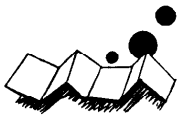
解　説

大規模データベースとその実現技術†



村上国男†† 森　道直††† 中野良平†††

1. はじめに

近年，データベースサービスの発展に伴い，データベースの大規模化・広域分散化・高水準化・マルチメディア化などの動向が顕著になった．データベースは本質的にスケールメリットを追求するものであり，一方に，パーソナルデータベースの発展を促しながら，データベースの大規模化傾向は今後も一層著しくなっていくものと思われる．特に，システム規模が全国的な広がりを持ち，扱うデータ量も膨大な国家的あるいは大企業のデータベースでは，100 G バイトを越えるものが出現し始めている．

データベース大規模化のインパクトは，データベース技術全体に及び，現行のデータベース規模*(約20～30 G バイト以下) では現われなかった，あるいは比較的容易に解決された問題点が一度に顕在化することとなる．たとえば，大容量・高速な2次記憶装置，膨大なトラヒックを処理するシステム構成技術，多様なアプリケーションを実現するデータベース設計と業務プログラムの開発，迅速な障害回復，高速な再編成／再構成，情報および構造に関するインテグリティ，複雑なシステムの性能評価技術，データベース管理者，多様なユーザに対する情報保護を行うセキュリティ大規模データベースの作成技術などがあげられる．

本稿では，これらのインパクトを，特にシステム開発の面からとらえ，大規模データベースシステムの構成に関する技術と開発・運用に関する技術の両分野に絞って，主要な課題について概要を述べることとしたい．

2. 大規模データベースの動向

大型計算機設置数は前年比で約 10% 伸びており[2]，また大型システムにおけるデータベース導入比率は 1980 年では約 50% 程度であるが，1990 年には 90% に達するであろうと予想されている[3]．このような大型システムにおける急速なデータベースの普及は，大規模なデータベースを増加させることとなろう．

次に，データベース規模の増大とともにシステム処理能力も増強されねばならないことが予想される．両者の関係を，電電公社が開発している大規模なデータ通信システムを例にとって図-1 に示す．これによると両者はほぼ比例して増大する傾向が見られ，100 G バイトを越えるデータベースが出現し始めたことが注目される．

今後，文書・画像などの非コード化情報の取扱いが大きな比重を占めるようになると，現在のコード化情報のデータベース規模はさらに1桁程度[4]巨大化するといわれている．

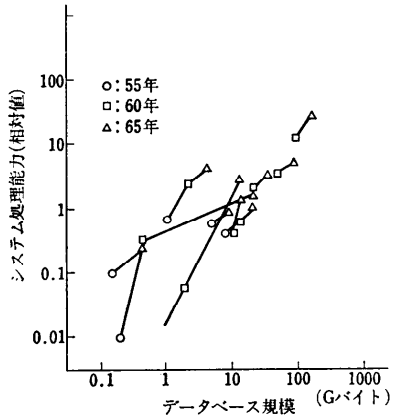


図-1　公社データ通信システムにおけるデータベースの大規模化の傾向

† Data Base Technology for Very Large Data Bases by Kunio MURAKAMI (Institute for New Generation Computer Technology) and Michinao MORI, Ryohei NAKANO (Nippon Telegraph and Telephone Public Corporation Yokosuka Electrical Communication Laboratory).

†† (財)新世代コンピュータ技術開発機構研究所
††† 日本電信電話公社横須賀電気通信研究所

* 本稿では，大規模データベースとは再編成／再構成処理が許容される運用時間（たとえば，週末における 64 時間）内に完了しないものという定義[1]に従い，約 20～30Gバイト以上を想定する．

3. システム構成に関する技術

大規模データベースを実現するシステム構成に関する技術として，大規模データベースを格納・蓄積するハードウェア技術と，大規模システムを分散形システムとして構成するシステム化技術を取り上げて概要を述べることとする．

3.1 ハードウェア技術

データを格納する記憶システムは，価格（ビット当り）と性能（アクセス時間）を考慮して階層構成を採用しており，図-2 に示すように各々の階層で低価格化・高速化が進んでいる[5]．一般にデータベースは，大容量でブロック単位のランダムアクセス性が必要とされるため，それを格納する2次記憶装置としては，大規模データベースにおいても磁気ディスク装置が主流である．この傾向は，連続薄膜媒体・薄膜ヘッド等の技術開発により，当分の間続くといわれている．

磁気ディスク装置の記録密度は10年間に約10倍向上[6]しているが，アクセス速度はほとんど改善されていないため，単位情報当りのアクセス耐力が低下することとなり，高トラヒックのデータベースの場合はシステム全体のボトルネックになることが考えられる．これに対して，高速ファイル装置またはディスクキャッシュが提案され実用化が進められている．前者としては，固定ヘッドディスク・磁気バブル・CCD・MOS メモリなどがあるが，揮発性という短所はあるが半導体技術の進歩によるビット当りのコスト低下を考慮すると MOS メモリファイル装置が有力と思われる．後者は，主記憶と磁気ディスク間に存在するアクセスギャップ（アクセス時間比：10^5 程度）を緩和する目的を持ち，製品化され始めているが，ファイルアクセスの局所性・プログラムの多重度などの複雑なシステム特性と関連付けたより定量的な評価が課題である．いずれの技術についても今後の計算機システムの記憶階層構成に影響を与えていくものと考えられる．

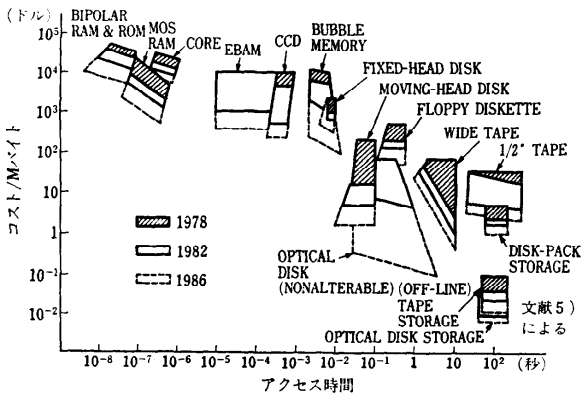


図-2 計算機システムにおける記憶階層構成の推移

数 10〜数 100 G バイトのファイル空間を構成する超大容量の記憶装置の必要性は，データベースの規模拡大とともに増してきており，代表的なものに MSS があるが，装置価格が高く，シーケンシャルアクセス速度が遅いなどの理由から，当初の予測より普及の速度が遅い．上記に述べた磁気ディスク装置の適用領域の拡大により，MSS は今後1桁以上の価格性能比の改善が望まれる技術であろう．また，光ディスクは磁気ディスクより1桁高い記録密度を実現しており，直径 30 cm 程度の光ディスク片面で 10〜100 G ビットの記録が可能であり，ランダムアクセス可能な大容量メモリとして期待されている．現状では，書き換え不可な DRAW (direct read after write) 方式を中心として開発が進んでいるが，ビット誤り率が 10^{-3}〜10^{-5} 程度であるため，当面はそれほど高品質の必要がない文書ファイルの用途が主である．

3.2 システム化技術

大型計算機の処理性能は，年率約 17% の割合で向上している．また米国における大規模ユーザを対象とした調査結果によると，システム処理能力の増加率は 1970 年代後半では 30〜40% 程度であったものが，1980 年代には 50% を越えると予想されている[7]．これは，データベースを利用するオンライン適用業務の急速な成長の結果であると分析されている．このような急激なシステム処理能力の増大要求に対して，単一プロセッサでシステムを構成する案は処理性能の限界および拡張性の無さなどの理由により，採り得ない．したがって，複数のプロセッサで大規模システムを構成する分散処理技術が必須となり，今後のシステム化に関する中核技術となっていくであろう．

分散処理システムは，機能分散形，負荷分散形および分散形ネットワークの3種類に大分類できる[8]．大規模データベースシステムは，これらの形態を複合するシステム構成で構築されることになろう．ここでは機能分散形と分散形ネットワークを複合するシステム構成のバックエンドアプローチについて説明する．

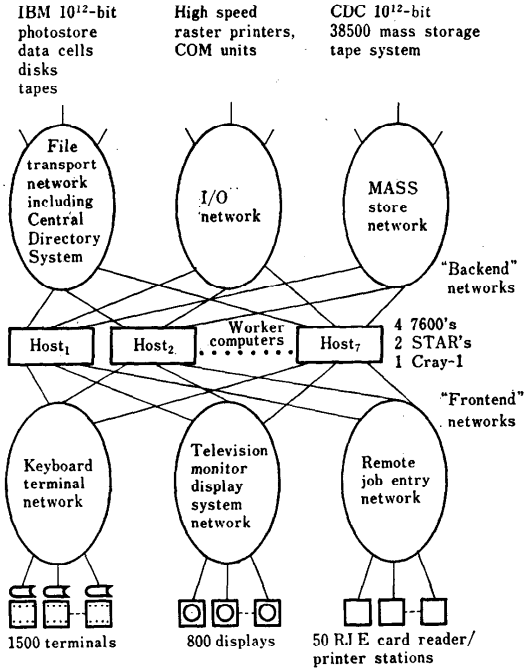


図-3 Octopus のネットワーク構造

バックエンドアプローチはベル研の XDMS を最初として，それ以降多くのバックエンドシステムが研究開発されている．その主要技術はデータベース処理の分散制御技術（分散データベース技術），プロセッサ間通信技術およびバックエンドプロセッサの専用化技術（データベースマシン技術）に分類できる．

1つのアプローチは，バックエンド記憶ネットワーク[9] (Backend Storage Network) と呼ばれるものである．BSN は，複数の異機種プロセッサが MSS・磁気ディスク装置などに格納されたデータベースを共用するもので，超大容量の記憶サブシステムを提供するものである．BSN の代表的なシステムである Octopus を図-3 に示す．BSN においては，大量かつ高速なアクセスが必要とされるため，数 M ビット/秒以上の転送速度と交換機能を提供するローカルネットワーク技術[10] が利用される．また，BSN はデータベースマシンを有効に接続するものとなって行くであろう．

もう1つのアプローチは，現在電電公社で開発中のバックエンドデータベースシステムであり，ホストプロセッサとバックエンドプロセッサはチャネルまたは回線で結合され．広域な分散処理システムが構築できる(図-4)．本システムは，その開発に先立ち，プロトタイプの機能分散システム[11]を構築して主要な技術に関する確認を行っており，高速かつ高水準のプロセッサ間通信法[12]などを実現している．

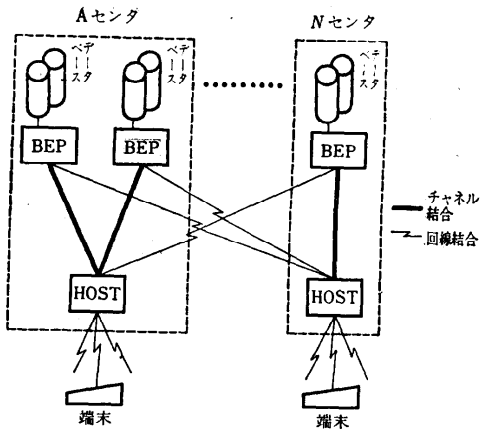


図-4 バックエンドデータベースシステム構成例

4. システム開発・運用に関する技術

4.1 設計技術

4.1.1 大規模データベースとデータモデル

大規模データベースの形態として，集中形と分散形を考え，各々におけるデータモデルの話題を述べる．

（1） 集中形大規模データベース

1センタに集中した形態の大規模データベースについては，CODASYL DBTG の提示したネットワークモデルは実績がある[13]．一方，理解の容易さ，データ独立性，理論的基礎などの利点を持つリレーショナルモデルの適用性はどうであろうか．最大の問題は性能にある．SYSTEM-R の性能評価結果[14] によれば，インデックスを用いれば性能 (CPU 時間，I/O 回数) はデータベース規模に依存しないという．しかし，対象としたデータベースは 2～38 M バイトの規模であり，数 10 G バイトになったときどうなるか即断できない．現実に使用されている範囲などからも推測して，汎用マシンの上に構築するリレーショナルモデルは数 10 M バイト位までしか適用性が確認されていないといえよう．

（2） 分散形大規模データベース

データベースが網内の各所に配置され，網全体として大規模なデータベースを共有する形態の分散形大規模データベースでは，集中形とは違った観点が重要になる．網内には，ネットワーク，リレーショナル，または階層などの各種のデータモデルのデータベースが

混在している．したがって，そうしたデータベースを利用者にどのように見せるかという問題が重要になる．即ち，異種データモデルを扱う統一データモデル[15]・言語とか共存アーキテクチャの研究が必須となる．データベースレベルのネットワークプロトコルの研究もこれらと密接に関連する．電電公社ではDCNAの開発の一環として，データベースビューおよび基本アクセスの仕様を研究し，規定している[16]．

4.1.2 データベース設計

データベースが登場した頃は，データベース規模も小さくアプリケーションも限られていたため，主に物理設計に研究の重点が置かれていた．最近ではデータベース規模の増大，アプリケーションの多様化などにより，データベース設計は極めて複雑かつ困難な作業となり，これに伴って研究も多様な広がりを持ってきた．1978年に斯界の第一線の研究者がNew Orleansに集まってworkshopを開いた．その報告書[17]はデータベース設計の実状と課題を適確に示していると思われるので，それをもとにこの分野の概況を述べる．データベース設計を要求分析，情報分析，論理設計，物理設計の4領域に分ける．

（1） 要求分析 (corporate requirement analysis)

要求分析とは，データベースシステムに関与する様様の人々からデータベース設計に必要な情報を収集することである．この領域での研究課題はデータモデルおよび方法論・ツールにある．要求分析のためのデータモデルは多様なエンドユーザが理解できる，簡単で自然なものでなければならない．人間の介在が期待できるので，少々数学的基盤に欠けるモデルであっても構わない．また方法論・ツールはデータベース設計，プロセス設計の双方を支援するものでなければならない．ソフトウェア工学の分野で開発されたものは一般的すぎるが，プロセス指向が強すぎるため，ほとんどが適用できない．データディクショナリシステムは改良されており，徐々に適用可能になりつつある．

（2） 情報分析 (Information Analysis and Definition)

情報分析は，ビューのモデル化とビューの分析・統合から成る．ビューのモデル化とは要求分析の結果をもとにビューを定式化することであり，ビューの分析・統合とはモデル化された多くのビューを分析して，1個のビューにまとめる作業である．これらの作業の中で概念モデル (conceptual model) が作成される．

以下に，この領域の主な研究課題を示す．

（a） 要求記述言語の研究：要求分析の結果を定式化する言語としては，PSL/PSA, CASCADE, CADISなどがあるが，完璧性・表現力・操作性に関して更に研究が必要である．

（b） ビューの統合手順の研究：ビューが統合可能か否かの判定基準，ビューの統合アルゴリズム，ビュー統合における衝突の型の分類と解決法などに関する研究が必要である．

（3） 論理設計 (logical database design)

論理設計とは，性能・完全性 (integrity)・整合性 (consistency)・障害回復などに関する制約を満たす，DBMS で処理可能な論理スキーマを作ることである．この領域で使用できるツールは，評価ツールと設計ツールに大別できる．論理設計の大半は依然として工芸 (art) の段階であり，設計者のスキルに大きく依存しているのが実状である．この領域をすべて自動化できるか否かは明確でない．今後の課題は，ツール群を多様な環境で順次使用可能なようにすること（ツールの統合）にある．

（4） 物理設計 (physical database design)

物理設計とは，論理スキーマ，DBMS 特性，プロセス特性などをもとに物理スキーマを設計し，その性能を評価することである．設計目的および性能尺度が明確に規定できるので，問題の定式化は他の領域に比べるとずっと容易であり，多くの問題に対して解が得られている．今後の課題は他の領域との自然な接続，単発ツールの統合，解析モデルの拡張（複数ファイルでの解析）などにある．

4.1.3 業務プログラムの設計

データベース規模が増大すれば，アプリケーションも多様化し，業務プログラム (application program) の規模も大きくなる．静的ステップ数で1Mから数Mに及ぶ大規模プログラムの開発・保守をどう進めるかはシステム総経費の増減に多大な影響を与える．プログラムの開発・保守において最も重要な点は環境 (environment) にある．1980年にソフトウェア開発環境に関するworkshopが米国標準局の後援で開催された．その中でHowdenを議長とする環境グループが提案した考え方[18]をもとに，大規模プログラムの開発環境について考えてみたい．

現在，ソフトウェア開発ツールの数は400を越えるといわれるが，それらは要求仕様，設計，コード化，検証，および管理のいずれかの分野に分類できる．大規模プログラムの開発環境として，必要最小限と本格的

の2レベルを考える．必要最小限の環境は 33 種類の技法またはツールを含み，購入費用は約 30 万ドル，本格的環境は 46 種類で約 300 万ドルと推定される．

プログラムの開発環境の改善において，ソフトウェア工学データベース (software engineering database) は中心的役割を果たす．プログラムの開発過程で各種ツールを次から次へと使用することを考えると，ツールの統合が問題となる．ツール間を直接接続するのはインタフェース整合に多大の労力を要するため好ましくない．Unix OS の例に見られるように，すぐれたツール総合体はデータを共通基盤にしていることを考慮すれば，ソフトウェア工学データベースを中心にして統合化を考えるべきである．即ち，各々のツールはデータベースを中核とし，そこから情報を入力し，結果をそこへ出力し，情報として蓄積する．なお，ツール標準化の問題も，機能の標準化でなく，ソフトウェア工学データベースの標準化の方向で考えればツール開発は円滑に行われるようになるであろう．

日本の現状は Howden グループの提示した環境にはほど遠いように思われる．しかし，今後は大規模ソフトウェアを開発する教訓の中から，システム全サイクルにわたった支援体系が徐々に構築されて行くであろう．

4.2 運用技術

データベースシステムの運用は多岐にわたるが，本稿では障害回復と再編成／再構成の話題に絞る．

4.2.1 障害回復

データベースシステムの障害回復技法は Verhofstad[19] によれば，①救助プログラム，②変分ダンプ，③追跡控え，④微分ファイル，⑤バックアップ版／現在版，⑥複数の写し，⑦慎重な置換えに分類される．オンラインデータベースシステムでは通常①，②，③，⑤の技法を中心に障害処理を構築している．救助プログラムを除く3技法は障害事前処理として日々の運用に密接に関係するので，以下，データベース規模の増大との関連を探ってみる．

データベース規模が大きくなれば，そこへのトランザクションのトラヒックも増大する傾向にある．それに伴って，追跡控え（ジャーナル）の量も増加する．現在，追跡控えは磁気テープに取得するのが普通であるが，1日のジャーナル量が 60〜70 本にもなると，その交換間隔は 10 分位になりオペレータの負担が多大になるため，大容量2次記憶装置（磁気ディスク装置，MSS など）の利用または磁気テープ操作の完全自動化などの対処が必要となる．

次に，データベース規模とバックアップ版の大きさの関連を考える．たとえば，100 G バイトのデータベースの場合には，バックアップ用磁気テープは約 1,000 本必要になる（複数世代および正・副・予備を考えると n 倍になる）．これに伴い，データベースの全ダンプ処理時間は長くなり，たとえば，100 G バイトのデータベースでは 100 時間を越え，週末に行うこともできない．これに対しては，次のような手法がある．

（1） 分割全ダンプ方式：データベースを幾つかの群に分けて，群単位にサイクリックに全ダンプを取得する．この方式は幾つかのシステムで実際に採用されている．

（2） 全ダンプ保守方式：データベースを幾つかの群に分けておき，長期のサービス中断期間（年末年始など）に全ダンプを取得する．それ以降は毎日，データベースの更新分に関する追跡控えを編集して，群ごとに更新データを累積作成しておく．こうして累積作成された更新データを群ごとにサイクリックに全ダンプに反映してバックアップ版を保守する．

なお，2次記憶装置に障害が発生した場合の回復は全ダンプに追跡控えを重ねて行き，障害直前の状態を再現する．400 M バイトの磁気ディスク装置で，バックアップ版のサイクリックな作成（または保守）の周期が 20 日の場合の回復時間は，1日の更新件数が装置当り 5,000 件以下のときには，約 40〜50 分と推定され，ほぼ許容される範囲となる．このように群分割されたバックアップ版が用意されていれば，2次記憶装置障害に対する回復は大規模データベースにおいても特に問題はない．

4.2.2 再編成／再構成

データベース規模の増大はデータベースの再編成／再構成に深刻な影響を与える．それは処理時間と費用の多大な増加を意味するからである．再編成／再構成は以下のような状況で必要となる[20]．

a）オーバフローの発生，削除レコードの累積により性能が劣化した．

b）対象とする情報構造が法律改正，組織整備などの理由により変化した．

c）最適なアクセス条件，空間条件，が時とともに変化した．

d）以前使われていたデータベースを統合して新データベースに変換する必要が生じた．

再編成／再構成の方式は次の3種類に大別できる．

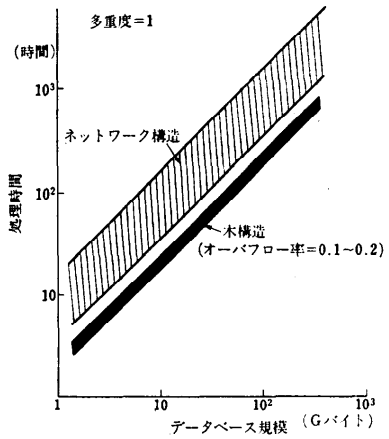


図-5 オフライン再編成／再構成処理時間（推定）

（1） オフライン再編成／再構成：通常のアクセスがないオフライン環境で再編成／再構成を実施する．

（2） 逐次 (incremental) 再編成：データベースに対する通常アクセスを契機に関連する部分の再編成を行う．オフライン処理は減るが，通常アクセスの性能が劣化することになる．

（3） 動的 (dynamic) 再編成／再構成：通常アクセスと並行して再編成／再構成を行う．

現在実用化されているのは (2) までの方式で，動的再編成／再構成は研究段階にある．広く採用されているオフライン再編成／再構成に必要な処理時間を推定して図-5 に示す．100 G バイトのデータベースの場合，約 200～1,000 時間必要である．I/O バウンダリの場合には，処理の多重化により処理時間の削減が可能であるが，それも CPU またはチャネルのネックにより限界がある．このようにオフライン方式は大規模データベースにとって耐えられないものとなってきている．これに対して，現実のシステムでは再編成／再構成の局所化などの対処をしているが，抜本的解決ではなく，動的再編成／再構成の技術開発が急務とされている．この技術は 24 時間運用が要求される軍，警察，医療関係のシステムにとっては必須のものである．これに関する研究動向は文献 20) を参照されたい．

5. むすび

大規模データベースについて，システムの構成・開発・運用の面から，技術動向および課題の概要を説明したが，本稿で述べなかった項目たとえばデータベースの変換，インテグリティの維持，セキュリティの向上などについても課題は多い．これらについては他の文献[21),22)] を参照されたい．

参考文献

1) Gerritsen, R., et al.: On Some Metrics for Databases or What is a Very Large Database?, ACM SIGMOD, Vol. 9, No. 1, pp. 50-74 (1977).
2) 日本情報処理開発協会編：コンピュータ白書 (1981年11月).
3) Creative Strategies International: Data Base Management Systems (Jan. 1980).
4) 草鹿，荻原：情報の蓄積・検索技術，情報処理，Vol. 22, No. 10, pp. 979-991 (1981).
5) Chi, C. S.: Advances in Computer Mass Storage Technology, Computer, Vol. 15, No. 5, pp. 60-74 (1982).
6) 金子，吉井：3・2 ギガバイト集合形磁気ディスク記憶の実用化，通研実報，Vol. 31, No. 1, pp. 241-247 (1982).
7) 藤田：IBM システム/370 の拡張アーキテクチャと MVS/XA，日経コンピュータ，pp. 65-84 (1982. 2. 22).
8) 関野：分散処理技術，情報処理，Vol. 20, No. 4, pp. 275-283 (1979).
9) Thornton, J. E.: Backend Network Approaches, COMPCON 80, pp. 217-223 (Mar. 1980).
10) Hsi, P. and Lissack, T.: Local Networks' Consensus: High Speed, Data Communications, Vol. 9, No. 12, pp. 56-66 (Dec. 1980).
11) 村上，森：FCP 機能分散方式の実用化，通研実報，Vol. 31, No. 2 pp. 349-358 (1982).
12) 中野，森：疎結合計算機システムにおける高速計算機間通信方式，第 32 回計算機アーキテクチャ研究会，pp. 23-34 (1981).
13) Michales, A. S., et al.: A Comparison of the Relational and CODASYL Approaches to Data-Base Management, Comput. Surv., Vol. 8, No. 1, pp. 125-151 (Mar. 1976).
14) Chamberlin, D. D., et al.: Support for Repetitive Transactions and Ad Hoc Queries in System R, ACM Trans. Database Syst., Vol. 16, No. 1, pp. 70-94 (Mar. 1981).
15) Date, C. J.: An Introduction to the Unified Database Language (UDL), Proc. 6th Int. Conf. on VLDB, pp. 15-29 (1980).
16) 河津，柴崎，南，大沼：DCNA のデータベースアクセスプロトコル，通研実報，Vol. 30, No. 3, pp. 799-823 (1981).
17) Lum, V. Y., et al.: 1978 New Orleans Data Base Design Workshop Report, IBM Research Report RJ 2554 (33154) (1979).
18) Howden, W. E.: Contemporary Software Development Environments, Comm. ACM,

Vol. 25, No. 5, pp. 318-329 (1982).
19) Verhofstad, J. S. M.: Recovery Techniques for Database Systems, Comput. Surv., Vol. 10, No. 2, pp. 167-195 (June 1978).
20) Sockut, G. H. and Goldberg, R. P.: Database Reorganization—Principles and Practice, Comput. Surv., Vol. 11, No. 4, pp. 371-395 (Dec. 1979).
21) Berg, J. L. (Ed.): Data Base Directions II: The Conversion Problem, Data Base & SIGMOD Record ACM, Vol. 12, No. 2 (Jan. 1982).
22) Denning, D. E. and Denning, P. J.: Data Security, Comput. Surv., Vol. 11, No. 3 (Sep. 1979).

(昭和57年7月12日受付)