

異種分散型データベースシステムとRDBS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(DDBS)技術が中核になると考えられる。DDBS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。データベースモデルとは、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], CODASYLモデル[CODAS73])の相違とする。異種のDBSからなるDDBSを、異種DDBSとする。異種DDBSを実現する為には、各DBSの異種

性、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、当協会で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の全体アーキテクチャ、第3章では異種問題の解決法、第4章では分散問題の解決について述べる。

2. JDDBSの全体アーキテクチャ

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

DDBS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、DDBS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。^(TAKI78,79)ローカル内部(LI)層は、既存DBSがDDBS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(LML)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、DDBS全体の共通モデルによって、

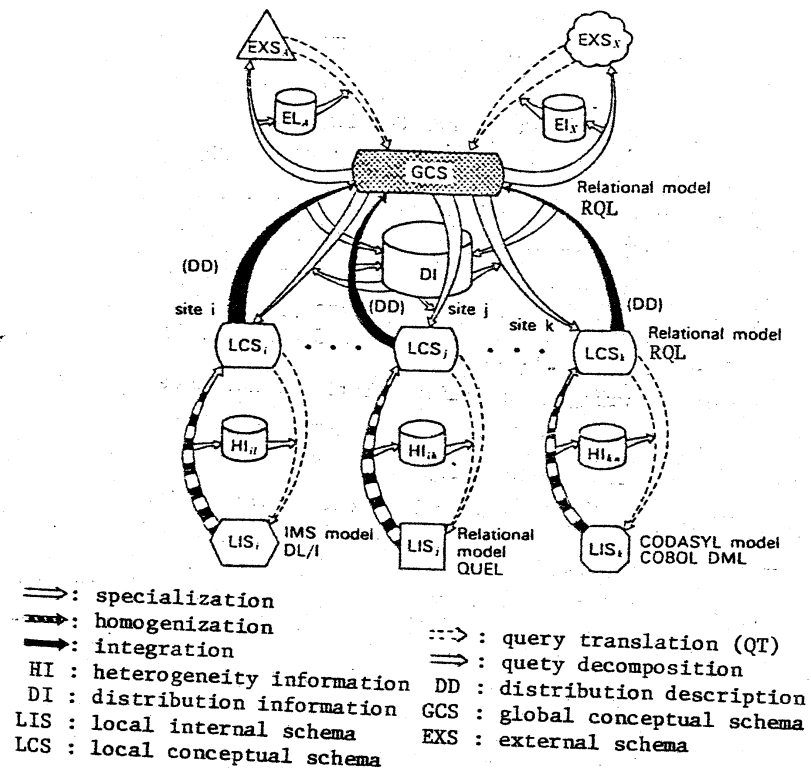


図 2.1 四階層構造

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(CORDE)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、て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、各DBS全体と一様な視野を表している。GC層は LC層と同一の関係データ構造と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、各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層は名応用に属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間の高信頼な種々の通信(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_i), ローカルデータベースプロセッサ(LDP_i), データベースシステム(DBS_i), 異種性情報(HI_i), 分散性情報(DI_i), 関係作業域(RDS_i)との6つのモジュールから成る。DBS_i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_i

は、DBS_i 上にある、LC 層を提供するインタフェースである。
 LDP_i は、LIS から LCS を生成するスキーマ変換と、LC 層の ROL 演算を LI 層の LIM に変換し、DBS_i 上で実行し、解を関係として RDS_i に格納する問合せ変換との二つの機能を備えている。
 HI_i 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_i は、LDP_i 上にある、GC 層を提供する。GDP_i 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 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_i は、GC 層の ROL 演算を、LC 層のものに変換し、DM 内/

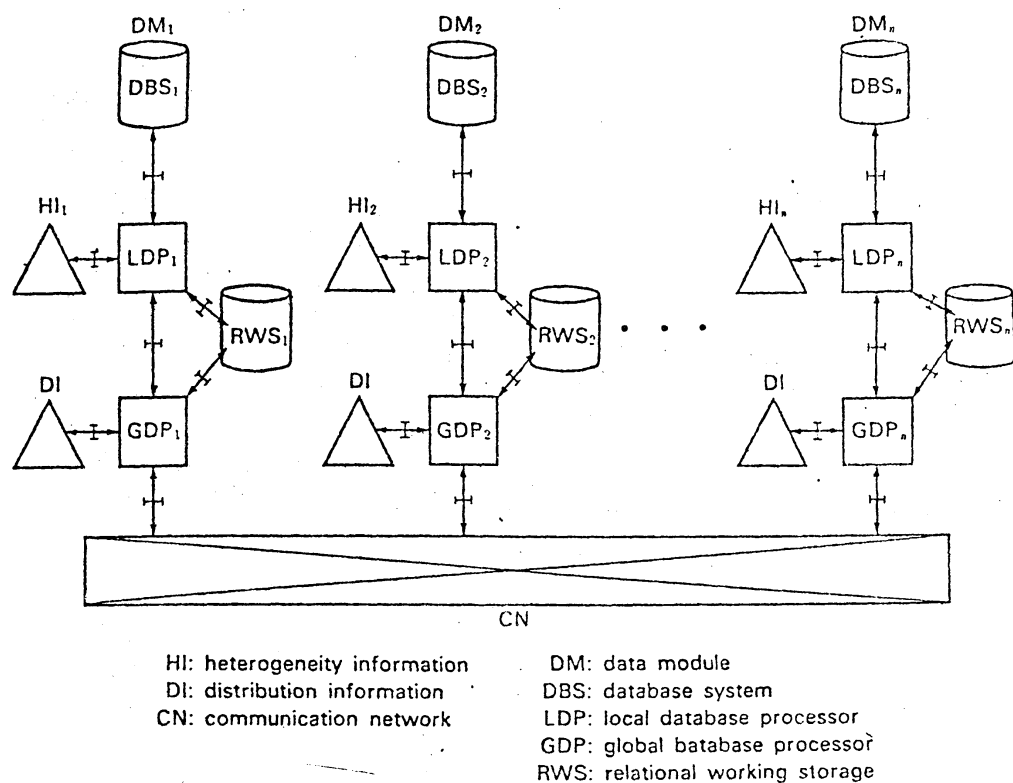


図 2.2 JDPBS のシステム構成

問通信処理により、解を得る分散問合せ処理機能を有している。これは、DIに基づいて行われる。GDP_iは、RDS_i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_iを用いる。

3. ロ-カルデータベースプロセッサ(LDP)

JDBBSでは、DBSとしてCODASYL型[CODASYL73, OUEY78]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ースシステムである[TAKIM80, 82b, 83]。LDPは、図3.1に示す様に、スキ-マ変換を行うHIP(HIプロセッサ)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。

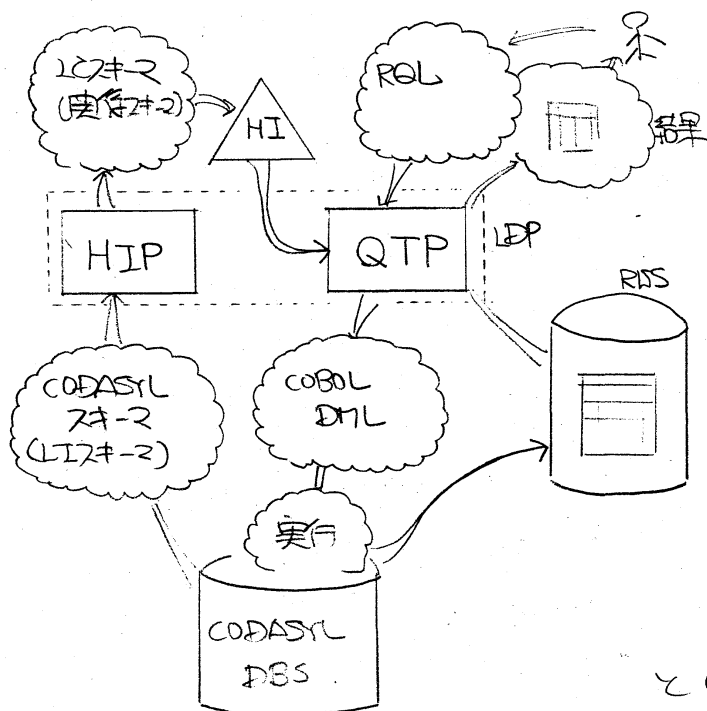


図3.1 LDP

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COBOL DMLに変換し実行させ、解を関係

としてRDSに格納する。

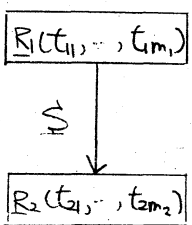
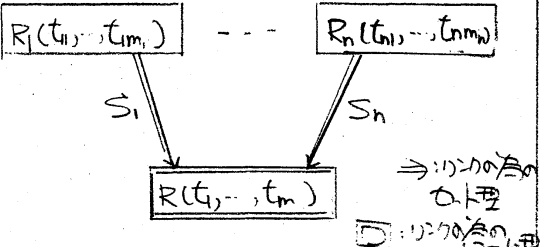
- (3) RQLは、QUELとほぼ

同等 (i.e. 制限、射影、結合、集約関数) 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)を生出し、LCSからLCSとLISの対応情報をHIとして生かす。CODASYLモデルとLCSモデルとの対応は[TAKIM83a]で論じらる。表3.1の表に示す。図3.2は、従業員、プロジェ

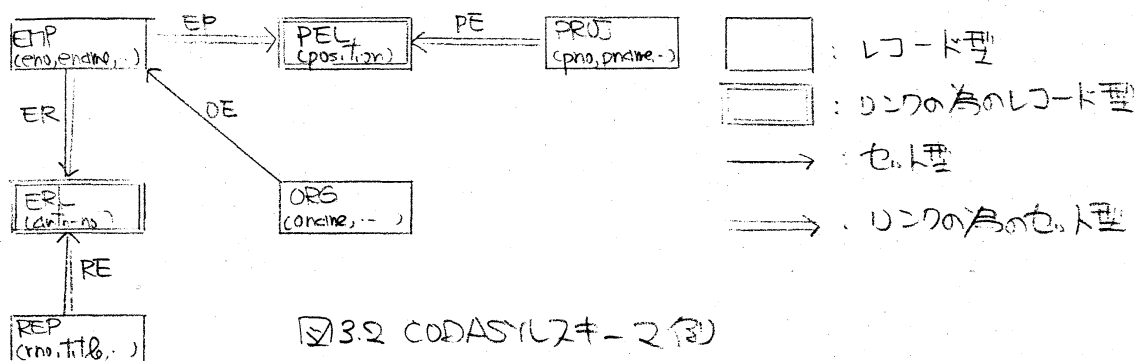
表3.1 CODASYLデータ構造とLCSデータ構造の対応

CODASYLデータ構造	LCSデータ構造
レコード型 $R(t_1, \dots, t_m)$	E型関係スキーマ $R(@R, t_1, \dots, t_m)$
セット型 	E型 $R(@R, t_1, \dots, t_{m_1})$ $R_2(@R_2, t_2, \dots, t_{m_2})$ B型 $S(@R_1, @R_2)$
リンク型 	E型 $R_1(@R_1, t_1, \dots, t_{m_1})$ $R_n(@R_n, t_{n_1}, \dots, t_{n_{m_n}})$ G型 $R(@R, \dots, @R_n, t_1, \dots, t_m)$

$@R$ は、レコード型 R のキーを値として取る関数で主属性とする。

以上、所居、論文を表すCODASYLスキーマ例である。これは

HIPにて、図3.3に示すLCスキーマとなる。



E型	EMP (<u>eno</u> , ename, ...)	B型	DE (<u>eo</u> , <u>ee</u>)
	REP (<u>er</u> , rno, title, ...)	G型	ERL (<u>ee</u> , <u>er</u> , auth-no)
	PROJ (<u>ep</u> , pno, pname, ...)		PEL (<u>ep</u> , <u>ee</u> , position)
	ORG (<u>eo</u> , oname, ...)		

図3.3 図3.2のLCS

3.2 QTP

問合せ変換アロセ、サQTPは、LC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、[TAKIM83]を参照されたい。

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) アロセス単位(集合単位とレコード単位)との2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

COL からレコード単位のアクセス手順を生成する方法を用いた。

3.2.1 RQL

RQL は QUEL [HEEDG75] と類似した述語論理形式の言語である。RQL による LCS 上の検索は 次の様に書かれる。

$$\text{range } (r_1, R_1) \dots (r_k, R_k);$$

$$\text{retrieve into } T(a_1 = \text{exp}_1, \dots, a_k = \text{exp}_k) \text{ where } \text{qual};$$
(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 ($T(a_1, \dots, a_k)$) であり、 qual は条件式、 $(a_1 = \text{exp}_1, \dots, a_k = \text{exp}_k)$ は目標リストである。関係 R_i の属性 a_i は、 $r_i.a_i$ と表され、 exp_i は定数、属性、又はこれ上の算術式である。

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. $x \cdot 0.5 + y \cdot 0.5$) 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「鈴木を第 1 著者とする論文の共著者名を求めよ」は RQL で次の様に書ける。

$$\text{range } (e, \text{EMP}) (e1, \text{EMP}) (r, \text{REP}) (er, \text{ERL}) (er1, \text{ERL});$$

$$\text{retrieve into } T(r.\text{rno}, e1.\text{ename}) \text{ where } e.\text{ename} = \text{"鈴木"} \text{ and}$$

$$e.\text{AE} = er.\text{AE} \text{ and } er.\text{auth-no} = 1 \text{ and } er.\text{ER} = r.\text{ER} \text{ and } r.\text{ER} = er1.\text{ER} \text{ and}$$

$$er1.\text{AE} = e1.\text{AE} \text{ and } er.\text{auth-no} < er1.\text{auth-no};$$
(2)

3.2.2 COL

COL は CODASYL 型 - 7 構造上の述語論理形式の問合せ言

語である。COL検査は次の様に行ける。

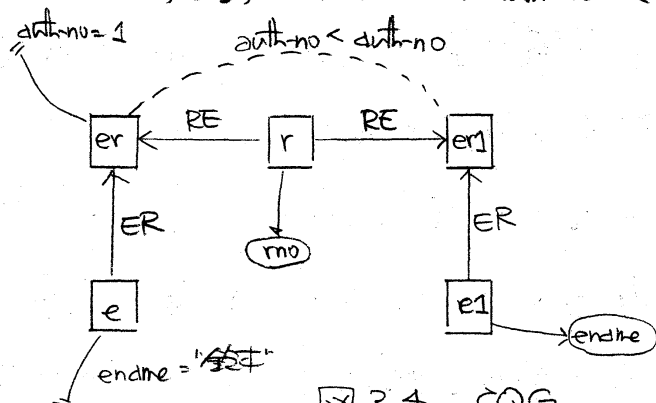
var (r, R₁) ... (r₂, R₂);
get into T(a₁=exp₁, ..., a₂=exp₂) where qual; } ... (3)

R_iはレコード型で、r_iはr_iの実現値(の要素)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は制限主語(r.t=r), 結合主語(r.t=r₁.u), 及びセ、ト主語P₃(r, r₁)とから成る論理式である。セ、ト型S(R₁, R₂)に対して、P₃(r, r₁)は次の様な主語である。

$$P_3(x, y) = \begin{cases} \text{true} & \text{if } x \in R_1 \wedge y \in R_2 \wedge (x, y) \in S, \\ \text{false} & \text{otherwise} \end{cases}$$

図3.2に対して、(2)と同じ意味を持つCOL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)



□: 点]→: 制限の
→: 主語]×: 結合の
---: 結合

図3.4 CQG

COL 模写 (4) は、図 3.4 に示す CONASYL 閉合せりグラフ (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。

ROL から COG への変換は、E 関係を表す組変数も 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り容易に行える [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ねばならない。この為に、以下の目標を定め [TAKIM80]。

- (1) 中間結果格納のための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少なくする発見法を用いている。

まず、次の性質を持つ COG G を巡航木とする。

(i) G は主辺について順言不で、(ii) G 内の全ての目標点 (目標辺に属する点) は木の最右の根と葉間の路内にあり、(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 が巡航木であるとき、ただ 1 つの順次ファイルに、データベースを巡航したがる目標集合 T の組 (レコード) を順々に書いていける [TAKIM83b]。巡航木 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、[TAKIM83]によって DML を生成すると必要なる COBOL DML 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 [TAKIM83] に示してある。COG G 内の主辺と点を逐次回訪することと許すと、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。我々は、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 [TAKIM86] によって見積っている。

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 RQL 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 によって実現されている。LDP は、どの様な RQL 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 DML 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 DML 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チンコールする方式による強化を行っている。

4. 全体データベースプロセッサ (GDP)

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

(i) 全体管理者 (GA) による GCS スキーマ (GCS) 定義式から、DML を生成し、全ての DM に配布する。

(ii) GCS による ROL 検索を分散処理して 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 [5] のために、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、更新にさけ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。

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

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

まず、従来の広域網等による1対1通信では次の様になる。

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

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

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
共有しているローカル網、無線網では次の様になる。

$$C_i(y_1, \dots, y_n) \alpha = R_i(y_1, \dots, y_n) \alpha = a + b \alpha \quad \dots (3)$$

即ち、全通信及び応答時間は同一で、ただ転送量 α にのみ依
存する。我々は、(3) 式の成り立つ網を収送網とする。収送網
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^①は、初期ローカル問合せ処理[EDNA 系]は、
全て終了し、DM 間の結合式のみかち成るとする。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なる様に再名前づけされているとする。
この時、①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 ($= A_{R_i} \cup \pi_{R_i}$)

$D_{R_i} \equiv R_i$ の存在するデータベース (DM_i)

4.3 DM 間問合せ処理

全体問合せの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) 通信処理スケジュール決定法(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スケジュール決定法を用いている。又転送す-タ及び結合属性集合値のコード化によるす-タ圧縮法を用いている。以下に、我々のDOPアルゴリズムを示す。

[DOPアルゴリズム]

- (0) $\hat{A} \leftarrow \emptyset$; 各 $R_i \in \Sigma$ に対して、 $\hat{A}_{R_i} \leftarrow \emptyset$; $\Sigma' \leftarrow \Sigma$; 各DM_i は、他のDM内の問合せについての情報を持つ全体情報GI_iを持つ。
- (1) 各DM_i はGI_iから最少の|R_i[Q]| (= C_{in}) を見つけ、又最少の|R_i[Q̂_{R_i]| (= Ĉ_{in}) を見つける(ただし card($\hat{A}_{R_i}$) ≥ 2)。C_{in} > Ĉ_{in} ならば (5)へ。C_{in} と Ĉ_{in} も見つからない時は、(8)へ。}
- (2) DM_i は、R_i' ∈ R_i[Q]; B₁_{in} ← R_i' // R_i' (4)へ。
- (3) DM_j (j ≠ i) は、R_i' を受信して、R_j' ∈ R_j[Q = Q] R_i'; B₁_j ← R_j'[Q] // R_i'; $\forall \hat{a} \in \hat{A}_{R_i}$ に対して、B₁_j_â ← R_j'[â] // R_j[â]; $\forall b \in A_{R_j} - \{a\}$ に対して、|R_j[b]| を求め、B₁とともに応答ACK_jを送る。R_j ← R_j';

*)

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

(4) 各 DM_k は全ての ACK_j を待つ。 ACK_j を受信したならば、bit-map の書きこみ。 $BM_{ka} \leftarrow BM_{ka} \wedge BM_{ja}$; $\forall a \in \widehat{AR}_k \cap \widehat{AR}_j$ に対して、 $\widehat{BM}_{ka} \leftarrow \widehat{BM}_{ka} \wedge \widehat{BM}_{ja}$; 全ての応答を受信したならば、bit-map により、 R_k を制限する。 ACK_j 内の結合属性情報 πGI_k にセットする。 R_{ka} 内の値を、ソートして、各値を順番でコード化し、 a を $\bar{a}$ とする。 $a \in \pi R_k$ の時は $\bar{a}$ を新たに R_k に加える。

$A \leftarrow A - \{a\}$; $\bar{A} \leftarrow \bar{A} \cup \{\bar{a}\}$; $AR_k \leftarrow AR_k - \{a\}$; $\widehat{AR}_k \leftarrow \widehat{AR}_k \cup \{a\}$; (2) $\wedge$ 。

(5) DM_k は、 $\widehat{R}_i \leftarrow R_i[\widehat{AR}_k]$ を作り $\widehat{R}_i$ を放送する。 $\widehat{BM}_i \leftarrow \widehat{R}_i // \widehat{R}_i$; (7) $\wedge$ 。

(6) $\widehat{AR}_k \cap \widehat{AR}_j = \widehat{A}_{kj} \neq \emptyset$ なる DM_j (4) は $\widehat{R}_i$ を受信した。 $R_j' \leftarrow R_j[\widehat{A}_{kj} = \widehat{A}_{kj}] \widehat{R}_i$; を行う。 $\widehat{BM}_j \leftarrow R_j'[\widehat{AR}_k] // \widehat{R}_i$ をつくる。 $\forall b \in \widehat{AR}_j - \widehat{AR}_k$ に対して、 $BM_{jb} \leftarrow R_j'[b] // R_j[b]$ をつくる。 bit-map のセットした ACK_j を放送する。 $R_j \leftarrow R_j'$; $\widehat{AR}_j \leftarrow \widehat{AR}_j \cup \widehat{AR}_k$;

(7) 全ての DM_k は全ての ACK_j を待つ。 ACK_j を受信したならば、

$\widehat{BM}_k \leftarrow \widehat{BM}_k \wedge \widehat{BM}_j$; $\forall b \in (\widehat{AR}_k - \widehat{AR}_j) \cap \widehat{AR}_j$ に対して、 $\widehat{BM}_{kb} \leftarrow \widehat{BM}_{kb} \wedge \widehat{BM}_{jb}$;

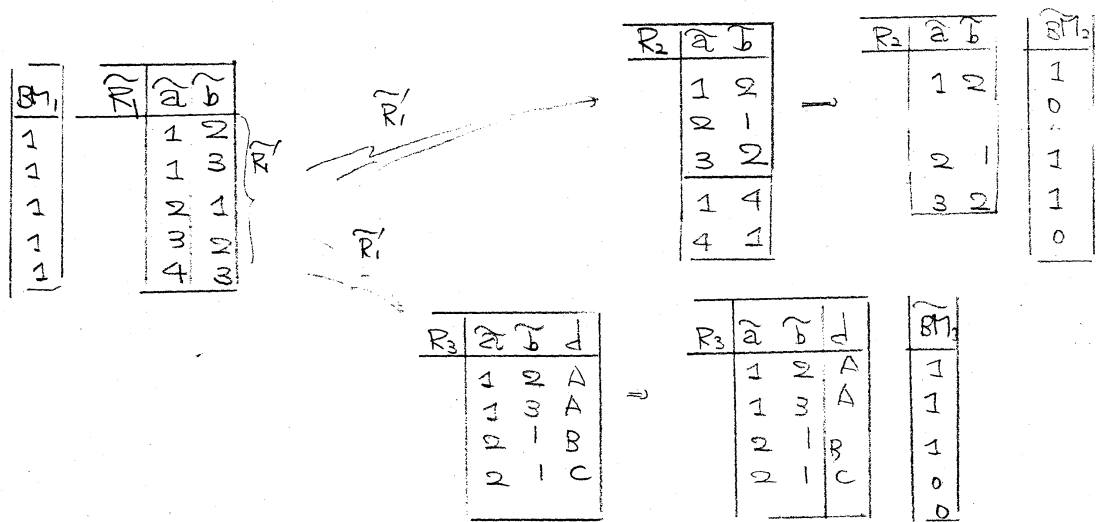
全ての ACK を受信したならば、bit-map をもとにして R_k を制限する。

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

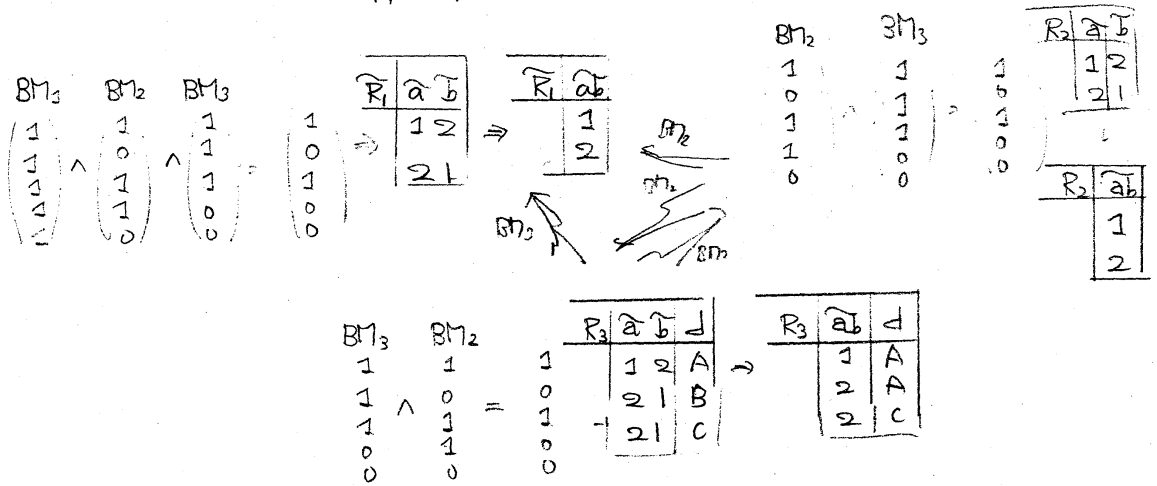
$\widehat{AR}_k \leftarrow \widehat{AR}_k - \widehat{AR}_k \cup \{\bar{a}_k\}$; (2) $\wedge$ 。

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

図4.1にDOPアルゴリズムの図を示す。



ω $\widetilde{R}_1$ の変換



ω bit-map 変換とコード化

図4.1 DOP の例

4.4 放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、全くされないかを保証する為の制御である。ここで、 $x_1, \dots, x_n$ は互いに異ったデータモジュール内に存在するとする。2相コミット制御を、放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]

(i) DBP アルゴリズムで、 DM_0 に $x_1, \dots, x_n$ に対する更新データを導出する。 DM_0 の安全な領域に、これを格納する。

(ii) DM_0 は、更新データを乗せた $\text{precommit}(x_1, \dots, x_n)$ を放送する。

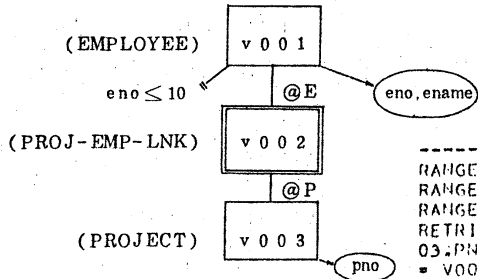
(iii) 各 DM_i は、 $\text{precommit}(x_1, \dots, x_n)$ を受信したならば、 x_i の更新データを安全領域 SS_i (e.g. ログ) に格納する。格納出来たならば、 precommitted_i を放送する。

(iv) 各 DM_j は、全ての precommitted_i を待ち、全て受信したならば、 SS_i 内のデータを用いて DBS_j 内の x_i を物理的に更新する。更新出来たならば、 Ack_j を放送する。

(v) DM_0 は全ての 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)



DM1内のLQ1

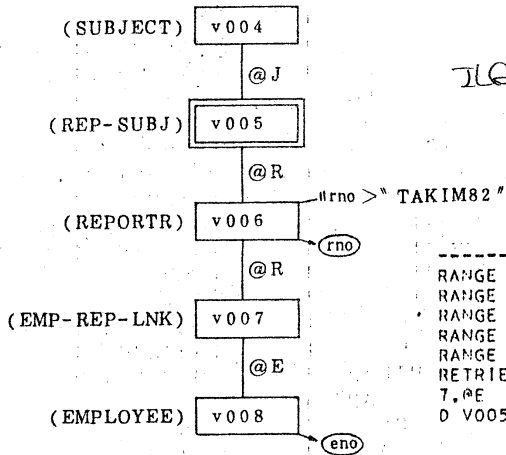
ILOPに於けるLQ1 (DM1, 1)

```

----- SUBMIT-LDP -----
RANGE (V001, EMPLOYEE      );
RANGE (V002, PROJ-EMP-LNK  );
RANGE (V003, PROJECT       );
RETRIEVE INTO RREL010051 (TL001=V001.ENAME, TL002=V001.ENO, TL003=V003.PNO) WHERE V003.PNO = V002.PNO AND V001.ENO <= 10 AND V002.PNO = V001.PNO;

```

(3-2)



DM3内のLQ3

ILOPに於けるLQ3
DM3, 1

```

----- SUBMIT-LDP -----
RANGE (V004, SUBJECT      );
RANGE (V005, REP-SUBJ     );
RANGE (V006, REPORTR      );
RANGE (V007, EMP-REP-LNK  );
RANGE (V008, EMPLOYEE     );
RETRIEVE INTO RREL030052 (TL001=V006.RNO, TL002=V008.ENO) WHERE V007.PNO = V008.PNO AND V006.RNO > "TAKIM82" AND V006.PNO = V007.PNO AND V005.PNO = V006.PNO AND V004.SNO = V005.SNO;

```

(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	ISCRITY
IYAMAMOTO	IYAMAK75	ISG79
IYAMAMOTO	IYAMAK75	ISG80

(4)

 $R_3 \leftarrow R_1[eno];$ R_3' $R_1 \leftarrow R_1[eno=eno]R_3';$ R_1

IRNO	IENO
ITAKIM82A	11
ITAKIM82B	11
ITAKIM82C	11
IYAMAK75	51
IYOKOM82A	11

ENAME	IENO	IPNO
ITAKIZAWA	11	IDDBS80
ITAKIZAWA	11	IJIPNET73
ITAKIZAWA	11	IMMUI80
IYAMAMOTO	51	IJIPNET73
IYAMAMOTO	51	ISCRITY
IYAMAMOTO	51	ISG79
IYAMAMOTO	51	ISG80

図4.2 GDPの実行例 (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の使用例を図4.2に示す。

(1)

EXEC MMUI(USTVJA)

PLEASE LOGON

LOGON GDP SUZUKI/MAKO;

/00010 GRP_GCSNAME11;

#00010 RANGE (R11, GRE111);

#00020 RETRIEVE INTO TEMP (R11.EMPNO, R11.REPNO, R11.PROJNO)

#00030 WHERE

#00040 R11.REPNO > "TAKIM82";

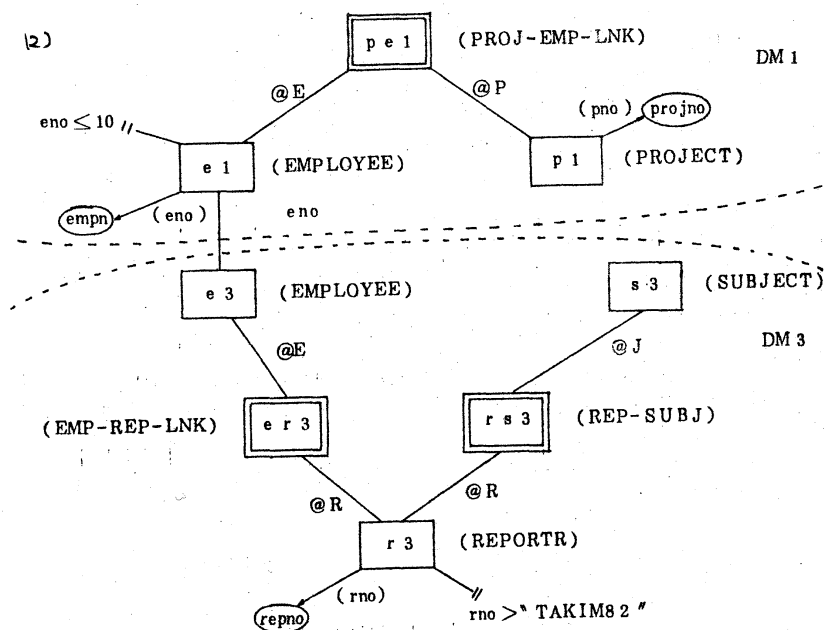
#00050 GO;

GCS条件 GRE111 (emph, repno, projno)

への向合せ

THE ELAPSE TIME FOR RETRVP,QM.. 67

(2)



向合せの2に関連になる表現

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-7800, IBM5550)上のDBMSの実現を言っている。これらのDBSをLANで結合したデータベースの統合利用システムを現在研究開発を行っており、この計画である。

参考文献

- [CODEX70] 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] Codasyl Data Description Language, Journal of Development, 1973

- [OLLET78] Olb, T., "The CODASYL Approach to Data Base Management," John Wiley & Sons 1978
- [STONE76] Stonebraker, M., et al., "The Design and Implementation of INGRES," ACM TODS, Vol. 1, No. 3, 1976, pp. 189-222.
- [KAMBI81] Kambayashi, Y., "圧縮型準結合による分散データベースシステムの質問処理," 信学会AL81-54, 1981.
- [TAKIM78] Takizawa, M., et al., "Resource Integration and Data Sharing on Heterogeneous Resource Sharing System," Proc. of the ICCS'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 と通常処理," 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., Yokobake, M., and Suzuki, M., "CODASYL DBS に対する関係インテグレーションシステム 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に対する非手続的更新プロセスの基本概念," (出版中) 1983
- [TAFIM83b] Takizawa, M. and Noguchi, S., "非手続的グラフ向合せのCODASYL DML表現," Jipde TR 02/83, 1983