイル内の物理レコードを、仮想空間内のページにどのように記憶するかを格納配置という。格納配置方法としては、近接配置、ランダム、ハッシングがある。

巡航的操作レベルでのアクセス単位は、論理レコードである。論理レコードの集合を論理ファイルという。論理ファイル内のレコードは次の二つの方法によりアクセス出来る。

(1) 順次アクセス
(2) 直接アクセス

直接アクセスでは、識別子ベースのアクセスと、値ベースのアクセスがある。値ベースアクセスでは、論理ファイルからある値を持つ論理レコードがアクセスされる。識別子ベースアクセスでは、ある識別子を持った論理レコードがアクセスされる。

順次アクセスでは、論理ファイル内の論理レコードがある順序で順々にアクセスされる。順序としては、識別子の順序、論理レコードの値の順序がある。値による順序の実現方法としては、ポインタ等がある。

巡航的アクセスレベルは、従来の網型モデル[CODA]に対応している。

最上位の非手続的操作レベルでは、関係型のSQLといった非手続き的データ操作言語により、データの巡航的なアクセス方法とは独立な論理的なデータ構造が操作される。

以上のデータベースサーバは、基本的なデータ操作演算を提供している。これに対して、データベースの応用は、データ操作演算を組あげたプログラムとして実現されている。利用者に対して、容易に自分の必要とする応用を作れることが必要になる。論理型言語Prolog[KOWA]は、データと手続きを均一に扱える利点を持った言語である。このために、我々は、利用者にPrologを提供し、これにより応用を開発させることを試みる。更に、プロダクションシステム、フレームシステム、意味網を用いた種々の専門家システムもデータベース応用の開発に用いることも試みる。

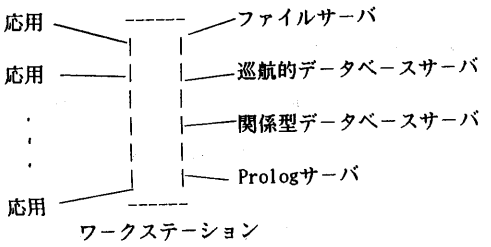


図2．応用とデータベースサーバ

3．システムアーキテクチャ

データベースサーバには、今示した種々のサービスレベルを提供するものがある。本論文では、これらの種々のレベルのサービスを、利用者が統合的に利用する方法について述べる。図3に情報システムの構成を示す。情報システムは、データベースサーバDBSi(i=1,...,m)とワークステーションWSj(j=1,...,k)が通信網CNにより結合された形態をとっている。

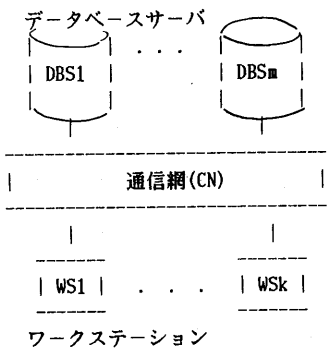


図3．情報システムの構成

3．1 データベースサーバ

データベースサーバとしては、種々のサービスレベルを提供するものがある。データベースサーバとしては、提供するサービスのレベルの差により、次の種類がある。

1) ファイルサーバ
2) 巡航的データベースサーバ
3) 関係型データベースサーバ
4) Prologサーバ

A．ファイルサーバ

ファイルサーバとは、物理ファイルレベルのサービスを提供するサーバである。ここでは、ファイルは、ストリーム形式であるとする。即ち、Unix又はMS/DOSのファイルのインタフェースを提供しているとする。ファイルサーバは、クライアントに対して、次の操作演算を提供している。

1) create　ファイルの生成。
2) open　ファイルのオープン。
3) close　ファイルのクローズ。
4) read　ファイルから指定した長さのストリームを指定したエリアに読み出す。
5) write　ファイルに指定した長さのストリームを指定したエリアから書き出す。
6) lseek　ファイルのポインタを指定したオフセットの位置に移動する。
7) remove　ファイルの消去を行う。

ファイルサーバの例としては、SunのNFS(Network File System)、Unix system VのRFS(Remote File System)がある。更に、索引、ハッシング等のアクセス方法を有したもの(ミナトバーチャルサーバ)もこの例である。

B．巡航的データベースサーバ

巡航的データベースサーバは、網型のデータ構造とこれに対する巡航的操作演算を提供するサーバである。いわゆる従来のCODASYL型[CODA]のデータベースシステムである。一般に、大型の汎用計算機に実現されている。巡航型データベースサーバは、従来のデータ操作言語DML[OLLE]を利用者に提供するものとする。DMLの形式的な意味については、[TAKI83]に述べてある。

C．関係型データベースサーバ

関係型データベースサーバは、関係型のモデル[CODD]、即ち、表形式のデータ構造とこれに対する非手続き的な操作演算を提供するシステムである。いわゆる関係型のデータベースシステムである。このデータベースサーバは、操作言語としてSQL[DATE]を提供しているものとする。

D．Prologサーバ

Prologサーバは、論理型言語Prolog[KOWA]を提供するサーバである。Prologサーバは、具象単位節の集合である事実ベースと規則節の集合である規則ベースから構成される。Prologサーバは、目標節による問合せに対して、SLD導出[KOWA,LLOY]により、答を求めるサーバである。

以上の四種のデータベースサーバが通信網により結合されているものとする。この他には、JICST、DIALOG等の情報検索システム、プロダクションシステム、フレームシステム等の専門家システムもサーバである。これらについては、今後考えていくことにする。

３．２　ワークステーション

ワークステーションは、 数10MB～100MB以上の二次記憶能力を有した高性能、高機能な計算機である。ワークステーションのOSとしては、UNIXが主に用いられてきている。ワークステーションは、利用者と種々のデータベースサーバ間のクライアントとしての役目を持つ。 クライアントは、サーバと利用者間のインタフェースの役目をもつプロセスである。

利用者は、自分のワークステーションを通信網に接続することにより、通信網上の種々のサーバの提供するサービスを利用する。このとき、各サーバのインタフェースの相違を解消したり、通信網上にどの様なサーバが在るかを知らせるのは、 クライアントの役目である。 本システムでは、種々のデータベースサーバに対して、データベースサーバの種類の差を意識させないで、同一の操作言語Prologにより操作させるクライアントをワークステーションに実現する問題について述べる。

３．３　通信網

通信網は、データベースサーバとワークステーションを結合し、これらの間での通信を提供するシステムである。通信網には、提供される通信サービスの種類により、次の二種がある。

(1) 放送通信網
(2) 一対一通信網

一対一通信網は、一対の通信実体間で、高信頼な通信を提供するシステムである。一対一通信が高信頼であるとは、送信されたメッセージは、 必ず届き、 紛失することもなく、冗長となることもなく、かつ送信した順に届くことである。 現在、OSIモデル[ZIMM]として、国際標準化されてきている通信プロトコルのコネクション指向プロトコルは、高信頼な一対一通信を提供しようとするものである。

また、DARPAのTCP/IP、XeroxのXNS、IBMのSNA等の既存の通信プロトコルも一対一通信網のプロトコルである。

一方、放送通信網は、高信頼な放送通信を提供する通信網である。 放送通信網での高信頼性とは、n(≥2)個の通信実体E1,...,En間で次の性質を満足する通信を提供することである。

1) ある通信実体Eiが送信したメッセージは、すべての通信実体Ej(j=1,...,n)に送信順に、 紛失することも、冗長化することもなく必ず届く。

2) ある通信実体EiとEjが各々メッセージmiとmjを送信したとき、 すべての実体Ek(k=1,...,n)でmiとmjを同じ順序で受信する。即ち、ある通信実態Ekでmi, mjの順に受信したならば、他のすべての通信実体でもmi, mjの順に受信する。

このような通信実体E1,...,Enの集合を群とする。 特に、1)の性質を持つ通信網を、単純放送通信網といい、1)と2)の両者の性質を持つ通信網を、単一路放送通信網という。群は、従来の通信プロトコルにおける二つの通信実体間のコネクションの概念をn(≥2)に拡張したものである。

群に対しては、群の開設、解除、送信、受信の操作がある。群では、この中のすべての実体は、ある実体が送信したメッセージを「同時」に受信出来る。即ち、すべての通信実体では、同じ順序で同じメッセージを受信出来る。放送通信網を従来のパケット交換網で実現しようとすると、n(≥2)個の通信実体にあるメッセージを送信しようとすると、 n本のコネクションを開設し、各々にパケットを送信する必要がある。 即ち、n個のパケットを送信する必要がある。これに対して、Ethernet、トークンリング、トークンバス等のローカルエリア網(LAN)や、 ALOHANET等の無線網では、 データリンク層のMAC(媒体アクセス制御)層で放送通信が提供されている。 従って、これらの網では、n個の通信実体にあるメッセージを送信するには、 MAC層で唯一つのパケットを網に送信すればよい。このことから、高信頼放送通信網の実現は、 LANと無線網を用いると容易になる。 このため、Ethernetのデータリンク(MAC)サービス上に、高信頼放送通信を行うトランスポート層のプロトコル(TCCP)[TAKI84]を実現している[図４]。

TCCPを用いると、分散型データベースシステムの分散問合せ処理、コミットメント制御、同時実行制御のための通信処理プロトコル(DDBP)の実現は容易になる[TAKI83,84]。

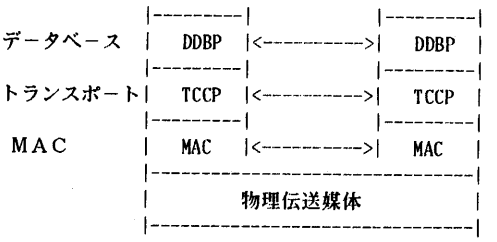


図４．通信プロトコル階層

４．ワークステーション

ワークステーションは、利用者とデータベースサーバ間のインタフェースであるクライアントとしての役目を持つ。ワークステーションは、種々のデータベースサーバに対して共通な論理型言語Prologを提供する。ワークステーションには、Prologと、次の種類のデータベースサーバとの間のインタフェースが実現されている[図５]。

ファイルサーバ
イメージファイルサーバ
巡航型データベースサーバ
関係型データベースサーバ
Prologサーバ

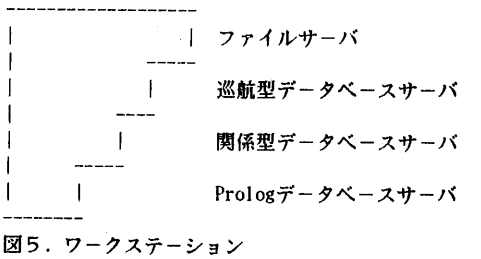


図５．ワークステーション

4．1 Prolog視野

利用者は、ワークステーションを通して、データベースサーバ内のデータを、論理型言語Prologの具象単位節として見れる。 具象単位節は、 P(a1,...,an)の形式をしていて、Pはn項の述語記号、aiは定数である。

A．ファイルサーバ

ファイルサーバ内の各ファイル様式F(f1,...,f)(m>0)に対して、 その各レコードr=<d1,...,dm>は、Prologの具象単位節F(k,d1,..,dm)と見える。 ここで、fiはファイルのフィールドであり、 diはその値である(i=1,...,m)。 また、kはレコードrの識別子である。図6に、四つのレコードからなるファイルEmpと、そのProlog視野を示す。

Emp	name	age
	a	18
	b	25
	a	18
	c	36

```
Emp(k1, a, 18).
Emp(k2, b, 25).
Emp(k3, a, 18).
Emp(k4, c, 36).
```

a) ファイル　　b) 単位節

図6． ファイルとProlog視野

B．巡航型データベースサーバ

巡航型データベースサーバ内のデータ構造は、レコード型R(t1,...,tm)(m≥0)と、 レコード型R1とR2の間の親子型S=(R1, R2)[図7]の二種の要素から構成される。ここで、tiは、 データ項目である。また、親子型Sで、R1とR2をそれぞれ親レコード型、 子レコード型という。Rの各レコード実現値r=<d1,...,dm>は、Prolog具象単位節R(k,d1,...,dm)と見れる。 ここで、kはレコード実現値rのデータベース鍵である。　　レコード型R1のレコード実現値r1=<k1,d1,...>とR2のr2=<k2,e1,...>の親子実現値は、具象単位節S(k1, k2)と見れる。網型のデータ構造とProlog節の論理的な対応関係については、 [TAKI86,87]を参照されたい。

図8に、部と従業員の関係を示す網型のデータ構造とその Prolog視野を示す。

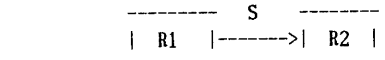


図7． 親子型

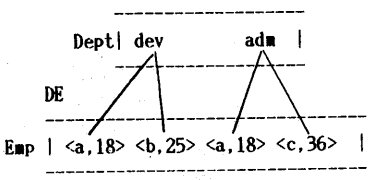
C． 関係型データベースサーバ

関係型データベースサーバでのデータ構造は、 関係R(a1,. ..,am)(m≥1)から構成される。 関係R内の各組t=<d1,....,dm>は、具象単位節R(d1,...,dm)と見れる。

図9に関係Empとそのprolog視野を示す。

D．視野

以上のようにして、情報システム内のデータベースサーバ内のデータ構造を、 ワークステーションの利用者は、Prologの具象単位節と見れる。利用者は、こうした具象単位節の上に、規則節により、視野を定義出来る。従来の関係型言語SQ Lと異なり、Prologでは、視野を再帰的に定義



a) 網型データ構造

```
Dept(d1,dev). Dept(d2,adm).
Emp(e1,a,18). Emp(e2,b,25). Emp(e3,a,18).
Emp(e4,c,36).
DE(d1,e1). DE(d1,e2). DE(d2,e3). DE(d2,e4).
```

b) Prolog視野

図8． 網型データ構造とProlog視野

Emp	no	name	age
	1	a	18
	2	b	25
	3	a	18
	4	c	36

```
Emp(1, a, 18).
Emp(2, b, 25).
Emp(3, a, 18).
Emp(4, c, 36).
```

a) 関係　　b) 単位節

図9． 関係とProlog視野

出来る利点がある。 例えば、図8に対して、開発部(dev)の部員を示す視野Dempは、次のように定義出来る。

```
Demp(X) :- Dept(Y, dev), DE(Y,Z), Emp(Z, X, _).
```

4．2 Prologインタフェースの実現

ワークステーション内のクライアントは、利用者からのProlog目標節を受け取り、各データベースサーバに対するデータ操作演算の手続きを合成する。 合成手続きの詳細は、[TAKI86,87]を参照されたい。

トランザクションを記述するために、次のような述語が用意されている。

1) begin()　トランザクションの開始を指示する。
2) write(ID,VAL) IDの指示するオブジェクトに、 値VALを書き込む。
3) commit() トランザクションの正常終了を指示する。
4) abort() トランザクションの異常終了を指示する。

begin()が、 導出で選択されたとき、これ以降で導出に失敗したならば、 begin()の前に後戻りする。commit()が選択されたならば、 begin()からcommit()の間には後戻り出来ない。 各節でbegin()より左の基本式は、書かれた順に選択されるものとする。

例えば、 ある人AからBに、X円を送金するトランザクションは次のように書ける。

```
trans(A, B, X) :- begin(), account(K, A, AC),
                          account(H, B, BC),
                          acccheck(AC, X),
                  write(K, account(K, A, AC-X)),
                  write(H, account(H, B, BC-X)),
                    commit().
acccheck(X, Y) :- X <Y, abort().
acccheck(X, Y) :- X >= Y.
```

5. 安全性制御

データが分散し、端末としてのワークステーションが計算と記憶機能を有した環境下では、データベースの安全性を保つことが、従来の集中システム以上に困難になる。こうした分散型システムにおいて特に重要となる安全性は、不正な情報流を防ぐことである。

5．1 安全性

まず、データベースに対する利用権を定義してみよう。ここで、oをオブジェクト、sをサブジェクト、tを演算の種類とする。一般に、オブジェクトはデータであり、サブジェクトはワークステーションの利用者である。 tは、オブジェクトに対して適用可能な演算の種類である。利用規則は、三つ組<s, o, t>で与えられ､情報システム全体での利用規則の集合を利用規則集合Rとする。 利用規則を書くことを利用権付与(authorization)とい、 書く人を利用権付与者という。利用規則が正しく保たれていることを保障するための制御をアクセス制御という。 各オブジェクトoに対する集合{<s,t> | <s, o, t> ∈ R}をアクセスリストAL(o)とする。一方、各サブジェクトsに対する集合{<o,t> | <s, o, t> ∈ R}をケーパビリティリストCL(s)とする。

アクセス制御によって制御出来ない問題に、不正な情報流を防ぐ問題がある。例として、図１０を考える。サブジェクトaとbがあり、オブジェクトfとgがあり、aはfをreadとwriteでき、 bはfを利用できないとし、 aとbは共にgをreadとwriteできるとする。ここで、aはfからデータxを読み、 gに書いたとする。bはgを読めるので、g内のデータxを読める。 この結果bはf内のデータxを読んだのと同じことになり、利用規則が破られてしまう。このとき、aからbに情報が流出したという。

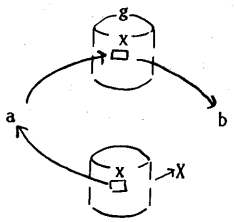


図１０． 情報流

情報流制御は、不正に情報が流出することを防ぐための制御である。ワークステーションから構成されるシステムでは、データベースサーバから導出されたデータを自分のワークステーション内に記憶出来る。ワークステーション内のデータが、他のワークステーションに導出されることにより、不正な情報の流出が起こり得る。従って、ワークステーションを主要な要素として構成される分散システムでは、情報流の制御が安全性を保つために重要となる。

5．2 情報流制御

情報流を制御するために、オブジェクトの概念に基づいた情報流制御方式を用いる。

情報システム内のデータの基本単位は、オブジェクトである。各オブジェクトoは、識別子(oid)と属性値を持つ。識別子は、情報システム全体で一意である。また、各オブジェクトoは、ある安全性クラスCに属する。オブジェクトoの安全性クラスをclass(o)と書く。 同一の属性構成と安全性クラスをもつオブジェクトの集合をオブジェクトクラスまたはクラスとする。 各ワークステーションwも、ある安全性クラスclass(w)に属する。情報システム全体での安全性クラスの集合をCCとすると、CCは半順序集合(CC, <)であり、束となる。ここで、＊と＋を各々、最大下限、最小上界を与える二項演算とする。 CCの任意の要素cとc'に対して、 c<c'であるとき、 c->c'と書く。c->c'であるとき、クラスc内のオブジェクトoの値をクラスc'に流せることを示している。例えば、oをオブジェクト、wをワークステーションとして、 class(o)->class (w)としよう。このとき、 oをwに導出出来る。 そうでなければ、導出出来ない。

o1,...,onをオブジェクトとし、opをo1,...,onに対する演算とする。 演算結果をop(o1,...,on)とし、この演算結果を記憶するオブジェクトクラスの安全性クラスをcとする。 このとき、 c<-class(o1)+...+class(on)ならば、o1,....,onに演算opを適用出来る。opの例としては、関係RとSに対する結合演算がある。演算結果は、新しいオブジェクトとなり、 安全性クラスはcとなる。 opの例は、o1,...,onからの検索がある。

更に、通信網に送信されるデータは、公開鍵方式により暗号化することにより、 通信の秘密性と確証性を保障する。

6．実現

図１１に、現在実現中のシステムの構成を示す。ファイルサーバ(ミナトバーチャルサーバ)と、ワークステーションとしてのFM16Bを二台とPC-9800を、 ローカルエリア網(LAN)OMNINETで結合した形態をしている。 バーチャルサーバは、ストリーム形式のファイルに加えて、索引機能を提供している。

ワークステーションとしては、 パソコンのFM16BとPC-9800を用いている。パソコンのOSは、MS/DOSである。パソコン内に、Prologインタフェースの実現を、Ｃ言語により行っている。 また、FM16Bの専門家システム開発ツールであるEshellインタフェースの実現も行っている。

現在、 光ディスクの組み込みと、Sun3/110Cの網ファイルシステムであるNFSの結合も検討している。

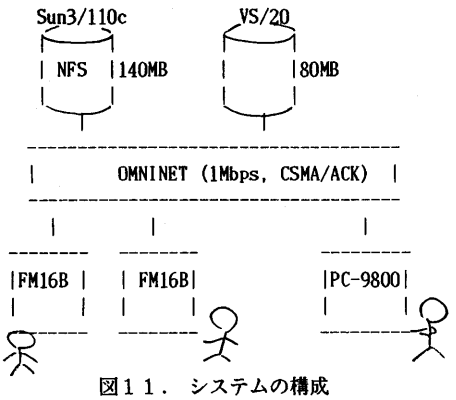


図１１．システムの構成

７．おわりに

今後の情報システムは、通信網に種々のサーバが結合された形態をとり、利用者は、これらのサーバのなかから必要なサービスを選択し利用していくことになる。このとき、ワークステーションの役目は重要である。多様化するサーバに対して、ワークステーションは、利用者に対して、サーバの相違とそれ等の分散と独立なインタフェースを提供する必要がある。本論文では、共通のインタフェースとして、論理型言語Prologを、種々のデータベースサーバに対して提供するワークステーションについて述べた。

現在、データベースサーバに加えて、プロダクションシステム、フレームシステム等の専門家システムの組み込みをめざしている。

謝辞

OMNINETとファイルサーバについて御協力を頂いているミナトエレクトロニクス社と、ワークステーション及び専門家システムEshellについて御協力を頂いているファコムハイタックのシステム第５部第２課の山本課長に感謝する。

参考文献

[APOL] Apollo, "AGEIS Operating System"

[CODD] Codd, E. F., "A Relational Model of Data for Large Shared Data Bank," CACM, Vol.13, No.6, pp.337-387.

[CODA] CODASYL DDL Committee, "CODASYL Data Description Language," Journal of Development, 1973 and 1978.

[DATE] Date, C.J., "Introduction to Database Systems," Addison-Wesley, 1981.

[DENN] Denning, "Informartion Flow Model"

[KOWA] Kowalski, R., "Logic for Problem Solving," Research Studies Press, 1979.

[LLOY] Lloyd, D., "Foundation of Logic Programming," Springer-Verlag, 1985.

[MINA] ミナトエレクトロニクス, "バーチャルサーバ説明書"。

[NFS] Sun Micro., "Network File System"

[OLLE] Olle, T., "The CODASYL Approach to Data Base Management," John Wiley & Sons, 1978.

[TAKI83] Takizawa, M., "Distributed Database System - JDDBS," JARECT, Ohm-sha and North-Holland, Vol.7, pp.262-283, 1983.

[TAKI84] 滝沢他、"LANを用いた分散型データベースシステムLISの通信処理手順について," 情報処理学会「LAN/マルチメディアの応用と分散処理」シンポジウム、1984.

[TAKI86] Takizawa,M., Itoh, H. and Moriya, K., "Logic Interface System on CODASYL Database System," Proc. of the Logic Programming Conference, pp.111-117, 1986.

[TAKI87] 滝沢、"演繹的網型データベースシステム," 人工知能学会誌、No.3、1987。

[ZIMM] Zimmerman, "OSI Referencial Model"