

## 異種分散型データベースシステムJDBBSについて

(財)日本情報処理開発協会 滝沢 誠 (Makoto Takizawa)

(財)日本情報処理開発協会 横塚 実 (Minoru Yokotsuka)

### 1. 序

オフィス情報システム等の今後の情報システムでは、個人計算機上に構築されてきた個人データベース、従来の大型データベースが種々の通信網で結合された形態の中で、データベースの統合利用技術、即ち分散型データベースシステム(DBBS)技術が中核になると考えられる。DBBSとは、意味的に関連した複数のデータベースシステム(DBS)を、通信網を用いて結合し、利用者に対して仮想的な1つのDBSサービスを提供するシステムである。ここでDBSとは、あるデータベースモデルを利用者に提供する自動機械とする。データベースとは、データベース構造と、この上の操作演算とを意味するものとする。DBSの異種性とは、各データベース(例、関係モデル[CODDE70], CODASTLモデル[CODAS73])の相違とする。異種のDBSから成るDBBSを、異種DBBSとする。異種DBBSを実現する為には、各DBSの異種

性と、各DBSが通信網上に分散している事によって生じる分散性との2つの問題の解決が必要になる。

本書では、当協会が1977年から82年3月まで開発を行った異種のDBS JDDBS(Jiploc DBS)の設計と実現結果について述べる。第2章では JDDBSの全体アキテクトチャ、第3章では異種問題の解決法を、第4章では分散問題の解決について述べる。

## 2. JDDBSの全体アキテクトチャ

異種性と分散性との2つの問題を解決する為のJDDBSの階層構造とシステム構成について述べる。

### 2.1 四階層構造

FDDBSとは、種々のDBSを通信網によって結合し、利用者にあたかも1つのDBSかの様にみせるシステムである。よって、FDDBSを実現する為には、各DBSの異種性と分散性との2つの問題を解決する必要がある。この2つの問題を独立して解決する為には、JDDBSは、図2.1に示す4つの階層から成る。<sup>(GAKIN78,79)</sup>  
 ローカル内部(LI)層は、既存DBSがFDDBSを通して共有可能なデータ構造を表すLISキーマ(LIS)と操作言語(LIML)とを提供する。LI層は、各DBS固有のデータモデルに基づいている。  
 ローカル概念(LC)層は、FDDBS全体の共通モデルによって、

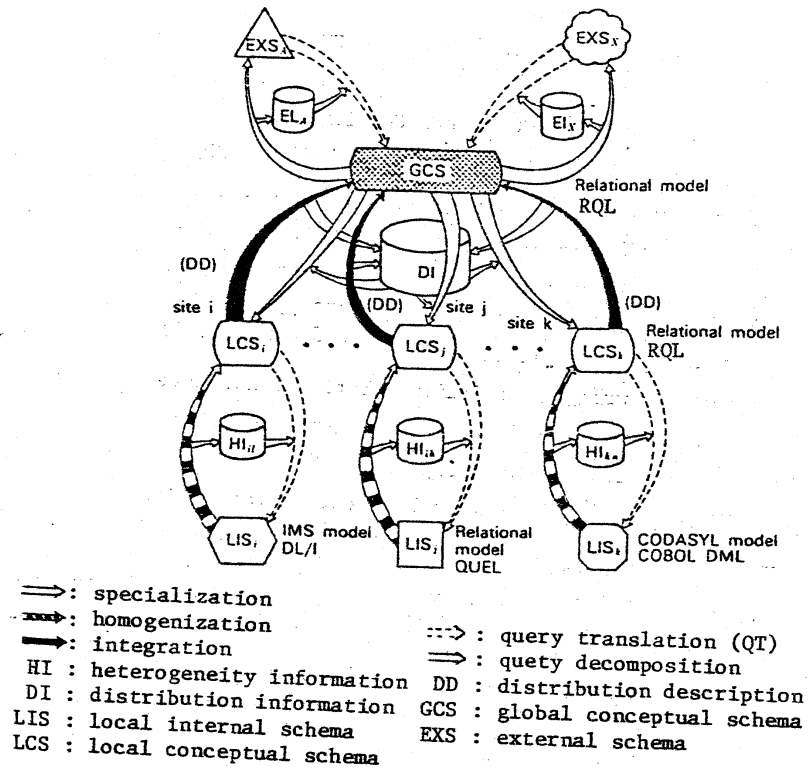


図 2.1 四階層構造

LI層を表している。LCスキーマ(LCS)は、LISの共通モデルの視野である。共通モデルとしては、簡単なデータ構造と非手続的操作言語を有する関係型[CODASYL]モデルを用いる。LC層では、利用者は各DBSの異種性を意識することなく、共通モデルによ、データ操作を行える。共通操作言語としては、QUEL[HELDG]と類似した言語RQLを用いる。

全体概念(GC)層は、各DBSのLC層上の、DBS全体と一意な視野を表している。GC層は LC層と同一の関係型データ構造とRQLとを提供している。GC層では、利用者は、各DBSの存在を意識することなく、DBSをあたかも1つのDBSかの様

に見る事が出来る。外部(EX)層は、EDBSの名応用固有のデータ構造としてのEXスキーマ(EXS)とEX操作言語(EXML)とから成る。EX層は各応用に適合したデータモデルに基づいている。EX層は、GC層でEDBSが1つのDBSに仮想化されているので、従来のDBSの外部スキーマ又は副スキーマ層と同じに考えれる。

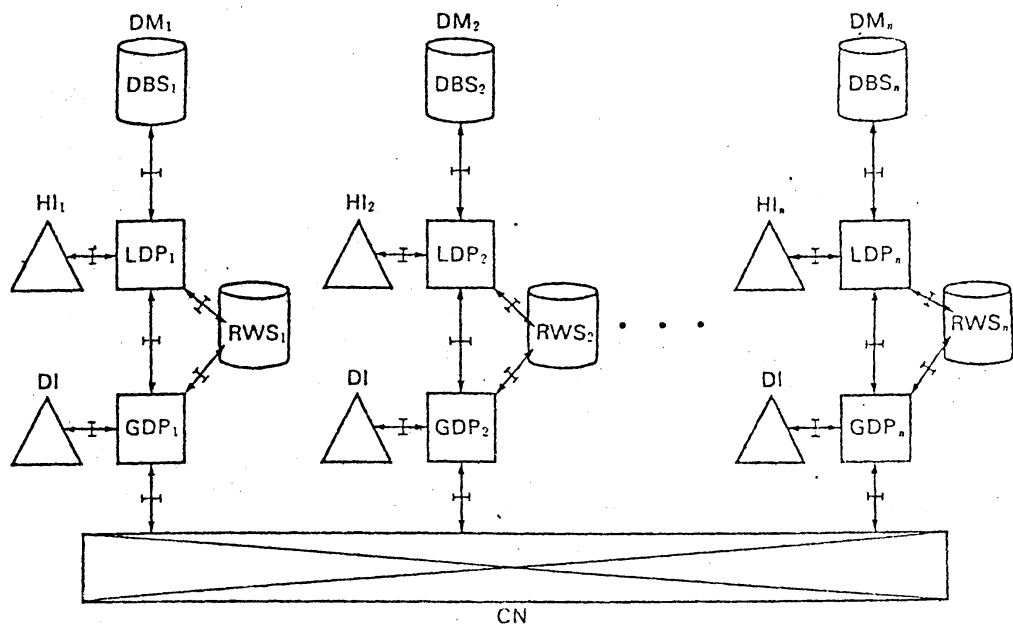
EDBSを以上の四層に階層化する事により、EDBSの異種性・分散性問題を独立して解決出来る。構成的DBSのDBMSの変更、DBSの追加・変更に対しても各々LC-LI, GC-LC間の層間写像により、独立して解決出来る。

## 2.2 EDBSのシステム構成

四階層構造に基づいたJEDBSのシステム構成を図2.2に示す。JEDBSでは、 $m(m \geq 1)$ 個のデータモジュール $DM_1, \dots, DM_m$ が、通信網(CN)で結合されている。CNは、DM間の高信頼な種類の通信(1対1, 1対多, 多対多)を備えている網である。JEDBSは、ローカル網(LAN)が有する1対多通信機能をCNの基本機能としている。

各 $DM_i$ は、全体データベースプロセッサ(GDP<sub>i</sub>), ローカルデータベースプロセッサ(LDP<sub>i</sub>), データベースシステム(DBS<sub>i</sub>), 異種性情報(HI<sub>i</sub>), 分散性情報(DI<sub>i</sub>), 関係作業域(RDS<sub>i</sub>)との6つのモジュールから成る。DBS<sub>i</sub>は既存のデータベースシステムであり、LISに対するLIMLトランザクションを実行出来る。LDP<sub>i</sub>

は、DBS<sub>i</sub> 上にあつて LC 層を提供するインターフェースである。  
 LDP<sub>i</sub> は、LIS から LCS を生成するスキーマ変換と、LC 層の ROL 変換と LI 層の LIMU に変換し、DBS<sub>i</sub> 上で実行し、解を関係として RWS<sub>i</sub> に格納する問合せ変換との二つの機能を備えている。  
 HI<sub>i</sub> はスキーマ変換を生成させ、問合せ変換を用いた、LCS 及び LCS と LIS の対応情報から成る DDI である。GDP<sub>i</sub> は、LDP<sub>i</sub> 上にあつて GC 層を提供する。GDP<sub>i</sub> は、全体管理者(GA)による LCS 上の GCS 定義より DI を生成し、他の GDP<sub>i</sub> に配布する。DI は GCS、及び GCS と LCS との対応情報から成る DDI である。  
 GDP<sub>i</sub> は、GC 層の ROL 変換を、LC 層のものに変換し、DM 内/



HI: heterogeneity information      DM: data module  
 DI: distribution information      DBS: database system  
 CN: communication network      LDP: local database processor  
                                          GDP: global database processor  
                                          RWS: relational working storage

図 2.2 JDDBS のシステム構成

間通信処理による、解を得る分散問合せ処理機能三有している。これは、DIに基づいて行われる。GDPは、RDS内の関係に関する演算を作用するとともに、GDP間の送受用の作業域としてRDSを用いる。

### 3. ロ-カルジョ-タペ-スプロセ-サ(LDP)

JDDBSでは、DBSとしてCODASYL型[CODASYL73, OLET78]のDBSをまず異種DBSとして考えている。LDPは、CODASYL DBS上の関係インタフェ-スシステムである[TAKIM80, 82b, 83]。LDPは、図3.1に示す様に、スキ-マ変換を行うHIP(HIプロセ-サ)と、問合せ変換を行うQTPとから成る。LDPは、次の機能を利用者に提供している。

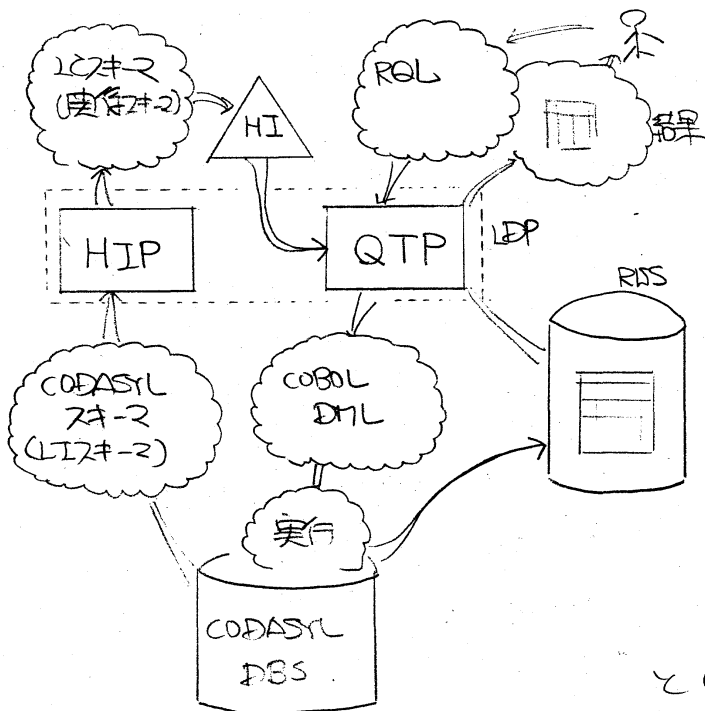


図3.1 LDP

の機能を利用者に提供している。

- (1) CODASYLスキ-マ(LIS)から関係(LC)スキ-マの生成(スキ-マ変換)
- (2) RQL検索、更新をCODOL DMLに変換し実行させ、解を関係

としてRDSに格納する。

- (3) RQLは、QUELとほぼ

同等(即ち制限、射影、結合、集約関係)の能力を保持している。

### 3.1 HIP

HIPは、CODASYLスキーマ(LIS)から関係スキーマ(LCS)を生成し、LCSからLCSとLISの対応情報をHIとして生成する。CODASYLモデルとLCSモデルとの対応は[TAKIM83a]で論じられる。表3.1の様に対応づけられた。図3.2は、従業員、プロジェクト

表3.1 CODASYLデータ構造とLCSデータ構造の対応

CODASYLデータ構造	LCSデータ構造
L型 $R(t_1, \dots, t_m)$	E型関係スキーマ $R(@R, t_1, \dots, t_m)$
セット型 $R(t_1, \dots, t_m)$ $\downarrow S$ $R_2(t_1, \dots, t_{m_2})$	E型 $R(@R, t_1, \dots, t_{m_1})$ $R_2(@R_2, t_1, \dots, t_{m_2})$ B型 $S(@R_1, @R_2)$
リンク型 $R_1(t_1, \dots, t_{m_1}) \dots R_n(t_1, \dots, t_{m_n})$ $\swarrow S_1 \quad \searrow S_n$ $R(t_1, \dots, t_m)$ <small>⇒ リンク型の O型 □: リンク型の E型</small>	E型 $R_1(@R_1, t_1, \dots, t_{m_1})$ $\dots$ $R_n(@R_n, t_1, \dots, t_{m_n})$ G型 $R(@R, \dots, @R_n, t_1, \dots, t_m)$

<sup>\*)</sup> @R は、L型 R の属性を値として取る属性で主属性とする。

以上、所屬、論又を表す CODASYL スキーマ的である。これは

HIPによる、図3.3に示すLCSスキーマとなる。

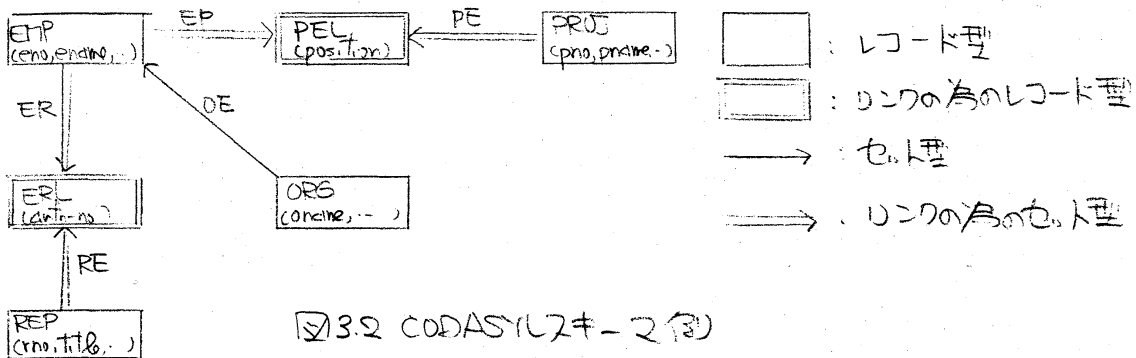


図3.2 CODASYLスキーマ例

- |    |                            |    |                        |
|----|----------------------------|----|------------------------|
| E型 | EMP (@E, eno, ename, ...)  | B型 | OE (@O, @E)            |
|    | REP (@R, rno, title, ...)  | G型 | ERL (@E, @R, auth-no)  |
|    | PROJ (@P, pno, pname, ...) |    | PEL (@P, @E, position) |
|    | ORG (@O, o.name, ...)      |    |                        |

図3.3 図3.2のLCS

### 3.2 QTP

問合せ変換アロセ、サQTPは、LCSスキーマ上のRQL問合せを、HIを用いてCOBOL DMLに変換し、CODASYL DBS上で実行させ、解を関係としてRWSに格納する。QTPはRQL検索と更新DMLに変換出来るが、ここでは検索について論じる。更新については、[TAKI83]を参照せたい。

述語論理形式の関係言語RQLとCOBOL DMLとの相違は、(i) 参照するデータ構造(関係モデルとCODASYLモデル), (ii) アロセ又単位(集合単位とレコード単位)との又点である。この為、我々はCODASYLモデル上の述語論理形式の問合せ言語COL(CODASYL問合せ言語)を設け、RQLをCOLに変換(構造変換)し、次に



SQL からレコード単位のアクセス手順を生成する手法を用いた。

### 3.9.1 RQL

RQL は QUEL [HELDG75] と類似した述語論理形式の言語である。RQL に基づく LCS 上の検索は 次の様に書かれる。

$$\begin{array}{l} \text{range } (r_1, R_1), \dots, (r_n, R_n); \\ \text{retrieve into } T(a_1 = \text{exp}_1, \dots, a_n = \text{exp}_n) \text{ where } \text{qual}; \end{array} \quad \dots (1)$$

$R_i$  は LCS 内の関係であり、 $r_i$  は  $i$  の組変数である。T は目標関係 ( $I(a_1, \dots, a_n)$ ) であり、qual は条件式、 $(a_i = \text{exp}_i, \dots, a_n = \text{exp}_n)$  は目標リストである。関係  $R_i$  の属性  $a$  は、 $r_i.a$  と表され、 $\text{exp}_i$  は定数、属性、又はこれら上の算術式である。

RQL は QUEL に対して、(i) 主属性間の比較演算子として  $=$  と  $\neq$  のみを用いるが、(ii) 主属性上の算術式 (e.g.  $x * X + y * Y$ ) は許されず、(iii) 集約関数としては count と any のみを用いる。

例えば、3.3 の LCS に対する検索「鈴木を著者とする論文の著者名を求めよ」は RQL で次の様に書ける。

$$\begin{array}{l} \text{range } (e, EMP) (e1, EMP) (r, REP) (er, ERL) (er1, ERL); \\ \text{retrieve into } T(r.rno, e1.ename) \text{ where } e.ename = \text{"鈴木"} \text{ and} \\ e.@E = er.@E \text{ and } er.auth-no = 1 \text{ and } er.@R = r.@R \text{ and } r.@R = er1.@R \text{ and} \\ er1.@E = e1.@E \text{ and } er.auth-no < er1.auth-no; \end{array} \quad \dots (2)$$

### 3.9.2 SQL

SQL は CODASYL による構造上の述語論理形式の問合せ言

語である。SQL検査は次の様に行け。

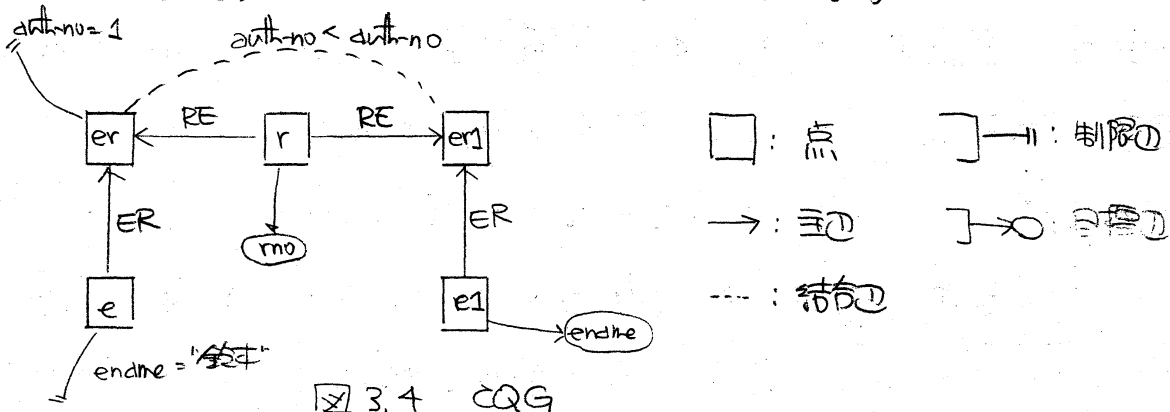
```
var (r, R1) ... (r2, R2);
get into T(a1=exp1, ..., an=expn) where qual; ... (3)
```

$R_i$ はレコード型で、 $r_i$ は $R_i$ の実現値(の)変数である。 $R_i$ のj-項目tは、 $r_i.t$ と表される。 $exp_j$ は定数、項目、又はこれ上の算術式である。 $qual$ は制限述語( $r_i.t = v$ ), 結合述語( $r_i.t = r_j.u$ ), 及びセ、ト述語 $P_j(r_i, r_j)$ と可了する論理式である。セ、ト型 $S(R_i, R_j)$ に対して、 $P_j(x, y)$ は次の様な述語である。

$$P_j(x, y) = \begin{cases} \text{true} & \text{if } x \in R_i \wedge y \in R_j \wedge (x, y) \in S, \\ \text{false} & \text{otherwise} \end{cases}$$

図3.2に対して、(2)と同じ意味を持つSQL検査は次の様に書ける。

```
var (e, EMP)(e1, EMP) (r, REP) (er, ERL) (er1, ERL);
get into T(r.rno, e1.ename) where e.ename = "鈴木" and
ER(e, er) and er.auth-no = 1 and RE(r, er) and RE(r, er1) and
ER(e1, er1) and er.auth-no < er1.auth-no; ... (4)
```



COL 検査 (4) は、図 3.4 に示す CODASYL 閉合せグラフ (COG) にあて表せる。点は 0 変数で、有向辺はセ、ト述語を表し主辺と呼ぶ。点の点線の辺は結合辺と呼ばれ、項目間の結合式を表す。

COL から COG への変換は、E 関係を表す組変数を 0 変数に主属性間の等結合式をセ、ト述語に変換することによる。参考見に行える [TAKIM80]。

### 3.2.3 アクセス経路生成

非手動的な COG から、レコード単位のアクセス手順を生成せよばなすたい。この為に、以下の目標を定め [TAKIM80]。

- (1) 中間結果格納の為に中間ファイル数の最少化
- (2) 応答時間の最少化

我々は、中間ファイル数を最少化 (=0) し、なるべくアクセスされるレコード実現値数を小さくする発見法を用いている。

まず、次の性質を持つ COG  $G$  を巡回木とする。

(i)  $G$  は主辺について順言不で、(ii)  $G$  内の全ての目標点(目標)に接する点は木の最右の根と葉間の路内であり、(iii) 全ての結合辺の両点は同一の根と葉間の路内にある。

$G$  が巡回木であるとき、ただ1つの順次ファイルに、データベースを巡回したがる目標集合  $T$  の組(レコード)を順々に書いていける [TAKIM83b]。巡回木  $G$  の根から始めて、主辺とその子

の点の対を逐型に訪ねながら、[TAKI170.03]による、そのMLを生成すると必要分のCOBOL MLを生成できる。

又、任意のCOGに対して、等価な巡航木が存在することも[TAKI170.03]に示してある。COG G内の巡回と点を逐次回訪することを許すと、全ての目標点と、結合辺に属する点とを含む路を求めたからである。或るは、なるべくアクセスされる実現値が最小となる様に、アクセスされる実現値数を制限式の置換度、セ、ト型の結合度といった統計情報[TAKI170.06]による、を記録している。

### 3.3 LDPの実現

LDPは、現在QUELとほぼ同等の能力(制限、射影、任意の結合、集約関数、更新)を持つROLを利用者に提供している。

LDPは、現在当協会のAIM(M-170F)とADBS(Acos-700)上に、PL/Iにより実現されている。LDPは、どの様なROL問合せに対しても、50秒以下の経過時間(M-170F)で数1000行のCOBOL MLを生成出来る。現在、日本語処理機能の付加と、性能向上の為に、COBOL MLを生成せずに、あらかじめコンパイルされたアクセスモジュールをサブルーティンコールする方式による強化を行っている。

#### 4. 全体キータバースプロセッサ (GDP)

GDP は、GCS と LC 層とのインタフェースとなり、以下の機能を有している。

- (i) 全体管理者 (GA) による GCS キー (GCS) 定義式から、DPI を生成し、全ての DPI に配布する。
- (ii) GCS による ROL 換算を分散処理して 解を利用者に出力する。

このについては、GCS 内の関係は、LC 層の関係の結合、射影、制限による視野とする。GCS に対する問合せは、問合せ修正手法 [STONIM] により、LC 関係を参照する問合せ (全体に問合せ) に変形出来る。本論文では、全体に問合せの分散処理手法について述べる。又、更新に要するコミットメント制御手法についても述べる。

##### 4.1 通信網 (CN)

通信網 (CN) を、次の2つの「コスト (時間)」によってモデル化する。

$C_{i\{j_1, \dots, j_n\}}(\alpha)$   $\equiv$  量  $\alpha$  を  $n (\geq 1)$  個の DPI  $DPI_{j_1}, \dots, DPI_{j_n}$  に送る全通信時間

$R_{i\{j_1, \dots, j_n\}}(\alpha)$   $\equiv$  このときの応答時間

まず、従来の広域網等による「対」通信では次の様に作る。

$$C_{i\{j_1, \dots, j_n\}}(\alpha) = m \cdot (a + b \cdot \alpha) \quad \dots (1)$$

$$R_{i\{j_1, \dots, j_n\}}(\alpha) = m \cdot (a + b \cdot \alpha) \quad (1 \leq m \leq n) \quad \dots (2)$$

ここで  $a$  と  $b$  は定数である。  $a$  は通信リニアの開設、切断の  
為の時間である。これに対して、Ethernet の様に通信媒体を  
共有しているコ-カル網、無線網では次の様に有る。

$$C_i(u_1, \dots, u_n) \alpha = R_i(u_1, \dots, u_n) \alpha = a + b \alpha \quad \dots (3)$$

即ち、全通信及び応答時間は同一で、ただ転送量  $\alpha$  にのみ依  
存する。我々は、(3) 式の成り立つ網を放送網とする。放送網  
は、DDBS の分散問合せ処理とコミットメント制御を簡単で  
かつ有効なものとする。

#### 4.2 仮定

全体に問合せ<sup>Q</sup>は、初期D-カル問合せ処理[HEVNA]は、  
全て終了し、DM間の結合式のみかた成るとする。又結合と  
しては等結合のみを考える。問合せ内の属性は、等結合述語  
内の結合属性は同一と有る様に再名前づけされて有るとする。  
この時、Qに対して以下を定める。

$\Sigma \equiv \{R_1, \dots, R_n\} \equiv Q$  の参照する関係の集合

$\pi \equiv Q$  の目標属性の集合  $A \equiv Q$  の結合属性の集合

各関係  $R_i \in \Sigma$  に対して、以下を定める。

$A_{R_i} \equiv R_i$  の結合属性集合  $\pi_{R_i} \equiv R_i$  の目標属性の集合

$\Omega_{R_i} \equiv R_i$  の属性集合 ( $= A_{R_i} \cup \pi_{R_i}$ )

$D_{R_i} \equiv R_i$  の存在するD-カル  $q_1, q_2, \dots, q_n$  (DM)

### 4.3 DM 間問合せ処理

全体との問合せのDM間処理では 次の3点を検討せねばならない。

- (i) 通信網(1対1又は放送網)
- (ii) 通信処理アルゴリズム決定法(静的又は動的)
- (iii) 各DMの実行制御手法(分散制御又は集中制御)

JDDBSでは、放送網による、完全分散制御による動的アルゴリズム決定手法を用いている。又転送于一タビ結合属性集合値のコード化による于一タビ掃取法を用いている。以下に、我々のDQPアルゴリズムを示す。

#### [DQPアルゴリズム]

(0)  $A \leftarrow \emptyset$ ; 各  $R_i \in \Sigma$  に対して、 $\hat{A}_{R_i} \leftarrow \emptyset$ ;  $\Sigma' \leftarrow \Sigma$ ; 各DM<sub>i</sub>は、他のDM内の問合せについての情報を持つ全体情報GI<sub>i</sub>を持つ。

(1) 各DM<sub>i</sub>はGI<sub>i</sub>から最少の|R<sub>i</sub>(A)| (= C<sub>in</sub>)を見つけ、又最少の|R<sub>i</sub>( $\hat{A}_{R_i}$ )| (=  $\hat{C}_{in}$ )を見つめる(ただし card( $\hat{A}_{R_i}$ )  $\geq 2$ )。C<sub>in</sub> >  $\hat{C}_{in}$  なるは (5)  $\wedge$ 。

C<sub>in</sub>  $\neq$   $\hat{C}_{in}$  も見つかりない時は、(8)  $\wedge$ 。

(2) DM<sub>i</sub>は、R'<sub>i</sub>  $\leftarrow$  R<sub>i</sub>(A); B $\Gamma_{iA}$   $\leftarrow$  R'<sub>i</sub> // R'<sub>i</sub>; (4)  $\wedge$ 。

(3) DM<sub>j</sub> (j  $\neq$  i) は、R'<sub>i</sub> を受信して、R'<sub>j</sub>  $\leftarrow$  R<sub>j</sub> [a=a] R'<sub>i</sub>; B $\Gamma_{jA}$   $\leftarrow$  R'<sub>j</sub>(A) // R'<sub>i</sub>;  $\forall a \in \hat{A}_{R_j}$  に対して、B $\Gamma_{ja}$   $\leftarrow$  R'<sub>j</sub>(a) // R'<sub>i</sub>(a);  $\forall b \in A_{R_j} - \{a\}$  に対して、|R'<sub>j</sub>(b)| を求め、B $\Gamma$ とともに応答ACK<sub>j</sub>を送る。R'<sub>j</sub>  $\leftarrow$  R'<sub>j</sub>;

\*)

$\alpha // \beta$  は、集合  $\alpha$  ( $\subseteq \beta$ ) の  $\beta$  に対する bit-map とする。

(4) 各  $DM_{R_i}$  は全ての  $ACK_j$  を待つ。  $ACK_j$  を受信したならば、 bit-map の書込を中止。  $BM_{R_i} \leftarrow BM_{R_i} \wedge BM_{j_a}$ ;  $\forall a \in \widehat{A}_{R_i} \cap \widehat{A}_{R_j}$  に対して、  $\widehat{BM}_{R_i} \leftarrow \widehat{BM}_{R_i} \wedge \widehat{BM}_{j_a}$ ; 全ての応答を受信したならば、 bit-map に基づいて、  $R_i$  を制限する。  $ACK_j$  内の結合属性情報  $\Sigma GI_{R_i}$  にセットする。  $R_i$  の値を、  $\gamma$ -トークンして各値を順番でコード化し、  $a$  を  $\bar{a}$  とする。  $a \in \mathcal{T}_{R_i}$  の時は  $\bar{a}$  を新たに  $R_i$  に加える。

$$A \leftarrow A - \{a\}; \quad \widehat{A} \leftarrow \widehat{A} \cup \{\bar{a}\}; \quad A_{R_i} \leftarrow A_{R_i} - \{a\}; \quad \widehat{A}_{R_i} \leftarrow \widehat{A}_{R_i} \cup \{\bar{a}\}; \quad (2) \wedge$$

(5)  $DM_{R_i}$  は、  $\widetilde{R}_i \leftarrow R_i[\widehat{A}_{R_i}]$  を行い  $\widetilde{R}_i$  を放送する。  $\widehat{BM}_i \leftarrow \widetilde{R}_i // \widetilde{R}_i'$ ; (7)  $\wedge$ 。

(6)  $\widehat{A}_{R_i} \cap \widehat{A}_{R_j} = \widehat{A}_{ij} \neq \emptyset$  なる  $DM_j$ ; (4) は  $\widetilde{R}_j$  を受信した。  $R_j' \leftarrow R_j[\widehat{A}_{ij} = \widehat{A}_{ij}] \widetilde{R}_i'$ ; を行う。  $\widehat{BM}_j \leftarrow R_j'[\widehat{A}_{R_i}] // \widetilde{R}_i'$  をつくる。  $\forall b \in \widehat{A}_{R_j} - \widehat{A}_{R_i}$  に対して、  $BM_{j_b} \leftarrow R_j'[b] // R_j[b]$  をつくる。 bit-map をのせて  $ACK_j$  を放送する。  $R_j \leftarrow R_j'$ ;  $\widehat{A}_{R_j} \leftarrow \widehat{A}_{R_j} \cup \widehat{A}_{R_i}$ ;

(7) 全ての  $DM_{R_i}$  は全ての  $ACK_j$  を待つ。  $ACK_j$  を受信したならば、

$$\widehat{BM}_{R_i} \leftarrow \widehat{BM}_{R_i} \wedge \widehat{BM}_j; \quad \forall b \in (\widehat{A}_{R_i} - \widehat{A}_{R_j}) \cap \widehat{A}_{R_j} \text{ に対して } \widehat{BM}_{R_i} \leftarrow \widehat{BM}_{R_i} \wedge \widehat{BM}_{j_b};$$

全ての  $ACK$  を受信したならば、 bit-map をもとにして  $R_i$  を制限する。

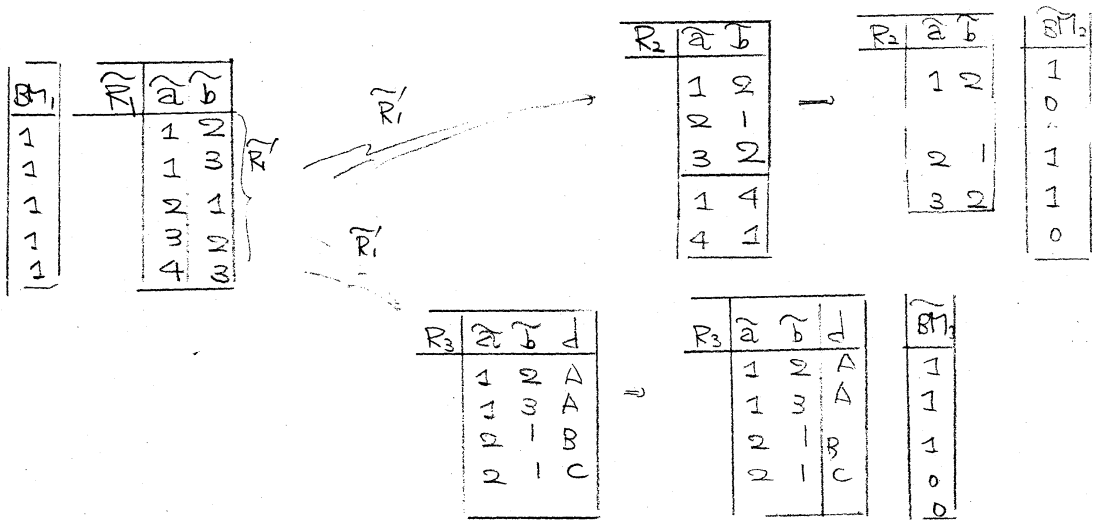
$R_i[\widehat{A}_{R_i}]$  を (4) と同様につコード化し、  $\bar{a}$  とする。  $\widehat{A} \leftarrow \widehat{A} - \widehat{A}_{R_i} \cup \{\bar{a}\}$ ;

$$\widehat{A}_{R_i} \leftarrow \widehat{A}_{R_i} - \widehat{A}_{R_i} \cup \{\bar{a}\}; \quad (2) \wedge$$

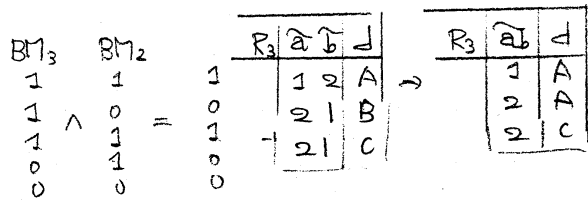
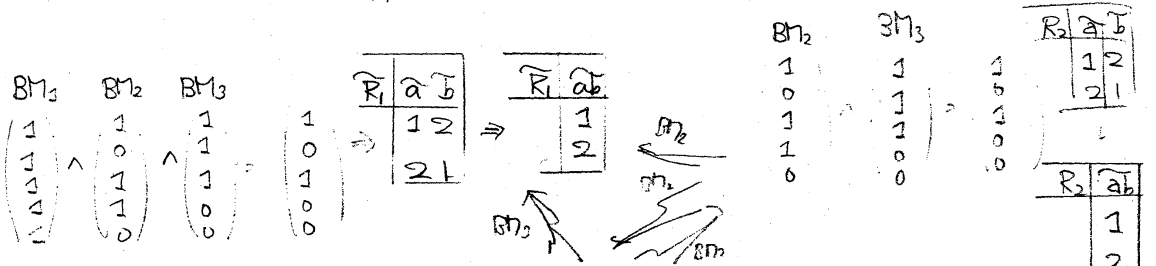
(8)  $\mathcal{T}_{R_i} \neq \emptyset$  なる  $R_i$  を利用者のいる  $DM$  に送り  $\widehat{A}$  について結合して利用者に出力する。  $\square$

図4.1にDOPアルゴリズム4の図を示す。





ω R の変遷



ω bit-map 変遷とノード化

図4.1 DOP の例

#### 4.4 複送網によるコミットメント制御

コミットメント制御とは、トランザクションが更新するオブジェクト (関係、組) を  $x_1, \dots, x_n$  ( $n \geq 1$ ) とすると、全てのオブジェクトが更新されたか、全くされなかったかを保証する為の制御である。ここで、 $x_1, \dots, x_n$  は互いに異なり、データベース  $DB_i$  に存在するとする。2相コミット制御を複送網と分散制御を用いると次の様に行える。

##### [B2P アルゴリズム]

(0) DBP アルゴリズムで、 $DB_0$  に  $x_1, \dots, x_n$  に対する更新データを導出する。 $DB_0$  の安全な領域に、これを格納する。

(i)  $DB_0$  は、更新データを乗せた  $\text{precommit}(x_1, \dots, x_n)$  を複送する。

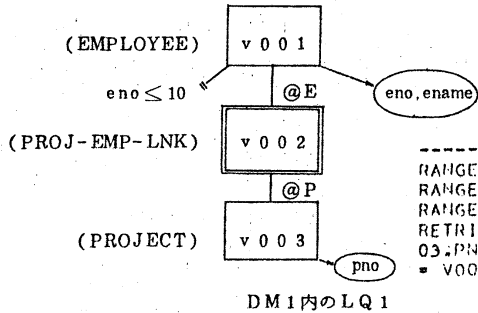
(ii) 各  $DB_i$  は、 $\text{precommit}(x_1, \dots, x_n)$  を受信したならば、 $x_i$  の更新データを安全領域  $SS_i$  (e.g. ログ) に格納する。格納出来たならば、 $\text{precommitted}_i$  を複送する。

(iii) 各  $DB_j$  は、全ての  $\text{precommitted}_i$  を待ち、全て受信したならば、 $SS_i$  内のデータをを用いて  $DB_j$  内の  $x_i$  を物理的に更新する。更新出来たならば、 $\text{ACK}_j$  を複送する。

(iv)  $DB_0$  は全ての  $\text{ACK}_j$  を受信したならば、トランザクションを commit する。□

従来の1対1通信ベースの方式では  $4m$  回の通信が必要であるが、我々のB2Pでは  $2m+1$  回の通信でよい。

(3-1)

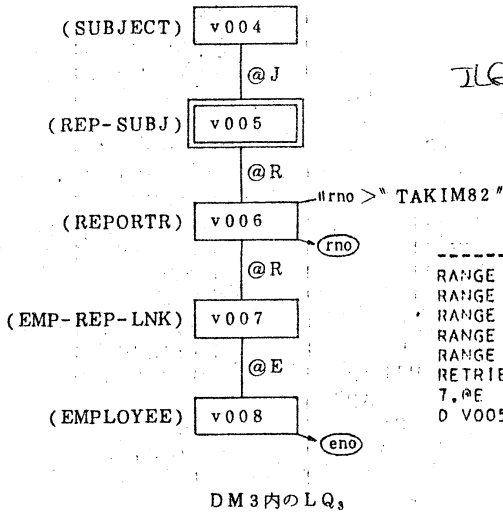


ILOPに於て LQ1 (DM1, n)

```

----- SUBMIT_LDP -----
RANGE (V001, EMPLOYEE      );
RANGE (V002, PROJ-EMP-LNK  );
RANGE (V003, PROJECT       );
RETRIEVE INTO RRELO10051 (TL001=V001.ENAME , TL002=V001.ENO , TL003=V003.PNO ) WHERE V003.PNO = V002.PNO AND V001.ENO <= 10 AND V002.ENO = V001.ENO ;
    
```

(3-2)



ILOPに於て LQ3 (DM3, n)

```

----- SUBMIT_LDP -----
RANGE (V004, SUBJECT      );
RANGE (V005, REP-SUBJ    );
RANGE (V006, REPORTR     );
RANGE (V007, EMP-REP-LNK );
RANGE (V008, EMPLOYEE    );
RETRIEVE INTO RRELO30052 (TL001=V006.RNO , TL002=V008.ENO ) WHERE V007.ENO = V008.ENO AND V006.RNO > "TAKIM82" AND V006.ENO = V007.ENO AND V005.ENO = V006.ENO AND V004.ENO = V005.ENO ;
    
```

(4)

最終結果

\*\* TEMP \*\*

TEMPN	IREPNO	IPROJNO
ITAKIZAWA	ITAKIM82A	IDDBS80
ITAKIZAWA	ITAKIM82A	IJIPNET73
ITAKIZAWA	ITAKIM82A	IMMUI80
ITAKIZAWA	ITAKIM82B	IDDBS80
ITAKIZAWA	ITAKIM82B	IJIPNET73
ITAKIZAWA	ITAKIM82B	IMMUI80
ITAKIZAWA	ITAKIM82C	IDDBS80
ITAKIZAWA	ITAKIM82C	IJIPNET73
ITAKIZAWA	ITAKIM82C	IMMUI80
ITAKIZAWA	ITANAY81A	IDDBS80
ITAKIZAWA	ITANAY81A	IJIPNET73
ITAKIZAWA	ITANAY81A	IMMUI80
ITAKIZAWA	IYOKOM82A	IDDBS80
ITAKIZAWA	IYOKOM82A	IJIPNET73
ITAKIZAWA	IYOKOM82A	IMMUI80
IYAMAMOTO	IYAMAK75	IJIPNET73
IYAMAMOTO	IYAMAK75	ISCRTY
IYAMAMOTO	IYAMAK75	I5G79
IYAMAMOTO	IYAMAK75	I5G80

(4)

$R_3' \leftarrow R_1(eno);$   
 $R_1' \leftarrow R_1(eno=eno)R_3';$   
 $B_{T1}$

IRNO	IENO	IENAME	IENO	IPNO
ITAKIM82A	11	ITAKIZAWA	11	IDDBS80
ITAKIM82B	11	ITAKIZAWA	11	IJIPNET73
ITAKIM82C	11	ITAKIZAWA	11	IMMUI80
IYAMAK75	51	IYAMAMOTO	51	IJIPNET73
IYOKOM82A	11	IYAMAMOTO	51	ISCRTY
		IYAMAMOTO	51	I5G79
		IYAMAMOTO	51	I5G80

図42 GDPの実行(2)

### 4.5 GDPの実現

GDPは現在、M-170FのAIM DB/DCを用いて、夜送通信をシミュレートし、M-170F内の4つのAIM DBS間の通信処理を実現している。GDPの実行例を図4.2に示す。

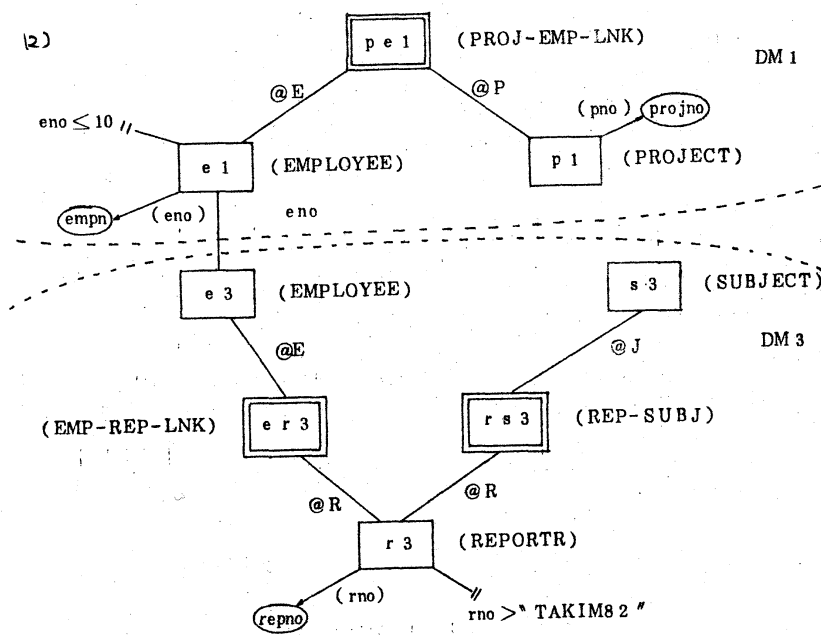
```

0)
EXEC MMUJ(USTVJA)

PLEASE LOGON
LOGON GDP_SUZUKI/MAKO;
/00010 DRP_GCSNAME11;
#00010 RANGE (R11, GRE111);
#00020 RETRIEVE INTO TEMP (R11.EMPNO, R11.REPNO, R11.PROJNO)
#00030 WHERE
#00040 R11.REPNO > "TAKIM82";
#00050 GO;
    
```

GCS階層 GREU11 (empno, repno, projno)  
の問合せ

THE ELAPSE TIME FOR RETRVP,QM.. 67



問合せの2段階による表現

図4.5 GDPの実行例

### 5. まとめと今後の課題

異種ODBSとしてのJODBSは、現在1つのホスト(M-170F)内ではあるが、4つのAIM DBSに対して、分散性と異種性を意識せずに、関係問合せを利用者は入力出来る。この意味で、異種ODBSを実現出来たと考えている。今後、性能面での評価と共に、実際のLANによる実現を検討している。AIM, ADBSといった従来の大型DBSに加えて、個人計算機ベースのDBSが今後、オフィス情報システム等で重要となることから、個人計算機(PC-5800, IBM5550)上のDBMSの実現を行っている。これらのDBSをLANで結合した多ホストの統合利用システムを現在研究開発を行っている計画である。

### 参考文献

- [CODDEXO] Codd, E.F., "A Relational Model of Data for Large Shared Data Banks," CACM, Vol. 13, No. 6, June 1970, pp. 337-387.
- [HELDG75] Held, G., "INGRES - A Relational Data Base System," AFIPS Conf. Proc., 1976, pp. 409-416.
- [HEDNA78] Hevner, A. and Yao, S.B., "Query Processing on a Distributed Databases," Proc. of the 3rd Berkeley Workshop, 1978, pp 91-107
- [CODAS73] Co dagel Date Description Language, Journal of Development, 1973

- [COLLET78] Olb, T., "The CODASYL Approach to Data Base Management," John Wiley & Sons 1978
- [STONM76] Stonebraker, M., et al., "The Design and Implementation of INGRES," ACM TODS, Vol. 1, No. 3, 1976, pp. 189-222.
- [KAMB781] Kambayashi, Y., "圧縮型ネットワークによる分散データベースの運用処理," 信学会AL81-54, 1981.
- [TAKIM780] Takizawa, M., et al., "Resource Integration and Data Sharing on Heterogeneous Resource Sharing System," Proc. of the ICCO'78, 1978, pp. 253-258
- [TAKIM79] Takizawa, M., et al., "The Four-schema Concept as the Gross Architecture of Distributed Databases and Heterogeneity Problems," JIP (IPSS), Vol. 2, No. 3, 1979, pp. 134-142.
- [TAKIM80] Takizawa, M., et al., "Query Translation in Distributed Databases," Proc. IFIP'80, 1980, pp. 451-456
- [TAKIM81] Takizawa, M., "分散型データベースシステム JEDBS-II と通信処理," 信学会AL81-22, 1981
- [TAKIM82a] Takizawa, M., "Distribution Problems in Distributed Databases," JIP (IPSS), Vol. 5, No. 3, 1982, pp. 139-147.
- [TAKIM82b] Takizawa, M., Tokotaka, M., and Suzuki, M., "CODASYL DBS に対する関係データベースシステム LDP-VIS の設計と実装" IPSS論文誌, Vol. 23, No. 3, 1982, pp. 665-675.

- [TAFIM83a] Takizawa, M. and Noguchi, S., "CODASYL DBSに對する非手続的更新の二つの一又の基本概念," (出版中) 1983
- [TAFIM83b] Takizawa, M. and Noguchi, S., "非手続的の方向合せのCODASYL DML表現," Jipde TR 02/83, 1983