

# データベース研究

## — 21世紀に向けての挑戦 —

上林弥彦

京都大学工学部 高度情報開発実験施設

現在データベースの研究開発は、75年について、第2の変革期にあるといえる。とくに、大量データを用いる種々の応用プログラムをデータベースの上で容易に開発できるようなシステム、分散データベースの実現、オブジェクト指向などのソフト技術や大容量主記憶などのハード技術の利用といった課題が重要となっている。本稿では、21世紀の始めに来ると考えられる次の変革期にむけたデータベース研究について考察するとともに、日本のデータベース研究の問題点について述べる。

# Database Research

## - Challenges towards 21st Century -

Yahiko Kambayashi

IMEL Faculty of Engineering

Kyoto University

Sakyo Kyoto 606 Japan

yahiko@kuis.kyoto-u.ac.jp

The current database research faces the second turning point after the first one occurred around 1975 when relational database systems became practical and textbooks on databases were published. Realization of systems on which various application programs requiring large amount of data should be easily developed, realization of multidatabase systems, use of software technology such as object orientation and hardware technology such as large main memory to database systems, are major research topics. In this paper we will discuss research topics towards the third turning point which will come around the beginning of 21st century and problems of database research in Japan.

## 1. まえがき

現在のデータベース研究開発は1975年に次いで第二の大きな変革期にあるといえる。この時期に過去を振り返って、21世紀に向けたデータベース研究開発を考えることは有意義であろう。

データベースは1960年代に商用システムが開発され、1975年前後には大学での正式科目として取り入れられ、情報工学の学問としても認められるようになった。代表的な国際会議であるSIGMODやVLDBが開始され、DateやMartinの教科書が出版された。関係データベースシステムのプロトタイプが出来始め、構造型のデータベースとの優劣について激しく議論されたのもこの頃である。その後の10年間で関係データベースを中心としたデータベース理論が構築され、パソコンやワークステーションといった新しい環境のもとでは特に関係データベースが広まることとなった。

現在のデータベース研究は、この時以上の大きな変化を迎えつつある。この15年間に主記憶や二次記憶のコストは大幅に減少し、コンピュータ応用の分野で、大量のデータを扱う必要のあるものや、他のプログラムとデータを共有する必要のあるものが増えてきたためである。このため、コンピュータ利用の全ての方面にデータベース技術が拡散してきており、データベース向きのオペレーティングシステム、永続性のあるデータを扱うことのできるプログラミング言語、CADシステムとデータベースの統合、実時間制御システムへのデータベース利用等データベースと関連した研究開発が行われている。

これらの変化に伴い、関係データベース対オブジェクト指向(ないしは演繹)データベースといった優劣論が盛んであるが、これらの新しいデータベースは新しい応用分野により適したものとなり得るであろう。関係データベースでORACLE等のベンチャービジネスの商品化が先行したように、オブジェクト指向についても数社のベンチャービジネスが商品化を行っている。これらの新しいモデルは、オブジェクト指向プログラミングや人工知能といったソフトウェア上の技術進歩に大きく関係している。

データベース技術と進歩の著しいコンピュータハードウェア技術との係わり合いも非常に重要な問題である。もちろん、ソフトウェア技術はハードウェアの技術に支えられて変化してきたものではあるが、特に重要な課題は分散と超並列である。分散システムは今後の高度情報化社会の基盤技術であり、分散データベースの実用化はそのための必要不可欠な事項といえる。特に異種データ

ベースでは当初目標とした透明性(ひとつの統合データベースとして扱える)の実現は不可能で、ある程度現実と妥協した(全域的な一貫性を強く要求しない)システム構築の理論が必要である。超並列システムは、現在のスーパーコンピュータの速度の壁を破るものとして非常に期待されている。しかしデータベースにおけるデータの扱いでは、直列可能性という言葉に代表されるように、直列が不可避の部分があり、超並列向きのデータベースについてはまた新しい考え方が必要となるであろう。

すでにデータベースが使われているような用途でも、より高性能化をめざした研究が重要である。特に実時間データベースを中心とする高速処理のデータベースに対する需要は非常に多い。また扱うデータの多様化(マルチメディア、時間、不完全データ)に対する対処や、科学用データベース、統計データベース、安全性といった従来から扱われている問題についてもより効率の高い実現法が期待されている。

新しいデータベース応用は新しい需要を生み出すことになり、そしてその需要からのフィードバックがデータベースシステムや理論を改良する。このサイクルが安定するのにあと10年はかかるであろう。そして21世紀の始めには、その間のソフトウェア技術、ハードウェア技術の進歩や新しい需要を反映した新しいデータベースへの探求-第3の変革-が必要となるであろう。その時のハードウェアとソフトウェアはデータオリエンティドが主流になってくると想像している。

## 2. データモデルの展望

関係モデルのデータベース研究に与えた影響は非常に大きい。大きな特性は次の3つである。

(1)データ独立性:データ自体の特性と使い方を分離した。

(2)非手続き言語:データベース言語として非手続き的な表現のできるクラスの言語を導入した。

(3)効率良く計算できる意味制約:表の形のデータ表現で扱うことのできる意味制約として、従属性及び参照一貫性を考えその性質を明らかにした。

(1)は、関係データベースの使い易さや客観的な設計の実現に役立ち、データベース理論の構築を容易にしている。また、データベーススキーマを必要に応じて変化させるときに大幅なシステム変更を避けることができるため、システムを利用しながらスキーマ変更が行える。この結果データの寿命を長持ちさせることができる。また、利用者によるデータの扱いの違いを自由度の高いビューに

よって扱っている。Coddのひとつの大きな貢献は、関係完備な言語を定義したことである。この言語は、チューリングマシンの能力を持たない代わりに、非手続き的表現ができ使いやすいため、利用者言語として優れている。データベースを普通に使うにはほぼ充分の能力があり、この能力で不足する部分のみを親言語で補って応用プログラムを書くこととなる。この言語の他の特色は、若干の制限で等価性判定が可能であること、答がデータ量の多項式時間で求まることである。チューリングマシンの能力を持つ言語では、一般に等価性判定ができないこと(本当の意味の最適化ができない)や、プログラムが無限に動き続けることの判定ができないことを考えると、いつかは答の求まるようなものしか表現できない言語は非常に有用である。(3)の従属性や参照一貫性は、大きなオーバヘッドなしにデータが矛盾していないかどうかを調べることができるため、実際の意味制約である。

この関係モデルの言語や意味制約が以上のような特色を持つ反面、これで表現できない質問や意味制約の扱いは初期の時代から問題とされていた。主要な問題点は次の3つであった。

(a)多様な意味制約の扱い  
(b)部品展開問題の扱い  
(c)集合の扱い、さらに一般的なデータ構造の扱い  
例えば、(a)に関しては、Smith&Smith [TODS77]に汎化と集約の必要性が示されている。またEswaran et. al[Software Eng.76]には、一般的な意味制約表現のためトリガーを実現する方法の検討が示されている。(b)についても、INGRESの初期の論文に「自分より地位の上の人で自分より給料の少ない人のいるような従業員を求める」質問の扱いの例があり、QBEでもその機能を実現している。(b)や(c)については、上林、田中ら[VLDB77]にも扱いの考察があり、前者は一般にグラフ(全順序、半順序、非巡回、有向、無向等)を表現する二項関係で扱い、後者は集合識別子を用いて、集合や集合の集合といったものとその間の制約について扱っている。(c)については特に非正規表現と従属性の関係を示した牧之内[VLDB77]が有効である。しかし、これらの方法は問題別に対処するもので、より一般的なモデルを与えるものではなかった。

演繹データベースは主として(a)と(b)の扱いを動機として考えられたものである。言語の能力を上げることにより、データベース言語、データ表現、意味制約を全て統一的に扱えるという点で特色があるが、その反面非手続き性や多項式性といっ

た関係言語の良さも失う結果となっている。現在では、PROLOGの部分クラスであるDATALOGと多項式時間計算可能質問言語との関係(DATALOGが含まれる)についての研究等が行われている。また、このようなシステムでは特に最適化の問題(主記憶だけでなく二次記憶も考えた)がシステム実用上重要であるといえる。

オブジェクト指向データベースは、プログラミング言語側とデータベース側の双方からの必要性から開発された。プログラミング側では共有大量データを用いるプログラムを開発するために、関係を含む種々のデータ型やデータの永続性といった性質が必要となっていた。データベース側では、関係モデルのままでは、新しい応用プログラム開発に対する親言語の負担が重く、特に内部表現できない意味制約については、必要な度に応用プログラム内で繰り返し表現する必要があること、多様なデータ操作と適用できるデータの関係を明確化されていないこと等の問題があるためであった。

オブジェクト指向データベースの特性については、DOOD'89で発表された「オブジェクト指向データベースシステム宣言」(西尾、田中訳、bit90)によくまとめられている。この中には関係データベースの拡張で対処できるものと、オブジェクト指向独自のものが混在している。

オブジェクト指向独自のものは、クラス、カプセル化、クラス階層における継承といったものであろう。過去のデータベース応用がデータの読み、書き操作を基本としたものであったことに比べると、マルチメディアデータ等は非常に多くの操作がある。この操作をクラス階層で整理し使い易くした点は、特に使い易い利用者インターフェースの実現といった面で優れた点である。また、同時に使われるデータをクラスの形でまとめて一括して主記憶に読み込むことにより、データ操作の高速化が期待できる。

このためオブジェクト指向データベース処理の最適化はデータのクラスタリングが主流となっている。関係データベースは応用プログラムの性質を反映しないこと(データ独立)を特色としているため、特定のクラス階層に対応するクラスタリングによる効率向上は行っていない。

しかし、逆にオブジェクト指向データベースでは次のような問題点がある。

1)言語能力が高い(一般にチューリング機械の能力)ために、非手続き性や多項式計算性、等価性判定可能性といったデータベース言語の利点が失われてしまう。

II)関係データベースのビューのような多様な使い方を許すと、クラスタリングを利用した処理の最適化ができない。また、スキーマ変更操作は大変である。これは、データ独立性を保証していないため当然の結果といえる。

III)データベース操作の対象とするデータがひとつのクラス内のデータオブジェクトとしてまとまっていればよいが、応用プログラムによっては多くのクラスを参照することになり、その場合は性能はかなり悪くなる。

分散データベースシステムでも、当初いわれていた利点(局所処理が多いと非常に有利)はあるが、大域的処理が避けられないのでその場合の処理効率が非常に低下する点と、III)の問題は非常によく似ている。

これらの問題点は、プログラミング言語とデータベース言語の本質的な差異によって生じているもので、オブジェクト指向プログラミング言語ではこれらは問題とはならない。即ち、I)については考えなくてよい。データベースでは、多様な使い方(解釈)やスキーマ変更の扱いが重要であるが、プログラミング言語では例えばスキーマ変更はあまり起こらない。したがって、オブジェクト指向データベースの研究としては、複数のビューの実現やスキーマ変更の効率化が重要となっている。「オブジェクト指向データベース宣言」では、教条的にオブジェクト指向に固守したため、従来の関係データベースの利点を失わせる結果となった。この点に関して従来から、実際のデータベースについて活発に研究しているグループが出したのが「第三世代データベースシステム宣言」である(SIGMOD RECORD90)。主要なポイントは、関係データベースの特色を生かしたまま、オブジェクト指向や演繹データベース的な特色を追加していくものである。型、継承、メソッド定義、カプセル化、ID(主キーのないとき)、永続性、ルール(トリガーで実現)等の導入と共に、非手続き型言語との併用、更新可能なビューといった性質も持っている。

彼らの方式では、データ独立性にこだわっているため、オブジェクト指向におけるデータクラスタリングは陽に実現されていない。しかし、関係データベースにクラスタリング最適化も導入できるようにして、オブジェクト指向を利用者ビューとして実現するという方法も可能である。主記憶が大幅に増えるとその差もあまり問題とならない可能性がある。「ネットワークデータベース対関係データベース」と「関係データベース対オブジェクト指向データベース」との状況はかなり異

なっていて、すんなりオブジェクト指向が有利という状況ではない。前者の場合は、ネットワークデータベース上に関係データベースインターフェースを作ってもその機能は不十分なものになったが、後者の場合、オブジェクト指向データベースは関係データベース上で実現できる。また、前者の場合、ネットワークデータベースが汎用機を主な対象とし、初期の商用関係モデルは小型機を主な対象としたり、従来できなかったアドホックな質問を簡単に作れる等の特色があった。これに対し、後者は同じワークステーション上のシステムである。このため、関係データベースのベンダーも手を拱いてオブジェクト指向データベースを見ている訳ではない。最近の関係データベースシステムでは利用者が必要とすれば拡張機能を使えるようになっており、その主なものとして、トリガー機能、非正規関係、可変長データ型などがある。オブジェクト識別子に近いものもINGRES(リリース6.3)が実現しているといわれている。非手続き型言語のレベルの利用者が圧倒的に多いことと、今まで使っていたシステムのままでオブジェクト指向的使い方でもできるようにすることとで、オブジェクト指向システムへの移行をくい止めにかかると思われる。結果として、オブジェクト指向的なシステムが効率良く実現できるような機能追加を行った関係データベースシステムが商用化され、オブジェクト指向機能は、ハイパーカードのような使いやすい、かつ用途に応じたオブジェクト指向的利用者インターフェースとして実現されるであろう。また、データベースの応用分野と考えられるマルチメディアデータベースや設計データベースでは、バージョン管理、長時間トランザクション、共同作業といった要請がある。これらの機能はモデル独立に議論しうるものであるが、従来の関係データベースでは、扱いが困難であるか扱えないものである。最後に、オブジェクト指向と分散システムの類似性から、分散データベースに適している可能性があることをつけ加えておく。

従来からあるデータベースの問題のうち重要なものは次の3つである。

- a)実時間データベース、高可用性データベース
- b)科学用データベース
- c)データベースセキュリティ

実時間データベースは重要な課題で、演繹データベースやオブジェクト指向データベースとは明らかに異なる問題を対象としている。

b)の科学用データベースは、集められたデータから仮説を出したり、仮説に合うデータを選んだ

りする機能や、一部のデータを実験的に変更してその影響を調べたりする仮定実験をする機能、統計データベース機能、計算結果の図的表示機能等が重要である(遺伝子プロジェクト等)。これらが単なる機能の寄せ集めですむのか、新しいデータモデルが必要なのかは今のところはっきりしない。

システムセキュリティで昔から仮定されているものに\*性がある。これは、セキュリティレベルをデータと利用者に割り当て(例えば、0、1、2、3)あるレベルの利用者はそれ以下のレベルのデータしか見ることができない、それ以上のレベルのデータにしか書き込めないというものである。この性質でレベルの上のデータが下へ行くことを禁止している。しかし、このモデルでは上のレベルのデータに嘘を書き込むことが可能である。下のレベルに情報が漏れないようにするため、本来禁止されるべき操作が出来ることがある。\*性に替わる妥当な仮定やセキュリティ管理が効率良く行えるシステムモデルの研究は特に重要である。

不完全情報や曖昧さの扱いも重要であるが、曖昧な情報を正確に扱うためには全てのデータに対し曖昧さを矛盾無く定義するというおそらく実現できない問題と係わり合わなければならない。否定情報も演繹データベースの成果の一つであるが、処理可能性との関係では困難な問題といえる。

### 3.分散データベースの課題

分散データベースは、分散システム上のデータベースとして必要性の高いもので、高度情報化社会の基盤となるべきものといえる。大規模システムを作る実際的な方法であり、使用中にシステムの変更ができることや耐故障性、安全性といった点に特色がある。70年代の終わりから80年代の始めにかけて、同種および異種分散システムの研究が盛んにおこなわれた。80年代の終わりに再び盛んになったのは、より弱い統合をめざしたシステムで、とくに連合システムとかマルチデータベースシステムという名前でも呼ばれることも多い。

とくに、10年前に目標とした分散からの透明性の実現(集中型と同じように使うことができる)は、分散システムの利点と反する要求であったため、その部分をより弱い条件にして各システムの自律性を認めるというのが大きな目標となっている。

(1)分散システムは、データの局所性を利用してゐるため、全域的な意味制約が必要であれば効率が非常に低下する。

(2)並行処理の正しさは直列可能性で定義されるがこれも全域的な意味制約である。

また、この10年間のデータベース技術の進歩(ワークステーションの高機能化とクライアント・サーバーモデルの実現、オブジェクト指向等の新しいデータベース技術)や標準化動向も研究課題の変化に影響している。自律性は、集中と分散とのトレードオフという困難な問題に対応しており、現実的な解決策を探している最中といえる。

この方面の最近の研究テーマの参考として91年4月に京都で開催された「マルチデータベースシステムの相互運用性に関する国際ワークショップ」(IEEEと情報処理学会が主催)での発表論文テーマを示す。

データベースシステムの統合

知識ベースシステムの統合

統合のためのオブジェクト指向的方法

統合のための知能ベース的方法

統合技術 マルチデータベース言語

意味的な多様性 質問処理 データモデル

トランザクション処理 信頼性セキュリティ

等価性(名前、制約等)の扱い

分散データベース応用

この分野は、解決しなければならない問題が分かっているが、効率的な実現法を探しているという段階にあるといえる。また、標準化も進みつつある。具体的な問題点については、ACM Computing Surveysの90年9月号に特集があり、共立出版からbit別冊「コンピューター・サイエンス」として日本語訳が出版される予定があるので省略する。

なお、92年6月に横浜で開催されるDistributed Computing Systemsの国際会議の1つのテーマは分散データベースシステムである(論文締切は10月15日, Prof. Joseph Urban, Arizona State Univ.)。

### 4.データベースハードウェアの動向

関係データベースでは少ない基本演算数でデータベースが実現できるため、関係データベースの概念が導入されてからこれらの基本演算を実現するハードウェアを用いたデータベースマシンの研究開発が盛んになった。今後はオブジェクト指向等に適したマシンの開発が行われるのであるが、少ない基本演算数という利点を考えると、関係データベースマシンの機能拡張を主体として実現される可能性が高い。

過去において高水準言語マシン、データベースマシン、PROLOGマシン等が開発されてきたが商用システムとして成功した例は少ない。特にデータ

ベースマシンの問題点としては、次のような要素がある。

(1)従来のソフトウェア資産が生かせないことが多い。データベースソフトはかなり大きなものとなっており、データベース演算等をハードウェアで高速化しても、バックアップ、回復処理、安全性処管理、並行処理等ソフトウェアで処理する(ないしは新たにハードウェアを設計する)べき問題が多い。すでに蓄えたデータがそのまま新システムに移行できるようにするのは、非常に手間のかかる仕事となる。

(2)LSIの進歩が急速である。このため、設計に時間をかけるより少しでも新しいLSI技術を使う方がシステムの高速化に役立つ。例えば、同じハードウェア技術ではCISC(多くの命令を実現したコンピュータ)の方がRISC(縮小命令集合コンピュータ)より高速になるとしても、前者が設計に2年かかり後者が半年かかるとすれば、設計に使うLSI技術が一代違うことになる。その間にクロック周波数が1.5倍になれば、同時に製品化されたRISCの方が高速の演算を実現できる(実際にマイクロプロセッサの高速化への寄与は、アーキテクチャ上の工夫よりもクロック周波数の改善の方が遥かに大きいことが知られている)。データベース専用マシン等を設計すればその複雑さはCISC以上であり、汎用性から考えると有用とはいえない可能性が高い。RISCの有利さと同様データベースマシンを実現するならオブジェクト指向等ではなく最も単純な関係データベースマシンを選ぶべきであろう。

(3)ディスクへの入出力のボトルネックの対策を立てなければならない。よく知られているようにCPUスピードの向上とディスク入出力のスピードの向上を比べると、後者の方がずっと悪く(トラックアクセスはもっと遅い)、処理スピードを増してもシステム全体のスループットの改善につながらない可能性がある。LSIの素子に比べバスを駆動する素子はどうしても遅くなるため、ディスク入出力の並列化が必要である。

このため、現実的なデータベースマシンは次のような構成になると考えられる。

- a)大容量主記憶の利用
- b)複数の汎用マイクロプロセッサの利用
- c)ディスクキャッシュ、ディスクアレイ、線数の多いバス(または光バス)による二次記憶の入出力の高速化の実現

複数のマイクロプロセッサを利用すると分散データベースと同じような問題が生じる。このため、各マイクロプロセッサにデータベースを割り当て

共有データベースを持たせないShared-Nothingアプローチがオーバヘッドが少なく有効である。

TeradataやTandem等はこの方式で成功した例であるが、複数のデータベースにまたがる更新処理が多いような処理には適していない。従って、適合業務としては、クレジットカードシステム、銀行口座システム、旅行予約(座席、ホテル等)システム等が考えられる。今後高速コンピュータとして期待されている超並列システムは、構成要素相互間の通信量が多すぎると効率が低下してしまうので、やはりこのような局所処理を主体としたデータベースシステムへの利用が中心となるであろう。

ハードウェア実現を行う場合に、他の処理とデータベース処理で本質的に違うのは、扱うデータ量の範囲が非常に大きなことである。例えば、コンピュータ内の乗算回路を考えると、せいぜい64ビットまでの数しか扱わないため、64ビットの回路を使って32ビットやそれ以下の乗算を実行する場合と、それらの場合には別に作った32ビットの回路を使って乗算する場合との時間を比較しても違いはあまりない( $n$ ビットの乗算は $\log n$ に比例する段数で実現できるため)。しかし、データベースではデータ量が数個の場合から数百万個の場合までを扱う必要がある。このため、

- a)計算時間が回路サイズの関数ではなく、データサイズの関数となるような回路を用いる。
- b)データサイズ $n$ に対する回路サイズも、例えば $n^2$ に比例するものは少し大きな $n$ までしか実現できないため、 $n$ 以下( $n$ 、 $\sqrt{n}$ 、 $\log n$ 、定数等)になるものを選ぶ必要がある。
- c)どんなに大きなハードウェアを用意しても、それで扱いきれないデータ量を扱う必要が生じることがあるため、その場合に備えたソフトウェア的方法との併用方式を用意しておかなければならない。

関係の選択、射影等は、データが組単位で与えられるとハードウェア量は一つの関係に含まれる最大の属性数となり、処理時間は組数に比例する。結合はハッシュを用いると、同じハッシュ値の入るバケツサイズを適切に選べばデータ量に比例する時間で処理できる。長い系列の中に与えられたパターン集合の要素が含まれているかどうかを調べる系列の部分マッチは、ハードウェア量がパターン集合の大きさに比例し計算時間は系列長に比例する。

データベース内で比較的よく使われ、ハードウェア量が少なく(対象とするデータ量に無関係等)、ソフトウェア的処理では時間のかかる演算を

ハードウェア化すると非常に効果がある。例えば喜連川のデータベース操作の高速化や、日立の部分マッチハードウェア等をあげることができる。データのソートに対して、よく知られているToddのソータは上記a)の条件を満足しておらず、田中譲のソータ[IFIP80]はこの条件を満足している。

回路の大きさがデータ量 $n$ の関数となるようなハードウェアを用いてデータベースの高速化をはかる場合、自己矛盾的な要素がある。即ち、高速化をしたいのは $n$ の大きな場合であるが、そのような大きな $n$ に対してハードウェアを作ることは不可能である。従ってある $n$ に対する回路を繰り返し用いることになり、一般に定数倍程度の改良しか達成できないことになる。

主記憶のコスト低下や光ディスクやビデオ技術による大容量化を考えると、今までの2段階の記憶構成ではなく、多段階の記憶構成を考えるべきで、そのようなシステム上での並行処理、ファイル構成、回復処理には新しい方法が必要である。主記憶データベースが主流になっても、二次記憶はその大容量性と不揮発性から不要になることは当然考えられない。

電源異常や $\alpha$ 線照射といった外的条件の他に、ソフトウェアのバグや主記憶ないしはその周辺回路のハードウェア的故障といった原因で、主記憶の内容が変化する可能性が高く主記憶の二重化では対処できないためである。

現在ワークステーションで、CPUの速度が50MIPS、主記憶が1ギガバイトといったものがある。単純に3年で4倍になるとすると9年後には、3000MIPSで主記憶64ギガバイトといったワークステーションができることになる。これは驚くべき数字であるが、9年前に我々が利用できたパソコンやミニコンの能力と現状を比べるとこの程度以上の開きがあることが分かる。この反面、通信コスト(電話代)はこの9年間ほとんど下がっていない。衛星等の利用も周波数割当があるため、ローカルネットワークを除けば通信コストはハードウェアほどコスト低下しないとみるべきである。従って、分散データベースでは局所処理が増えても通信量を極力減らす方式を採用しなければならない。例えば、過去に通信量最小という条件で最適化を行う研究があったが、それを一般化して各サイトの計算量も含めた最小化を行うという論文がデータ工学国際会議で発表されていたが、このような論文はすぐに役に立たなくなるであろう。

RISCがCISCに優っているのは技術進歩の速さというものが主な要因である以上、数年間かけてデー

タベースの研究をする場合、数年先のハードウェアの状況を考えた(ないしはそれと独立な)研究をするべきであろう。

## 5.日本におけるデータベース研究の活性化

データベース研究の重要性の認識には、日本とアメリカで大きな違いがある。以下思いつくまま列挙してみる。

- ・日本のICOTに対抗して作られたMCCでは、演繹データベースだけでなくオブジェクト指向データベースの研究が行われている。

- ・NSFでは1989年末から90年始めにかけて、データベース研究の動向に関する4つのワークショップを主催している。結果としてNSFからデータベース研究に出る研究費が20%増額されたという。

- ・日本ではデータベースという用語とデータベースシステムよりむしろ大量に集めたデータを示すことも多く、データベース作成のためには文部省科学研究費の枠がある。

- ・本年度のアメリカ国防省の作成した重要技術リスト(全21件、内コンピュータ関係は8件)の中にデータ融合(日経新聞の訳)が含まれている。

- ・本年のアメリカのJTEC(日本の技術を調査する委員会)のテーマは日本のデータベース研究であった。

- ・日本の通産省の大型プロジェクト「計算機相互運用データベースシステム」は予定の7年の最終年度となったが、主力を注いだのはISOのOSIの実現で、分散データベースシステムやマルチメディアデータベースについては基本的な成果を得ただけにとどまっている。Interoperableという言葉をはやらせ、外国の研究者が注目していたのに、開発が中心となり、「なぜ、あまり論文がでないのか」と聞かれたことがある。

- ・データ工学国際会議では本年度からRIDE(データ工学における研究課題)というワークショップのシリーズを企画し、第1回目は本年4月京都で「マルチデータベースにおける相互運用性」で来年の第2回は「質問処理とトランザクション処理」であるが、これらの重要課題に対応する日本の研究者はほとんどいない(全体としても少ないが)

- ・アメリカの大学では、カリキュラムが重視され、短大も含めてデータベースの講義があるのが普通であるが日本では全くない大学がまだ多い。

- ・ソフトウェア科学会では、データベースがソフトウェア技術であるという認識があまりない(ソフトウェア科学会の企画した岩波のソフトウェアシリーズにはデータベースがない!)

・アメリカでは、特に軍関係の研究援助を受ける時、人工知能という言葉は避けて一般化されたデータベース等と記すのに、日本ではデータベースより人工知能の方が研究費が取り易い(そうである)。

・ACMのComputing Surveysの1986年から1990年にのったサーベイ論文の数は次の通りで、進歩の速いデータベースに関するものが多い。

データベース関係	18件
情報システム	3件(オフィスオートメーション、グループウェア、問題と解決)
人工知能	4件
アルゴリズム	2件
ソフトウェア工学	3件
耐故障性・テスト	3件
データフロー	1件
プログラミング言語	8件
パフォーマンスモデル・シミュレーション	5件
利用者心理	2件

49件中18件もあるのは、この分野の注目度を示しているといつてよい。

データベースシステムは、アメリカが特に自信を持っている分野で、上記で示したようにこのままでは21世紀になっても差が縮まるとは考えられない。データベースの重要性が急に低下するとは考えられないことから、日本でどのような対策を採るべきかを真剣に議論すべきであろう。

日本の企業でも、松下のプリンストン研究所では、アメリカの価値基準で判断したのか、データベース研究が中心となっている。(Lipton, Garcia-Molinaら一流の研究者を集めている)。さし当たってできることは、情報工学科のカリキュラムにデータベースを入れる、データベースのない大学には進んで教えに行く、田中(克己)らの産学共同コンソーシアムのようなものを幾つも試み、企業での研究活動を盛んにする、学会研究会活動を活発にする、どれもとても21世紀に研究活動を盛んにするのに間に合いそうもないが…

## 6. むすび

科学技術のほとんど全ての分野で、アメリカと日本の活発な研究者の数を比較すると10対1位に感じられるという。ACM SIGMODの会員数4000余りに対し情報処理学会データベースシステム研究会の会員数は500足らずでやはりその法則が当てはまる。先進国は自己のリスクで新しい分野を開発して行く義務があると思われ、事実米ソは宇宙開発等多くのプロジェクトを実現してきた。(アメリ

カ)対(日本)対(ヨーロッパ全体)のGNP比が5対3対6位になり、コンピュータ電子関係の生産では日米間にほとんど差がなく、日本の方がヨーロッパ全体を合わせたよりずっと大きいという現状を考えると、知的冒険に対する日本の貢献は余りにも少なすぎる。これには、最近はやりの「大学の崩壊」以外に「企業における研究の姿勢」といった点にも問題があると思われる。「知的分野で世界に貢献することが必要である」という認識が社会の通念とはなっていないことが全ての原因であろう。ソフトウェアやアルゴリズムの特許がけしからんと叫ぶよりも、知的所有権を主張できるような国になるべきである。)。

データベースシステム研究会では、オープンな議論をして、若い人をどんどん取り込み、国際交流を活発にするような地道な努力を、今以上にして頂きたいと思う。

「21世紀への挑戦」という勇ましい題をつけて内容は「21世紀に挑戦できたらいいなー」といったものになってしまったが、皆が危機感を持って地道に努力していけばやがて情勢も変化することを期待している。

付録:本稿をまとめるため、ACM SIGMOD国際会議の90年、91年の発表論文の整理を行った。参考のため、この2年間の分野別論文件数を挙げておく。

データベース設計	2件
データベースの処理	8件
データベース言語	1件
利用者インターフェース	2件
セキュリティ	3件
ファイル構成	8件
図形処理とファイル	6件
分散データベース	7件
並行処理(ファイルや分散にも含まれる)	6件
統計データベース	1件
CADデータベース	3件
主記憶データベース	1件
テキストデータベース	1件
オブジェクト指向	12件
演繹+オブジェクト指向関連	5件
並列ルール処理	3件
知識ベース、論理関係	13件

なお、91年4月に京都で開催されたIEEEと情報処理学会主催の次のマルチベースワークショップの会議録は情報処理学会から入手できる。

**First International Workshop on Interoperability in Multidatabase Systems**