

解説

異種分散型データベースシステム†



滝 沢 誠††

1. はじめに

計算機技術の進歩のなかで、VLSI などのハードウェア技術と通信網の発展によって、従来の集中型システムから、複数の計算機からなる分散型システムが経済性、ソフトウェア生産性、発展性、信頼性、可用性、安全性などの点から優れてきている。今後の情報システムでは、計算機サービスを提供するサーバ群とワークステーションが通信網で相互結合され、利用者は、種々のサービスをワークステーションをとおして利用できる。種々のサービスのなかで、データベースシステムが、多くの計算機応用で重要である。

こうしたなかで、複数のデータベースシステムからなる分散型システムの実現が試みられてきている。一つは、性能、信頼性、安全性、コストなどの点から、データを複数のデータベースシステムに最適に分散するトップダウン的方法である。この型のシステムは、一般に同種の関係型データベースシステム⁷⁾から構成され、関係をシステムの目的にあうように分割、重複させる^{4), 20), 24), 26)}ことが問題となる。もう一つは、既存のデータベースシステムのサービスを継続させながら、複数のデータベースシステム間の応用をサポートしていくとするボトムアップ的方法である。この型では、異種のデータベースシステムを考える必要があり、異種性の解決が重要な課題となる。前者のシステムについては、すでに、紹介されている^{23), 24)}ので、ここでは後者のシステムについて述べる。

まず、基本用語の定義を 2. で行い、3. ではこれまでの分散型データベースシステムの研究開発について述べる。4. では、分散型データベースシステムの論理的な構造について述べる。5. では、データベースシステムの異種性の解決方法について述べる。6. では、システム例を簡単に述べる。

2. 基本定義

【定義】分散型データベースシステムとは、1) 複数のデータベースシステムが通信網で相互接続された形態で、2) 利用者にあたかも一つのデータベースシステムかのようなサービスを提供するシステムである。

【定義】データベースシステムとは、あるデータモデルを提供するシステムである。

データモデルは、データ構造、操作演算、インテグリティ制約で与えられるもので、利用者からみたデータベースシステムの構造と動作のモデルである。データ構造には、時間不変なスキーマと、実際にデータを与えた時間可変なスキーマ例とがある。データ言語はデータ構造を操作する言語である。データモデルには、関係型⁷⁾、網型^{8), 34)}という型があり、各型はデータ言語で特徴づけられる。各型ごとに標準化が進み、関係型は SQL¹⁰⁾ を、網型は DDL⁸⁾ と DML³⁴⁾ を提供している。

分散型データベースシステムをボトムアップ的に設計するとき、データベースシステム相互の類似性と異種性が問題となる。類似性には、名前、データ構造、定義域、データ言語の類似性がある。定義域の類似性は、二つの定義域が共通部分をもつことである。

異種性としては、以下の相違がある。

1) 名前の相違。実世界の同一実体に異なった名前を与えている場合と、異なった実体に同一の名前を与えている場合がある。システム全体で一意的な名前を与えることにより解決できる。

2) 定義域の相違。定義域の異なる類似した二つの実体に対して、これらを一般化した型を設け、isa 階層^{10), 21), 36)}を構成する方法である。

3) データ表現の相違。データの精度、単位、値の相違である。これは、データ表現間の変換を、表または手続きにより行う方法がある。

4) データ構造の相違。あるデータモデルの実体を、他のデータモデルでは関連とみるといった実世界

† Heterogeneous Distributed Database Systems by Makoto TAKIZAWA (Information and Systems Engineering Department, Tokyo Denki University).

†† 東京電機大学理工学部経営工学科

に対する視点の相違である。分散型データベースシステムに対して、一つの視点を設ける必要がある。

5) データ言語の相違。SQL, DML といったデータ言語の相違である。言語では、参照するデータ構造、手続き的か非手続き的か、集積関数、グループ化機能などの相違がある。

二つのデータベースシステムが構文的に異種であるとは、おのおののデータモデルの型、すなわち、データ言語が相違していることとし、他の相違を意味的異種性とする。

データベースシステムは、データの論理的、物理的独立性を提供するものであったが、分散型データベースシステムでは、新たに二つの独立性が必要となる。ここで、異種性独立性とは、分散型データベースシステムの利用者が、各データベースシステムの型、すなわちデータ言語が変化しても影響されないことである。分散性独立性とは、構成するデータベースシステムのデータの意味が変化しても、利用者に影響を与えないことである。

3. 研究開発状況

A. 第1期分散型データベースシステム(1976~80)

例として、SDD-1²⁵⁾, Polypheme¹⁾ があり、同一の関係型データベースシステムを広域網で結合し、複数のデータベースシステム内の関係を参照する問い合わせを分散処理させることができた。通信網の低速性によって十分な性能が得られなかったが、分散型データベースシステムの表現の可能性を示したものである。

B. 第2期分散型データベースシステム(1980~)

(1) 高性能、高信頼システム

トランザクションの高速、高信頼な実行を目的にしたシステムである。System R^{*21), 23)}, Encompass²⁾, 分散型 INGRES²⁶⁾, DDM⁴⁾ などが例である。関係型データベースシステムを、LAN などの高速通信網に結合し、関係を分割、冗長化して分散させ、システムの信頼性、可用性を向上させ、並行処理により性能向上を目指している。Encompass などの商用システムが登場してきている。今後、オンラインシステムなどで、トップダウン的に利用されていくものと思われる。

(2) 異種分散型データベースシステム

関係型だけではなく、網型などの種々のデータベースシステムの相互結合を目的としているもので、例として、Multibase^{11), 19), 25)}, Sirius-Delta¹³⁾, DDTS¹²⁾,

HD-DBMS³⁾, 多データベースシステム²²⁾, JDD-BS²⁷⁾⁻³⁰⁾, 相互運用可能データベースシステム³⁵⁾ などがある。既存のデータベースシステムを新しい応用のなかで活用していく場合に重要なシステムである。

4. 論理的構造

分散型データベースシステムの論理的構造を考えると、データモデル階層とシステムのモジュール構成の二つの視点がある。

4.1 データモデル階層

各データベースシステムのデータモデルと、異種性と分散性独立なデータモデルの差は、いくつかの階層を設けることにより解決が容易となる。この階層構造には、大別して、統合型と連邦型の二つがある。

A. 統合型分散型データベースシステム

統合型は、物理的に分散したデータベースシステムを、ある共同体の要求に合う一つの全体モデルを設けることにより論理的に集中化したシステムである。統合型は、図-1 に示す4つの階層²⁷⁾から構成される。各階層は、下位層の提供するデータモデルを用いて、上位層にある価値を付加したデータモデルを提供する。

ローカル層では、各データベースシステムのデータモデル(LM)が上位層に提供される。ローカル層は、各データベースシステムの外部層である。

ローカル概念層では、ローカル層から提供されるデータモデルに対して、分散型データベースシステム全体で共通の型のデータモデル(LCM)が提供される。この層の利用者は、各データベースシステムをあたかも同型のデータモデルかのようにみることができる。すなわち、各データベースシステムを、構文的な異種

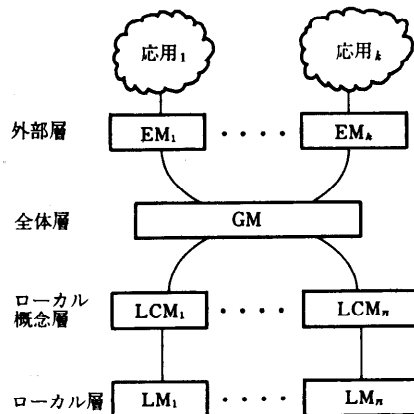


図-1 階層構造

性とは独立に、同一のデータ言語により操作できる。

全体層では、ある共同体にとって意味のある全体データモデル (GM) が、ローカル概念層のデータモデルの集合に対して定義される。共同体の視点のもとで、データモデル間の類似性を求め、意味的異種性を解決することにより一つのデータモデルが提供される。この層の利用者は、分散型データベースシステムを、データベースシステムの異種性と分散性と独立に、あたかも一つのデータベースシステムかのように操作できる。

外部層では、各応用が必要とするデータモデル (EM) が提供される。応用は、データベースシステムの異種性と分散性独立に、かつ、論理独立に分散型データベースシステムを操作できる。

統合型分散型データベースシステムは、同一分野、企業といった、類似性の高い、比較的小数のデータベースシステムの統合に適している。

B. 連邦型分散型データベースシステム

連邦型分散型データベースシステムでは、全データベースシステムを一つの共同体に論理的に集中化しない^{15), 22)}。種々の共同体の緩やかな連合が連邦である。複数の全体モデルが、ローカル概念モデルと他の全体モデル上に定義される (図-2)。したがって、外部層と全体層の区別はない。このシステムでは、データモデルの操作演算に加えて、データモデル間でのデータの複写 (ダウンロード、アップローディング)、転送などのデータモデル間の入力、輸出演算¹⁵⁾が必要になる。巨大システムでは、膨大な数のデータベースシステムをもつので、これらに対して唯一つの全体モデルを構成することが困難である。こうした巨大システムに対しては、連邦型が適している。

4.2 システムのモデル

分散型データベースシステムは、ローカルデータ

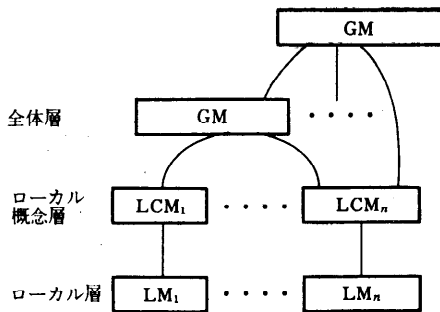


図-2 階層構造

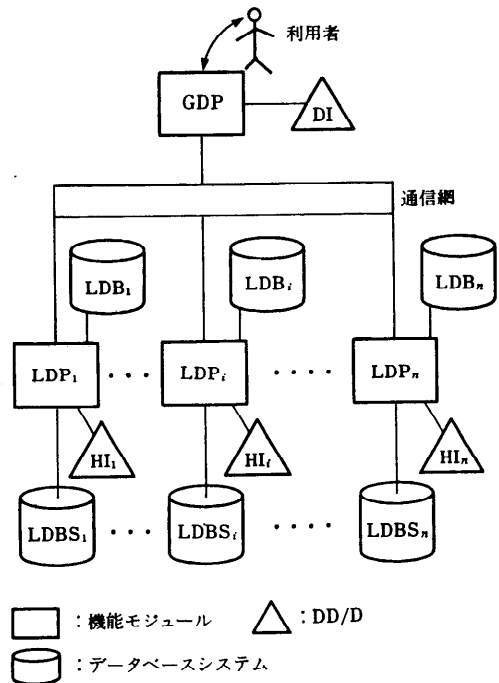
ベースシステム (LDBS), ローカルデータベースプロセッサ (LDP), 全体データベースプロセッサ (GDP), 異種性情報 (HI), 分散性情報 (DI) から構成される (図-3)。

LDBS は、従来のデータベースシステムである。

LDP は、各 LDBS に対して共通の型のローカル概念モデルを利用者に提供する。LDP はデータベース LDB をもち、新しいデータベースの格納、導出結果の格納、ローカル概念モデルの全機能をもたない LDBS に対して、LDBS からデータを導出し、この機能を実行するのに用いられる。たとえば、網型の LDBS に対して、検索の閉包性を LDP が提供する。

HI と DI は DD/D である。HI はローカル層とローカル概念層の間の写像情報であり、データ構造とデータ操作演算の対応情報をもっている。DI は全体層とローカル概念層の間の写像情報であり、各 LDBS の所在などの情報をもっている。

GDP は利用者に全体モデルを提供する。利用者のデータ操作演算 (トランザクション) を、複数の LDP で協同動作させて解を求める。利用者のトランザクションの分散実行モデルには、サーバモデルとプロセスモデルの二つがある。サーバモデルでは、GDP がトランザクションを実行し、データ操作が必要となると



□ : 機能モジュール △ : DD/D
 ○ : データベースシステム

図-3 分散型データベースシステムの構成

LDP にデータ操作要求を送信して LDP から結果を応答として受け取る。プロセスモデルでは、トランザクションは、各 LDP ごとの部分トランザクションに分割され並行実行される。前者は、実現が容易であるが、ハンドシェイクによる通信のオーバーヘッドがあり効率的でない。後者は、並行実行できる利点があるが、各部分トランザクション間の同期をとるための通信の制御が必要になる。

各機能をシステムのホスト、ワークステーションといった構成要素にどのように配置するかが、システムの実現時に問題となる。

5. 共通モデルとスキーマ変換

異種性の問題を解決するには、1)共通のデータ言語を提供し、2)異なったデータモデル間の類似性を識別し、相違を解決するために、共通型のモデルが必要となる。1)としては、データ構造の簡潔性と言語の非手続き性の点から、関係型が望ましい。2)では、データモデルがより多くの意味を表現できる必要がある。これまでに、実体-関連 (ER) モデル⁹⁾、関数型モデル Daplex¹⁹⁾、Prolog¹⁷⁾ のような論理型言語が用いられてきている。例として、図-4 に示す網型と関係型のスキーマをもったデータベースシステムを考える。

A. 関係型モデル

図-4(b)に対する関係型スキーマを以下に示す。親子関係は、外来キー dname により示される。外来キーについての参照制約¹²⁾により、親子型のメンバシップクラスの制約を示せる。

Dept (dname, location)

Emp (eno, ename, age, dname)

B. ER モデル

ER モデルは、実世界の基本要素である実体と実体間の関連を示す二つの要素からなる。同じ性質の実体の集合を実体型という。実体型間の関係を関連型という。図-5 に図-4(b) の ER スキーマを示す。箱が実

(a) 関係型スキーマ

Author (ano, name, address)

Books (bno, title, cost)

AB (ano, bno)

(b) 網型モデル

Dept (dname, location)

Emp (eno, ename, age)

図-4 網型と関係型スキーマ

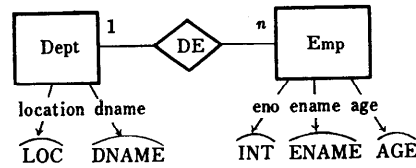


図-5 ER スキーマ

体型を、菱形が関連型、丸が定義域、矢が属性を示す。HD-DBMS, DDTS は、共通型モデルとして、ER モデルを用いている。

C. 関数型モデル

Multibase の共通型データモデルである関数型モデル Daplex は実体から構成される。実体の性質は関数で示され、同じ性質をもつ実体の集合を実体型という。関数は、実体に対して、定義域内のある値または関連する実体または集合を与える。

関係型スキーマから、関係に実体型、属性に関数を設けることにより、Daplex スキーマを定義できる。網型スキーマから、レコード型に実体型、データ項目に関数、親子型に、親実体の子実体集合を与える多値関数を設けることにより Daplex スキーマを定義できる。図-4 に対する Daplex スキーマを図-6 に示す。DE は多値関数であり、dname は関数である。

共通のデータ言語を備えたローカル概念モデルから全体モデルを定義するには、データの意味的な相違を解決する必要がある。このためには、定義域の異なる類似した実体型の上位に、これらを一般化した型を定義することと、実体型を集めた集積物としての上位の型を定義する方法がある。前者は isa 階層であり、後者は partof 階層である³⁶⁾。Daplex モデルでは、isa 階層が提供されている。たとえば、図-6 の Emp と Author に対して、これらを一般化した実体型 Person

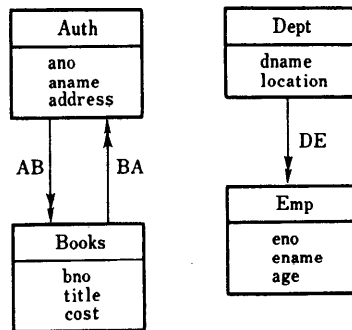


図-6 Daplex スキーマ

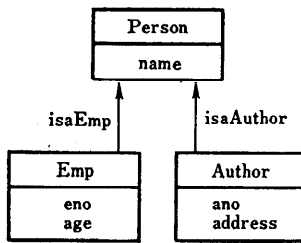


図-7 isa 階層

を定義できる。上位型の性質は、下位型に継承する。

D. 論理

データ構造を、第一階述語論理³⁷⁾の言語Lの論理式で表現し、Lのモデルとしてデータベースを考える演繹的データベースシステム¹⁴⁾がある。インテグリティ制約は公理として表現される。たとえば、関係 Emp (eno, ename, age) の組 (1, A, 35) は、述語 Emp (1, A, 35) と書ける。Emp で属性 eno が主キーであるという制約は、以下の論理式で書ける。

$$\forall N \forall M \forall A \forall M1 \forall A1 (Emp (N, M, A) \& Emp (N, M1, A1) \rightarrow N=N1 \& M=M1)$$

種々のデータ構造に対する公理は、文献17)に示されている。推論規則として分離規則がある。第一階言語、公理集合、推論規則からなる体系を第一階理論といい、健全で完全なことが知られている³⁸⁾。

論理式をホーン節に制限し、SLD 導出¹⁷⁾を推論規則とした体系が Prolog システムである。データベースを具象単位節、視野を規則節として与えることにより、どのデータベースシステムも同一に扱える。また、視野を再帰的に定義でき、データ構造と操作を同一に扱える。関係型モデルとの関係は文献 14), 18), 33) など、網型モデルとの関係は文献 29) で研究されている。

5. 操作演算変換

操作演算変換は、4. で述べたローカル概念モデルの操作演算を、ローカルモデルの操作演算に変換することである。

A. 関係型操作演算から網型操作演算への変換

JDDBS では、共通型モデルとして、二項関係データ関係性を示す関係型モデルによる表現を用いている。図-4(b)に対する関係型スキーマを以下に示す。

Dept (@D, dname)
Emp (@E, ename, age)
DE (@E, @D)

ここで、@の付いた属性は、データベースキーを値としてとる属性である。

SQL と DML とを比較してみると、データ構造と操作単位の二点の相違がある。このために、網型データ構造に対する非手続的な中間言語 CQL を導入し、SQL を CQL 演算に変換する。たとえば、「Aと同じ部の部員を求めよ」という SQL 問い合わせを考える。

```
select e. ename
from Emp e, Dept d, DE de, Emp el, DE del
where e. eno=d. eno and de. dno=d. dno and
      d. dno=del. dno and del. eno=el. eno and
      el. ename='A';
```

これは、図-8のようにCQLに変換される。節点はレコード実現値を指示する変数、二つの節点間の矢は二つの変数の指示するレコードで実現値が親子関係にあることを示している。このCQLグラフを縦型に探索して、レコード型の実現値数、親子型の平均の子の数といった統計情報に基づいてアクセスされるレコード実現値の期待数を最小とするようにアクセスパスが決まり、DMLプログラムが合成される。操作演算変換の負荷については、文献28)にまとめてある。

B. Daplex 操作演算からの変換

Daplex では、実体単位のループにより質問が表現される。たとえば、図-6(b)に対する質問「30歳以上の開発部員を求めよ」は、Daplexにより次のように書ける。Daplexでは、実体に対して変数を一つだけもてる。

```
for each d in Dept where dname(d) = "開発"
  loop for each e in DE(d) where age(e) >= 30
    loop print (name(e)) end loop
  end loop
```

Daplex プログラムから、以下のSQLが直接的に求められる。

```
select ename from Emp where dno in
(select dno from Dept where dname = "開発")
```

網型データベースシステムに対してもJDDBSと同様な手続きによりDMLが合成される。アクセスコストとしては、ページI/Oの数、結果を格納するファイルのサイズ、後処理コストがある¹¹⁾。

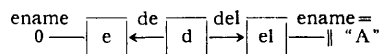


図-8 CQL

6. 例

異種データベースシステムから構成される分散型データベースシステムの代表例について紹介する。

A. Multibase

Multibase は CCA 社により開発された統合型システムである。Multibase のデータモデル階層は、図-1 と同様な階層構造で、下位からローカルホストスキーマ層、Daplex ローカルスキーマ層、Daplex 全体スキーマ層という。ローカル層と全体層には、Daplex という関数型のデータモデルが用いられている。

B. DDTS

DDTS はハネウェルで開発された統合型システムであり、ANSI の三階層構造³²⁾を基本にしている。まず各データモデルを関係型で共通記述してから、ER モデルにより異種性を解決した概念層が定義される。

C. JDDBS

JDDBS は、日本情報処理開発協会で開発された統合型システムである。図-1 の階層構造をもち、共通モデルとして関係型モデルが用いられている。

D. 多データベースシステム

連邦型の例として、多データベースシステム²²⁾がある。たとえば、Michelin と Gault-M データベースシステムが同名の料理店についての関係 R をもち、中華料理店を見付けたいとき、以下のように書ける。

```
open Michelin Gault_M er
-range (x R)
-select x
-where (x. type="Chinese")
retrieve
```

さらに、copy, transfer といったデータベースシステム間の操作演算が提供されている。

7. おわりに

異種のデータベースシステムからボトムアップ的に構成される分散型データベースシステムについて述べた。データモデル間のデータ構造と操作演算の変換を行えるようになってるのが現状である。今後は、共通型のモデルとして、より意味表現のできるデータモデルとこれにより全体モデルをいかに設計していくかの研究が必要である。また、データベースシステムのインテグリティを保つための同時実行制御、信頼性制御などの相違を考えていく必要がある。また、安全性の制御では、ワークステーションに導き

れたデータの情報流の制御が必要になる。

参考文献

- 1) Adiba, M. et al.: An Overview of the Polypheme Distributed Management System, Proc. of the IFIP '80, pp. 475-479 (1980).
- 2) Borr, A.: Transaction Monitoring in Encompass, Proc. of the 7th VLDB (1981).
- 3) Cardenas, A.F.: Heterogeneous Distributed Database Management: HD-DBMS, Proc. of the IEEE, Vol. 75, No. 5, pp. 588-600 (1987).
- 4) Chan, A., Dayal, U. and Fox, S.: An Ada-Compatible Database Management System, Proc. of the IEEE, Vol. 75, No. 5, pp. 674-694 (1987).
- 5) Chen, P.P.: The Entity-relationship Model—Toward a Unified View of Data, ACM TODS, Vol. 1, No. 1 (1976).
- 6) Chu, W.W. and Harverty, J.P.: Design Considerations for Shared Integrated Database Systems (SIDS), Computer Science Press (1983).
- 7) Codd, E.F.: Relational Database: A Practical Foundation for Productivity, CACM, Vol. 25, No. 2 (1982).
- 8) Codasyl: Report of the CODASYL Data Definition Language Committee, Journal of Development (1973).
- 9) Date, C.J.: Introduction to Database Systems, Addison-Wesley (1981).
- 10) Dayal, U. and Hwang, H.: View Definition and Generalization for Database Integration in a Multidatabase System, IEEE Trans. Software Eng., Vol. SE-10, No. 6, pp. 628-644 (1984).
- 11) Dayal, U.: Query Processing in a Multidatabase System, Conceptual Modeling, Springer-Verlag, pp. 81-108 (1985).
- 12) Dwyer, P.A. and Larson, J.A.: Some Experience with a Distributed Database Testbed System, Proc. of the IEEE, Vol. 75, No. 5, pp. 633-648 (1987).
- 13) Ferrier, A. and Stangret, C.: Heterogeneity in the Distributed Database Management System Sirius-Delta, Proc. VLDB 82 (1982).
- 14) Gallaire, H. et al.: Logic and Databases: A Deductive Approach, ACM Computing Survey, Vol. 16, No. 2 (1984).
- 15) Heimbigner, D. and McLeod, D.: A Federated Architecture for Information Management, ACM TOIS, Vol. 3, No. 3, pp. 253-278 (1985).
- 16) Kobayashi, I.: Classification and Transformations of Binary Relationship Schemata, Suuno Institute of Business Administration (1985).

- 17) Kowalski, R.: Logic for Problem Solving, North-Holland (1979).
- 18) Henschen, L. J. and Naqvi, S. A.: On Compiling Queries in Recursive First-Order Databases, JACM, Vol. 31, pp. 47-85 (1984).
- 19) Landers, T. and Rosenberg, R. L.: An Overview of Multibase, North-Holland (1982).
- 20) Lindsay, B. G.: A Retrospective of R*: A Distributed Database Management System, Proc. of the IEEE, Vol. 75, No. 5, pp. 668-673 (1987).
- 21) Lyngbaek, P. and McLeod, D.: An Approach to Object Sharing in Distributed Database Systems, Proc. VLDB 83, pp. 364-376 (1983).
- 22) Litwin, W. and Abdellatif, A.: An Overview of the Multi-Database Manipulation Language MDL, Proc. of the IEEE, Vol. 75, No. 5, pp. 621-632 (1987).
- 23) Masunaga, Y.: 米国における最近の分散型関係データベースシステム技術, 情報処理, Vol. 5, pp. 443-450 (1984).
- 24) Masunaga, Y.: 分散型データベースシステム, 情報処理, Vol. 28, No. 4, pp. 446-445 (1987).
- 25) Smith, J. M. et al.: Multibase-Integrating Heterogeneous Distributed Database Systems, AFIPS Conf. Proc. pp. 487-499 (1981).
- 26) Stonebraker, M. and Neuhold, E.: Distributed Data Base Version of INGRES, Proc. of the 2nd Berkeley Workshop, pp. 19-36 (1977).
- 27) Takizawa, M.: The Four-schema Concept as Gross Architecture of Distributed Databases and Heterogeneity Problems, Journal of Information Processing, Vol. 2, No. 3, pp. 134-142 (1979).
- 28) Takizawa, M.: Distributed Database System-JDDBS, JARECT Vol. 7, Computer Science and Technologies (Kitagawa, T. ed.), Ohmsha and North-Holland, pp. 262-283 (1983).
- 29) Takizawa, M., Itoh, H. and Moriya, K.: Logic Interface System on Navigational Database System, Lecture Notes in Computer Science, Springer-Verlag, pp. 70-80 (1987).
- 30) Takizawa, M.: Design of Highly Reliable Broadcast Communication Protocol, Proc. of the 11th IEEE COMPSAC, pp. 731-740 (1987).
- 31) Templeton, M. et al.: Mermaid-A Front-End to Distributed Heterogeneous Databases, Proc. of the IEEE, Vol. 75, No. 5, pp. 695-708 (1987).
- 32) Tschritzis, D. and Klug, A.: The ANSI/X3/SPARC DBMS Framework Report on the Study Group on Database Management Systems, Information Systems, Vol. 3, pp. 173-191 (1978).
- 33) Ullman, J. D.: Implementation of Logical Query Language for Databases, ACM TODS, Vol. 10, No. 3, pp. 289-321 (1985).
- 34) Olle, T.: The CODASYL Approach to Data Base Management, John Wiley and Sons (1978).
- 35) 植村: 大型プロジェクト「電子計算機相互運用データベースシステム」について, 情報処理学会「アドバンストデータベースシステム」シンポジウム (1985).
- 36) 上野: 知識工学入門, オーム社 (1986).
- 37) 野口, 滝沢: 知識工学基礎論, オーム社 (1986).
- 38) Enderton, H.: Mathematical Introduction to Logic, Academic Press (1972).

(昭和62年9月14日受付)