

ISSN 0389-4010  
UDC 681.3.066.

# 航空宇宙技術研究所報告

TECHNICAL REPORT OF NATIONAL AEROSPACE LABORATORY

TR-869

数値シミュレータ・データベースの概念設計

安 喜 隆 幸

1985年8月

航空宇宙技術研究所  
NATIONAL AEROSPACE LABORATORY

# 数値シミュレータ・データベースの概念設計\*

安 喜 隆 幸\*\*

## Conceptual Design of a Data Base for Numerical Simulator

Takayuki AKI

### ABSTRACT

Construction of a numerical simulator is one of the recent significant projects at the National Aerospace Laboratory of Japan. The numerical simulator is a large computer system which is composed of a supercomputer, large capacity files, and other device control systems including TSS terminals. One of the most important features of the numerical simulator is its ability to control a data base. This data base contains a large amount of wind tunnel data, results of numerical simulations, geometric information from test models, and so on.

The Purpose of the present report is to display the design procedure for such a large and complicated data base. From this display, the total information about the data base for the numerical simulator can be clarified.

### 1. はじめに

航空・宇宙に関する技術開発は、その先端性・革新性をもたらす直接の利益のみならず波及効果によるそれも非常に大きい。このため先進諸国間では優位性確保をめぐる競争がくり上げられている。航空・宇宙分野の技術は大規模のものが多く、その開発費・期間が常に大問題となる。開発費は少なく期間が短いのが良いことは言うまでもないことであるが、それをどのような手順により実現するかの開発手法を持たなければこれらの問題に解決の望みは無い。

費用・期間の問題解決には、近年急速な発展をみせた計算機を駆使する手法が最良であることが認められている。それは、開発対象の物理モデル及び数

式化を適切に行なった数値シミュレーションにより、対象物又はその模型を用いての試験検証（アナログ・シミュレーション）と同等もしくはそれ以上の成果を得ることが可能となり始めたためである。こうした数値シミュレーション技術が確立されると、開発費・期間の両面で大巾な利得の出ることが多くの例で実証されている。そして世界各国共、この技術を重要視し必要な投資を行なう傾向を強めている。

当研究所においては、昭和40年代後半このような趨勢見通しにより数値シミュレーション技術獲得のための動きを開始した。ハードウェア面におけるアレイ・プロセッサFACOM230-75APの導入、ソフトウェア面での翼型解析コード群の開発は代表的な例である。これらの蓄積を次の如何なる方向に発展させるべきかが昭和50年代後半の課題となったのは当然のことである。その一つが単独コード群を結集させ、実用計算目的のシステムに発展させることである。この方向は次に述べる事情により緊急

\* 昭和60年5月20日受付

\*\* 空気力学第一部

性が強まっている。

現在、数値シミュレーション技術は米国が世界中で最大の量と高い品質での蓄積を有している。これは今日迄において大型高速計算機の開発第一人者の地位を確保できていたことに由来する。しかし、ほんの僅かばかり前よりこの事情に変化が現われ始めた。言うまでもなく国産計算機の飛躍的発達である。評価基準が鮮明ではないが一般的観測として、国産機のハードウェア性能は米国勢のそれと同等もしくは凌いでおり、それは汎用のみならずスーパーコンピュータにおいても同様とするものが多くなった。この観測は我国においてよりもむしろ米国側に多くなり、各種刊行物上で節ないし章の単位で日本製計算機についての記述を見受ける機会が増えてきている。専門刊行物では全巻特集と言う場合すらある。これらの事情が背景と言っても恐らく過言でないと思えるが、数年前より米国は計算機のソフトウェア流出に強い制約を設けるようになった。この制約は先端技術ほど厳重で、研究報告上での原始プログラムの非公開はもちろんのこと、いわゆるノウ・ハウ記述の消滅など以前と様变りの状態となった。

上記事情により、膨大な数値シミュレーションを必要とする大型先端技術開発は自己開発のソフトウェアに依らねばならない時代を迎えた。更に多国間の協同開発を必要とするような大型技術が増し、独自の技術を提供できない場合極めて惨めな結果となることがはっきりしだした。

数値シミュレータ計画は我国独自の計算機技術、自主開発のソフトウェアを結合し、数値シミュレーション技術における国際的優位性の確保を目指すものである。この優位性はシステムはもちろんその基底をなす個別ソフトウェア・レベルにおいても確保されていなければならない。このためには、あらゆる種類及びあらゆるレベルのソフトウェアの品質・信頼性を検証することが必要であり、これらの検証を効率良く実行できる支援体制も又不可欠のものとなる。検証に必要なデータ蓄積とその有効利用法を提供し得るシステムの構築はその体制を非常に強固なものとする。このため数値シミュレータ計画では強力なハードウェア、すなわちスーパーコンピュータの設置のみならずデータベース・システムを重要課

題に採りあげその構築を当初より計画に含めていた。

データベース・システムは上述の検証目的のみならず、数値シミュレータにおいて実用目的のシステム・ソフトウェアを稼動させる段階においては、目的遂行のため為すべき仕事、採るべき行動の情報提供源としても動作し得る。この究極として人工知能（エキスパート）システムへつながる道を見ることができる。この将来へ向けての第一歩としても数値シミュレータ・データベースの構築意義は大きい。

さりながら、数値シミュレータ・データベースのようなシステムは前例が無く、その構築には多くの困難を伴なうであろうことを予測できる。このため事前作業として調査の積重ねと、それに基いた設計作業を進めて来た。構築の実行段階に移った現在、それらの作業結果を取りまとめ数値シミュレータ・データベースの像を公開しようとするのが本報告の目的である。

本報告の作業は、当所計算センターにおける研究テーマ「NS システムの研究 システム設計に関する調査・研究」の中で行われた。当初は計算センターのみのメンバーの手で行われていたものを本著者が上記テーマに参加し、後続させてきたものである。ここに記し、本報告執筆者の経緯説明とする。

## 2. システム設計

データベース・システム設計は規模の増大、利用法の多様化をはかるほど複雑かつ多岐な検討作業を必要とする。これらの検討結果から概念設計がまとめられ、実施段階における詳細設計への指針を与える。本報告の目的は、数値シミュレータ・データベースに関して行なった概念設計の内容を提示することである。しかしながら、それらの個々は詳細設計の実務者以外には興味の無いことが多いため、結果の羅列的展示よりも検討経緯の記述に重点をおく。一般利用者においては、こうした背景事情の説明のほうがデータベース・システムの理解に適切であると考えられる。

データベース・システム設計は要求分析 (corporate requirement analysis) から出発する。要求はデータベースの設計に必要な情報として、そのデータベースに関連を持つあらゆる人々から収集する。

データベースの利用、管理、運用、データ提供等いろいろな立場の人々の要求を集める。設計者はそれらの要求を分析し、整理統合などの措置を加えた後、要求処理に必要とされる技術条件としてまとめる。これらの条件をもとに情報分析 (information analysis and definition) 作業に入る。この段階で各種の技術情報をもとに要求分析結果との対応づけを行ない、データベースとしての概念モデルを仕上げる。ここでは利用者とデータベースとのインタフェースを行なう操作言語に関する定式化作業も入ってくる。以述に述べた分析結果を用いて、論理設計 (logical database design)、物理設計 (physical database design) に関する作業を進める。これらの作業の目的は、データベースの性能・完全性・整合性・障害回復などのシステム内部に関係する問題の解決である。従ってシステム設計専門家の助力を必要とする困難な作業であるが、一般利用者にとってはほとんど関知する必要の無い領域である。

今回は、これら四領域における設計作業のうち前半二つについて述べる。ただし二つを厳然と区別した形式ではなく、先に断った主旨に沿うよう改変して記述する。

## 2.1 システムの形態

本報告の到るところで繰返すことであるが、数値シミュレータ・データベースは蓄積データ量が100 GB台と想定される大規模データベースである。

大規模データベースでまず問題となるのが、形態として集中又は分散のどちらで実現を計ろうとするのかということである。何れの形態を選ぶかにより前述の設計段階における作業が内容・量ともに大巾に変化する。結論の前に両形態に関し簡単な説明をしておこう。

### (1) 集中形データベース

センタ・システム内に全てのデータを蓄積する最も単純な形態である。データモデルが一種類、利用形態が定形的であるなどの好都合条件において大規模化の実績がある。ただし、ネットワーク・モデルの場合なので関係モデルに対して成功するかどうかはわかっていない。モデル混在のデータベースについては判断材料が全く無い。

### (2) 分散形データベース

構成方法に、トップ・ダウン、ボトム・アップの二通りがある。これらが網目状に結合され全体として一つの大規模データベースとなる。全体として共通のデータモデルであるべき必要は無く、各種のデータモデルが混在していてもよい。ただしデータベースの利用者にどのように(出来れば一つのデータベースに)みせるかの手法が必要となる。このことに関して完全な解決法はまだ得られていない。また、データ伝送のための基本的なハードウェア・ソフトウェア両技術を始めとして、異種データ・モデル間のデータ変換、データ保守、安全対策等未開発の技術問題が多い。

以上のように両形態ともに問題が多く、何れを採るかの判定に迷うが、ここでは集中形データベースを考える。数値シミュレータ・データベースに関して先例が無く、試行錯誤的に構築を進めるしか方法が無い。従って小から大、単から複へと蓄積を重ねる手法が最も危険性が少なく確実性が高いと思えるからである。

## 2.2 ハードウェア

データベースの最も高度な利用法は研究や設計の種々の段階において、利用者が自分の思考過程を詰めるための情報参照を行なうことであろう。在来手法では適当な媒体情報の頻繁な参照、例えばデータシートを何度も繰ってこのことが行われてきた。これに相当する作業をデータ端末機の操作に代えた場合、必要な情報が迅速に提示されないと思惑中断が起き作業に大きな支障をもたらす。このことは、データベース・システムが利用者の思考過程と同期した情報の呈示、つまり応答速度を持たなければならないことを意味する。この応答速度に関する制約は、サーチ・ソートなどの準数値演算処理が大部分であるような比較的単純なデータベース・システムにおいても厳しいものである。このためデータベース・マシンと呼ばれる専用機構が研究され始め、商用機として発表されたものもある。商用データベース・マシンは関係モデル型のデータベースで使用することを前提としており、ここで想定しているデータベースにおける機能に対しては限定的である。われわ

れのデータベースでは大量データに数値演算処理を必要とする機能要求があり、準数値演算処理を高速化するだけでは解決し得ない特異性を持っている。多次元データの図形表示においては、幾何学的計算のため多量の浮動小数点演算処理を必要とする。一般に利用者の多様な要求に応じてパラメタの値を可変にした場合、パラメタ値に対応して処理済の型式でデータを保存することは現実的でない。

数値演算処理は準数値のそれと異なり局所性が強く一括处理的にシステム資源の使用を認めたほうが効率がよい。しかし、このことは処理装置の負荷を増しシステム全体として応答性を低下させる要因となる。このように相反する条件のもとで実用的な応答速度を維持するには、データベース用システムであることを明確に意識したハードウェア構成にしておく必要がある。とはいえ、商用データベース・マシンの現在機能が限定的であること、より高機能が期待される知識ベース・マシンは研究が始まったばかりという状況下である。想定される数値シミュレータの運用開始時点において選択し得るものは汎用機によるシステム構成とならざるを得ない。

本データベースを数値シミュレータ・データベースと呼ぶのは、あくまでも数値シミュレータ利用の支援を強く意識しているためである。よってデータベース運用のためのハードウェア構成は数値シミュレータ採用機及びそのシステム構成と一体のものとして考えるべきものと言える。それが具体的段階に至っていない現在、汎用機によるシステム構築と言う結論までは良いが実際的な構成まで議論を進めることは早計である。ここでは、そのような段階に至ったとき考慮すべき事項を列記するだけにとどめる。

#### (1) 中央処理装置

先にくどくどしく述べた理由から、高速浮動小数点演算能力に注意する。一般的な命令処理速度も十分高速でなければならず、MIPS 値で 2 桁のものが望ましい。場合により複数化を考えてもよい。

#### (2) 主記憶容量

ソフトウェア設計 (2.3) で述べるように、データベース管理システムは既存 (市販) ソフトウェアの組合せ構成とするから、それらの塔載に必要な物理記憶容量を確保する。応答性を高めるため、各ソフ

トウェアの手続部分で主記憶争奪を発生させないことが特に重要である。現在時点での見積りでは容量下限として 4MB を必要としている。従ってオペレーティング・システム及び想定同時利用者数とそのデータ部などに必要な量を加算して主記憶の全物理容量を定める。

#### (3) ファイル・システム

ファイル・システムでは、その容量は当然のこととして物理的配分にも十分注意を要する。格納効率よりもアクセス効率を重視すべきである。ファイル・システムの構成要素である磁気ディスク装置の記録容量は近年飛躍的な増大を実現してきたが、アクセス速度には見るべき向上が得られていない。このため情報を基準として見た場合の平均的応答速度が低下しシステムのボトルネックになってしまうことは十分ありうる。ディスク装置の容量のみにとらわれない構成を考えるべきである。

また磁気ディスク装置以外の新技術による大容量記憶装置を使用できる可能性が大きくなってきている。この場合には容量・アクセス効率に加えてそれらの信頼性も検討事項に含めておきたい。たとえば光ディスクは有力な新技術大容量記憶であるが、ビット誤り率に磁気ディスクと比較しまだ問題があることを考慮に入れておくべきである。

ファイル・システムに関してさらに述べるべきこととしてバックアップと障害対策の問題がある。これらは運用技術のみでは、どうしても対処しきれない問題を生ずる。数値シミュレータ・データベースが 100GB 台のデータ蓄積量を持つことを考えると、磁気テープのみでのバックアップは不可能である。磁気テープ一巻 (6250 RPI, 2400ft) はたかだか 100MB 台<sup>\*)</sup>の収容能力だから、1000 巻程度の量になってしまうので到底現実的と言えない。磁気ディスクによるバックアップはもちろん交替予備装置の設置数も通常の計算機システムより余分に保有すべきである。

---

\*) 計算上 180MB 一巻となるが IRG 等を考慮すると、その 1/2 ~ 2/3 が実効データ収容量である。

## 2.3 ソフトウェア

既にハードウェア構成において述べたように、数値シミュレータ・データベースでは機能実現のために特殊な処理装置を持たない。よって要求される機能のほとんどをソフトウェアにより実現させることになる。そしてデータベース操作コマンドから窺える通り、機能は複雑多岐である。これらの機能要求を完全に満たす市販のデータベース・ソフトウェアは現在のところ存在しない。そこで自主構築の道を考える。

自主構築の手法として、(1)新規開発、(2)市販ソフトウェアの組合せが考えられる。新規開発は自由度が大きく望ましい手法であるが、人材確保とか開発費用調達など現状勢下での最大難問に当面し実現性に乏しくなる。一方、市販ソフトウェアの組合せは新規開発と比較し上記のような意味での問題は減る。すなわち、個別ソフトウェアの新規開発負担が軽減できる。しかし、個別ソフトウェアを導入しただけでは全く意味をなさない。

市販個別ソフトウェアは単体で運用でき、その限りでは利用し易いように作成されている。その反面各ソフトウェア相互間の共通性、関連性に欠けるのが常である。これを利用者側から見ると、それぞれのソフトウェアに個有の使用法を習得しなければならない煩わしさとなる。また、データベース・システム作成ないし管理者側から見たとき個別ソフトウェアの仕様に合わせた異なるデータの準備などを要求され、創成や保守に問題が出る。以上のような共通性、関連性の欠如がデータベースの利用価値を激減させることは確実と言える。

まず何よりも重要なことは利用者にとって単一のデータベースとして使用できることである。従って、個別システム間にある問題を内部操作によって解決すること、すなわちインタフェース・ソフトウェアを作成する必要が生ずる。このため個別ソフトウェアとしては、インタフェース可能でありその仕様が明確となっているものを選択対象にしなければならない。つまり個別ソフトウェアの機能がいかに勝れていても、他ソフトウェアとの結合能力を持たないものを対象とすることはできない。

以上の前提のもとに数値シミュレータ・データベ

ースのソフトウェア体系を考えてゆく。データベース操作コマンドの機能を分析することにより準備すべきシステム機能がわかる。このシステム機能を最大限実現し得る市販個別ソフトウェアの仕様を調べる。この結果から、全機能実現に必要な新規ソフトウェア（個別作成もしくは既存品の拡充）およびインタフェース・ソフトウェアを洗い出す。以上がここでの作業手順である。

操作コマンドの機能分析についての説明は長くなり過ぎるので別の機会に譲る。分析結果はシステム機能として下記のものが必要であることとなった。

- (1) 検索
- (2) 数値情報管理
- (3) 図形処理
- (4) 会話処理
- (5) バッチ処理
- (6) 機密保護

以上のうち(6)の機密保護は操作コマンドの機能に露に出ないものであるが、システムの管理運用に不可欠な機能である。これに関してはシステムの安全対策(5.2.2)においてより詳しく述べる。

### (1) 検索

検索コマンドの対象となるデータは大部分が文字である。数値データは構造が単純で数量は少ない。これらに対しては関係モデルによるデータベース管理システム(Relational Database Management System; RDMS)で対処可能である。RDMSは多種のものが商用製品として市場に出まわっている。計算機メーカーのみならず応用ソフトウェア・メーカー製品も加わって多様である。基本的にはどれも同じであるが、他ソフトウェアとのインタフェースを条件とすると計算機メーカー製品のほうが有利である。計算機メーカー製品の場合、インタフェースの支援機能をも他のソフトウェアが存在し主記憶上でのデータ転送を可能にする。このためファイル経由のような迂遠な方法で応答速度を下げる心配が無くなる。

RDMSの場合、利用者インタフェースとして個有の非手続型言語を有する。その機能は関係操作を基本としてかなり高水準と言えるが、検索以外の他のデータに関して不足するものがあるので陽には使用しない。このため数値シミュレータ・データベ

スに検索コマンドを附加し統一性を保つと同時に、代償として検索コマンドを RDMS の操作言語に展開するインタフェース・ソフトウェアを作成する。

関係モデルは利用者からハードウェアやオペレーティング・システムを隠すので使い易いと言われる反面、それだけ内部処理の負担が増し性能上の問題を生ずる。そこへ更に負担を重くするインタフェースをつけるのは芳しいことではないが、統一性を徹底させるためあえて目をつぶる。その償いのためにも極力高速中央処理装置の導入を計らなければならない。

検索対象データの多くは次項に述べる各種数値データの原本に含まれている。特に風洞試験データではほとんどを原本より抽出することができる。このため RDMS が扱うファイル形式への変換をソフトウェアで行なうことが作業能率・データの整合性の面で望ましい。このようなデータ生成用ソフトウェアの作成が膨大量になるのを防ぐため、原本データの書式を統一化しようデータ提供者に協力を求めることも必要となる。

## (2) 数値情報管理

数値情報には、風洞及び数値シミュレーションなどの試験結果ならびに模型形状データ等が含まれる。情報単位当りのデータ数が多量であること、多次元データであること、データ相互間に特異な関連性を持つことなどの特殊条件が重なっている。これらの特殊条件は RDMS の対象データとして不適格であり、CAD/CAM システムが扱うそれに近いものと言える。RDMS と同様 CAD/CAM システムにも市販製品は数多くある。市販製品は適用形態を限定（たとえば特定業種向けに）するものが多いため、機能を追い求めると複数個のシステムが必要となる。これは重複機能の無駄を含んだままシステム規模を大きくすることになり得策でない。一般市販製品における最大の問題点は

(a) ホスト計算機やそのオペレーティングシステム

(b) 端末機器のハードウェア構成  
などが限定されることと、

(c) インタフェースの非公開  
である。応用ソフトウェア・メーカー製品は(b)、(c)に

関して特にこの傾向が強い。ホスト計算機メーカー製品は当然のことながら(a)、(b)が限定される。(a)は致し方無いとしても(b)の条件が問題となるのは経済的負担を増し好ましくない。

以上の理由により、数値情報管理に関しては特別な既製ソフトウェアの導入を当面見合わせる。ここで当面と断わるのは、数値情報管理に使用可能と思われるシステムの出現に希望を持てるためである。Data Dictionary/Directory (DD/D) がそれであるが、初期の頃の DBMS の支援ツールとしての役割を脱しつつあり、多様な用途への展開が計られつつある。外国では既に商用製品も発表されている。これらは DBMS との組合せ、又は単独で稼動可能なものである。データベースの大規模化に不可欠なものとしての認識が広まっており、国内でも生産される見込がある。少なくとも既存 DBMS の機能拡充又はサブシステムの形で提供されることは期待できる。問題は数値シミュレータ運用開始時期にどの程度の機能のものが入手可能かということである。

以上のことを念頭においたうえ、以下の手法で当面の対策とする。

(a) 前項でも述べたように、原本数値データより検索情報を抽出し RDMS 用のファイルを生成するソフトウェアの作成

(b) 原本数値データを一般ファイルの水準におく。オペレーティング・システムの RACF 機能を利用して一般的管理を行なう。それでもなお不足する部分（主に機密保護）は管理用ソフトウェアの作成でおぎなう。コマンド機能の実行ソフトウェア（コマンド・プロセッサ）を作成する。

(a)は DD/D のメタ・データに対応するファイルを作成することにより、検索コマンド処理の RDMS 内閉じ込め処理効率低下の防止をねらっている。(b)は末端利用者と原本数値データ間の隔離を確実にし、システムの安全性を高めるのが目的である。

コマンド・プロセッサ作成の負担はかなりなものになると予想される。また一旦作成しても、それが直ちに決定版となり得ないだろうことも十分言える。実践経験に基いた改善要求が出されることは必定だから、そのための作業が運用開始後にもあると思わ

ねばならない。DD/Dが入手し得たとしてもコマンド・プロセッサ部分の作業量はそんなに減らず、管理運用面での負担減のみであることに注意する。

### (3) 図形処理

図形処理ソフトウェアは広い意味でコマンド・プロセッサに入るが、図形は文章・数表などの文字に対するのとは全く違う処理法を必要とするので別個の副体系として扱う。

ここでの図形処理とは、(2)の管理下にある数値情報を視覚化表示する作業を示す。この意味は、風洞試験において可視化技術を用いて得られるような情報の計算機表示作業を、画像処理と呼んで区別したいためである。両者とも計算機の端末に表示されてしまった後では画像という共通語で呼べるけれども、データの取得や処理方法において相当異質な技術を用いることを認識しておく必要がある。両者の区別を際立たせるものは情報源或いは対象物と言ってもよい。図形処理では架空のものでも扱い得るが、画像処理では実在のものでなければならないと区別できる。

数値シミュレータ・データベースに画像データを含めることは非常に望ましいと思うが、以下に列举するような理由から今回の計画には含めなかった。

#### (a) ハードウェア関係

まず入・出力に特殊機器を必然とすることが第一の困難である。形状のみならず色調も情報であるとか、高精度、異種（写真・フィルム・ビデオ）情報源などの条件から入力装置が複数化する。出力装置がディスプレイのみならずハードコピーにおいても高級品を必要とする。

第二は、データ量が一般的に膨大となりファイル・システムをさらに大きくさせる。また処理においても画像のそれに個有のハードウェア機構を備えた専用機でないと実用的性能が得られないことを第三にあげる。

#### (b) データベース技術関係

画像データベース技術は画像処理自体のそれと比較し開発が遅れている。各分野の特殊事情を加味した上での研究は僅かずつ進んでいるようであるが、まだ一般利用者向けシステムと言えるものが無い。

画像処理に話が流れ過ぎたが、ここで元へ戻す。まず図形処理における市販ソフトウェアの現状に目を向ける。図形処理における問題点の一つに次元の差異によるものがある。二次元（平面）図形と三次元のそれとの間には技術的難度に大きな開きがある。それを反映して市販ソフトウェアの多くは機能的に二次元図形処理にとどまっている。また技術的難度のほかに厄介な背景事情も重なっている。それは図形処理に関する国際規格制定の問題である。FORTRANなどの計算機言語と同様、図形処理にも統一規格制定の機運が高まり国際的な協同作業が進められてきた。この結果がGKS（Graphic Kernel System）としてISOにまとまり、ANSIを始めとして各国別標準規格にも順次制定されるような段階となった。当然市販ソフトウェアはGKS適合を目指して開発ないし改造を行なうことになる。問題なのはGKSが現在二次元処理までの機能制定で、三次元については尚検討中ということである。このことは市販ソフトウェアにとって開発目標が定め難いことになり、三次元処理ソフトウェアが少ない理由に数えることができる。

市販ソフトウェアの特徴として更に特定ハードウェア必須という条件が多いことをあげる。図形に個有の処理をホスト計算機で全て実行せず端末側に分散させる傾向が強くなっている。高機能・高速・高操作性を要求する程端末側ハードウェア機構が増大し、この傾向に拍車をかける。結果的にそれらの存在を前提としてソフトウェアを製作しセールスポイントに掲げて出荷するという循環になる。

数値シミュレータ・データベースの利用に際して特定の端末機器を準備させるのは好ましくない。従ってホスト計算機の処理負担増にはなるが、ソフトウェアによる処理を考える。とは言っても如何なる端末機器にも対応することは難しいので、一般的処理部分と機器対応部分（ドライバ）にソフトウェアを分割する。一般的処理部分とは図形の回転伸縮のように端末機器に依存せず実行できるものを指し、ドライバは端末機器特性に合せた処理（コード変換、図形スケーリング等）部分を受持つ。ドライバと言えども無制限には作成し切れないので何種類かに制限することはあり得る。このため端末機器の選択が



完全自由とはなり得ないが、上述のように特定機器が必須という強い制約は緩められる。

二次元図形の場合、それが線で表現されるものならば効率を除いて処理技術上の問題は少ない。階調変化を伴った面情報としての出力が要求されると処理技術だけでは済まない問題が起こる。階調と情報の量的大きさの対応づけを如何に行なうかがその一つである。たとえば流れ場の密度分布を多色階調で表示することを考える。表示可能な階調の全てを密度の最大と最小に対応させるか、或いは部分帯域に対応させるかをあらかじめ決定しておくことは難しい。流れ場の状況とか利用目的により対応づけを任意に変更できるようにすればよいが、それでは図形処理専用システムの機能に近づき過ぎデータベースとしての性格が不明になる恐れを生ずる。現在のところどの程度の機能まで許容するかきめてないが、データベースとしては最低段階までとし、より高度なものは他の図形処理専用システムによるという使いわけが解になると思われる。

三次元図形において問題は二通りに分けられる。試験データ等で多パラメータ間に亘る情報を包括的に表示する場合と、模型データのごとく三次元構造物を表示する場合とである。

試験データ等は基本的に線を用いて表示する。これを三次元空間内の曲線として各パラメータ方向に変化する様子をわかり易く画くことは意外と難かしい。投影方向、視平面等の幾何学的条件を与えれば如何なる方向からでも空間曲線（群を含む）を観るように変換を行なうことは単純な計算で済む。それだけではどうしても理解できない、或いはどうしても解釈可能な図形になるのが普通である。図形の理解を助けるため立体観の附与を必要とする。投影法を変えるのも一つの手であるが、それがかえって無用の誤解を派生させることもある。ために科学・工学系統では平行投影がほとんどである。通常は補助線、座標面への投影を附加するか追加線を入れて曲面のように見せるなどの手法を用いる。この場合隠線消去手法を併用すると一層立体観が強調され理解し易い図形となることが多い。この時の隠線消去は後述のものと比較して遥かに単純な算法で実行できる。従って或程度の自由度を許容した機能を与えることが

可能である。

試験データの表示には全く二次元的処理で三次元的関係が理解できる手法もある。例えば、 $120^\circ$ 間隔で平面を分割し、それぞれを座標面と考えて曲線の投影を行ない三軸方向における座標値の増減と曲線変化の模様を読み取らせるものがある。このダイヤグラムの手法は三次元以上でも効果をもつ場合がある。上述の曲面化手法より処理時間を少くできるので積極的利用を考慮すべきである。特に設計関係では実用している場合もあると思われるので、在来習慣との継続性保持の意味からも使用する必要性があるかも知れない。

普通に三次元図形処理と言うとき、模型データのような構造物に対するものを指す。この場合は上述のものと非常に異なった処理になる。

三次元構造物（以下、物体とする）の形状を計算機処理データとして記述するとき、次の3種類のデータ・モデルの何れかになる。

- (a) ワイヤフレーム・モデル
- (b) サーフェス・モデル
- (c) ソリッド・モデル

ワイヤフレーム・モデルは物体を外形線のみで表現するときのもので、基本的なデータ構造は物体面上の点座標とそれら相互間の連結関係とからなる。示された連結関係と点座標を参照しながら順次表示を行なう。連結関係には、連結の有無、連結状態（直線・曲線）、線種（実線・破線など）、色調など必要なだけの情報を含めておく。実行時には、指定された投影法と視点位置などのパラメータ値から表示装置上の座標に変換する幾何学的計算を行なう。その計算量は補間等の介在作業が無ければ、点座標数にほぼ比例する。点座標・連結関係のデータ量が大きいときにはファイルから分割して主記憶へ読み出し処理する。処理は直列に行えるので、このことを考慮しても計算量は線形時間（処理時間がデータ量に比例すること）で済む。このため入力パラメータの任意指定の許容度を大きくすることも可能である。

ワイヤフレームの欠点は隠線消去が困難なことである。凹部を持ったり、多重構造の一般的物体に関しては隠線消去不能である。このため複雑な形状をもつ物体の表示に適さない。

サーフェス・モデルは物体表面を分割し、面ごとにその情報を貯える。面が曲線の場合は、近似的な多角形平面かパラメタ表示可能な曲面で構成させる。近似的多角形としては三角・四角形を用いることが多いが、より適切であれば五以上の多角形でもよい。滑らかさを求めると分割数を大きくしなければならない。少ない分割数で滑らかさを表現したい場合、パラメタ表示の曲面を用いる。Coonの曲面パッチはその一例で大域的に滑らかな曲面を少ないデータで表現できる。この曲面パッチは数学的にもよく整備された曲面理論を持ち、航空機設計などで実用済のものである。

面情報としては、頂点数、頂点座標のほか面特性（平面近似では不要だが曲面パッチでは縁線基準の曲率・振率・法線など）を持たせる。必要ならば色調等の情報を附加する。面の表裏は外向法線（表）を一定方向に見る（例えば左手）約束で頂点データを配置することで区別される。

表示装置座標への変換は線形時間の計算量であるが、隠面消去はデータ数の2乗に比例する多項式時間となる。面の総数を $N$ とすれば、ある面が他の面により隠されるかどうかは他の $(N-1)$ 面と比較しなければならないためである。もちろん最初から表示を要しない（見えない）面を前検査で取除くことはできる。そうした残りが $N$ 面としても、 $N$ が大きければ単純な比較算法では実用的でない場合が多い。このため隠面消去に関する効率的算法がかなり古くから研究されてきており、今日では基本的算法は出尽したと言われる程になった。何れの算法にしろソーティング問題としての性格を備えておりデータ構造と効率とが強い結びつきをもつ。このため風洞及び数値シミュレーション試験に使用する模型データとは大巾に異なったデータ構造をもたせる必要から、図形処理用の模型データを別個に用意しておかねばならない。

サーフェス・モデルは物体を表面のみで定義する。これは実物体の状況とは関係なしに内側は空洞と見ることに等しい。ソリッド・モデルは物体を中味の詰まっているものとして定義する。この結果サーフェス・モデルでは実現できない処理、たとえば切断面の表示などが可能となる。ソリッド・モデルにお

ける物体記述はサーフェス・モデルよりも複雑になる。時にはプリミティブ（球・直方体・錐体など）の集合として表わすこともできるが、滑らかさを要求すればデータ数は増加する。また今のところ個々の物体は凸であることが要求されるので、凹部を持つ物体は全て凸である部分物体に分解しなければならない。データ数の増加と記述構造の複雑化が避けられない。このようにして分解した物体を再表示する際、分解線消去の手間が余分にかかる。その他ソリッド・モデルにはまだ問題が多く、データ構造や算法に関して研究中と言える段階だが極めて現実的なモデルなのでCAD/CAM分野を始めとして三次元処理の主流となるのが確実である。

ソリッド・モデルにおける隠面処理はサーフェス・モデルと同様に行え、隠面消去の算法計算量とも同一である。ソリッド・サーフェス両モデルに共通に隠面消去の手間が大巾に簡略化される場合があるが表示装置が限定されるので一般的でない。また現実感溢れる表示法として最近注目を集めている光線追跡法は計算量が余りにも膨大過ぎる。などなどで、図形処理の最新技術による表示はそれ専用のシステムを用いることにし、データベースの三次元処理では考えないことにする。

データベースとしてはソリッド・モデルに限定したいが、在来的経緯（特に設計関係）の問題もありワイヤフレーム・サーフェスを必ずしも一方的に切捨てられないのが実情である。以上、図形処理に関してことさら長い説明を行なったが、その理由として図形処理ソフトウェア作成がかなり大きい作業であると思われたことをあげる。

#### (4) 会話処理

データベースの使用形態は会話型端末を利用するものと一括処理で依頼するものとに分けられる。これらを作業内容に応じて使い分けることが望ましい。一括処理については後で述べる。

会話型端末利用において、処理プログラム作成上決定しておかねばならない問題を述べる。まず利用者にデータベース状態を意識させるかどうかがある。

利用者にデータベース状態を意識させる場合、利用者はその状態を作らねばならない。実際にはデー

データベース呼出コマンドの投入のみで、データベース操作に必要な条件は処理プログラム側で設定する。もちろんデータベース作業終了時の終結コマンド投入も必要となる。このようなセッション移行は会話型端末において通常的に行われており、特別に異なったものではない。

これと反対にデータベース状態を意識させない場合、利用者はデータベース操作とその他の会話処理コマンドを任意に投入できる。データベースの一つのコマンドに対する処理が終了した後、常に次の任意コマンド入力待状態に自動復帰する。利用者側から見た便利さの点ではこちらのほうが勝れている。しかし、システム作成の側から見た場合非常に大きな問題がある。その第一は既存会話処理システムとの結合方法である。既存システムとの結合には、コマンドの解釈・実行部を全て含めた一体モジュールとして追加組込するのが考えられる。この場合処理効率は良いが、既存会話処理システムとのインタフェースや既存処理システム内の変更発生で新たな問題を生ずる懸念が十分にある。また既存システムのコマンドと整合性を計る（たとえば同一名のコマンドの有無）など複次作業も増す。次の結合方法としてコマンド・プロシジャ機能を用いることが考えられる。この方法だとコマンド解釈部の展開に難点が多く、利用者に直接見えない形式ではあるが処理プロセッサを順次呼出すことになり能率が低下する。以上のどちらにしる問題が多く、かつ将来的にみて計算機システム変更ごとにこの種の作業を繰返す不利益も考慮しデータベース状態を意識させない方式は採らないこととした。

データベース状態中に一般的会話処理コマンドをどうしても使用したい場合が起り得ると考えられる。この対策として一時的にデータベース状態を中断、すなわちデータベース状態を終結させずに一般的会話処理コマンドの投入可能状態を作る機能を附加する。幾分操作が面倒になるが、既に PFD (Programming Facility for Display user) システムなどに類似機能があり、経験済の利用者も多いと思われるので実用上大きな支障はないと考えられる。

#### (5) バッチ処理

作業内容に応じて会話型端末からの処理とバッチ

依頼のそれを使い分けるべきであることを前に述べた。端末の機器構成しだいにより相当なことが会話処理で実行可能であるが、本質的にバッチ処理でなければならぬものもある。代表例として図形のプロッタ出力がある。数値シミュレータ・データベースでは一般端末にプロッタの接続を前提としてはいないため、プロッタ出力要求をバッチ依頼と見做す。このような事情もあってバッチ処理とのインタフェースを考えることにした。

バッチ処理においては、データベース操作コマンドのみを順次的に処理させるコーティリティ方式と、利用者のプログラム中にデータベース操作コマンドの記述を許し利用者プログラム内で更に高次の処理が実行できるようにするホスト言語インタフェース方式とが考えられる。前者の方式は、検索・データ転送のような単純流れの作業のみの場合には適当と思われるが、より高度な利用法を考えたとき機能不足であることは否定できない。そこでホスト言語とのインタフェース機能を含めたものにする。この機能により、利用者プログラムは自分の記憶領域へデータベースから直接データを読み出しそのまま必要とする処理を続行させるようなことが可能となる。このことにより、計算機システム内におけるデータの重複保管の無駄（データベースの目的の一つでもある）を省き作業能率向上が望める。

上記二方式を同一に取り扱うため、前処理プロセッサを作成する。前処理プロセッサは利用者のプログラム中からデータベース操作コマンドを検出し、それを後続のコンパイラが認識し得る形式に展開する。この処理効率をあげるため、利用者にはデータベース操作の開始と終了宣言記述の負担をしてもらう。ちょうど会話型処理において、データベース状態への移行と脱出コマンド投入を必要とするようにしたのと同様である。もちろんプログラム内においてデータベース操作の開始・終了の実行回数には制限を設けない。ただし、会話型処理における中断状態の機能は制限される。このような条件を除き、コマンドの記述書式は会話処理の場合と完全互換を計るので実用上の煩雑さが増すことは無いと言えよう。

現在のところ、RDMS に市販製品を採用すること、利用者数の圧倒的大きさ等の条件からホスト言

語としてはFORTRANのみを対象としている。

### 3. 操作コマンド設計

操作コマンドは第2章のシステム設計に属するものであるが、数値シミュレータ・データベースの機能が一般利用者にもはっきりと理解できる部分なので特に章を改めて説明を行なう。

#### 3.1 データベース操作のための利用者インタフェースと言語

利用者がデータベースから情報を取出す時、何らかの操作を行なう。操作への利用者側からの第一要求は簡易性であり、第二は迅速性である。この場合、迅速性は最低限の操作で必要な情報取出が可能であるという意味とする。簡易性には、操作法習得に余分な労力を使いたくないという意味も含まれる。しばしば、簡易性と迅速性とは相容れない条件となることが起こる。このため操作方式及び操作言語について考えておくことが重要である。

データベースに限らず種々のシステムを見てみる

と、操作方式には二通りの型があるように思われる。ここでは仮にそれらをシステム主導型と利用者主導型とに区分する。

第一のシステム主導型の特徴は、操作条件設定をシステム側プログラムにより逐次的に行なうことにあると言える。この型は更に次のような方式に分類することができる。

- (1) メニュー (menu) 方式
- (2) チュートリアル (tutorial) 方式
- (3) 質問応答方式

上記三方式それぞれの特徴・利点・欠点などを表3.1に示し個別説明は省略する。表3.1の特徴欄からわかるように操作習熟労力は少なく、簡易性という観点からは好ましく思える。しかし、何れの方式においても簡易性が意味を持つのは操作習熟度の低い利用者に対してであって、熟練利用者にとっては煩わしさの方が勝り迅速性の欠けたものになる。また、欠点のうちシステム・ソフトウェアの負担が大きいことも見逃がせない。数値シミュレータ・データベースのように未経験のシステムにおいては将来

表 3.1

	メ ニ ュ ー	チュートリアル	質 問 応 答
特 徴	操作条件が一括表示され、利用者側の選択に従って作業が進められる。	操作条件やその選択に関する説明が表示される。利用者はそれを読みながら応答入力を行なう。	システム側からの質問にたいし応答入力を行なう。応答はYES/NO又は数項目からの選択の形式となる。
利 点	操作条件間の関係が把握し易く誤操作が少ない。 簡単な学習で操作可能	ほとんど学習無しでも操作可能。 端末ハードウェアの制限少。	条件は左の二方式より短時間で済む。
欠 点	スクリーン管理等で端末ハードウェアに制約がある。 システム側ソフトウェアの拡張・変更に対する柔軟性低い。 バッチ処理との整合性悪い。	端末表示量が増大し、習熟度が上ると三方式中で最も煩雑感が出る。 既設定条件と矛盾が起きても修正が困難。 バッチ処理との整合性悪い。	条件選択に関して正確な事前知識を必要とする。 マクロ化に難点があり、繰返・定形業務においては煩雑感が出る。 バッチ処理との整合は良。

的な改変の必然性を計算に入れておかねばならない。この時でも、改変の影響を狭い範囲に留めるとすれば上記方式は何れも好ましくないものになる。

第二の利用者主導型は必要条件の全てを利用者が一方的に入力する点に特徴があると言える。従って、利用者は操作に関して何がしかの知識の習得を強要される。これは習熟度の低い利用者或いは利用頻度の少ない人にとって面白くない方式となるが、そのことに目をつぶれば簡易性の妥協を計り易くなる。

この方式はシステム主導型と異なり操作に関しての情報があまり提示されないので、操作習得が容易であるように考えておかねばならない。ここでは操作習得に関して最も問題となる操作言語について述べる。

操作言語は大まかに

(1) 親言語 (host language) 型

(2) 独立 (self-contained) 言語型

に分類される。親言語型が既存のプログラム言語に基盤をおくのに対し、独立言語型はそのようなものに拘泥しないのを特質としている。

親言語としては CODASYL 方式のデータベースに見られるように、COBOL, FORTRAN などのコンパイラ言語が採用される。それぞれの言語単体における仕様の国際標準化が進んでいること、両言語の利用者数が多く親近感を与えるなどの強味を持つ。とはいえデータベースを操作するとなると通常水準の言語知識、たとえば技術計算・事務計算のプログラム作成が可能という水準をこえたものが要求される。特に操作言語とデータ構造との結合が強く、それらを正しく認識しなければならない点に難点がある。これは、いわゆる末端利用者にとって不便極まりないことであり、データベース利用の障害になりうる。このことは古くから認識され、改善が試みられようとしているが未だ混乱状態にある。

先に例をあげた COBOL, FORTRAN などの親言語は本来手続型言語である。手続型言語は計算機に対して「いかにすべきか」 (how the task is to be done) を指示するために開発された。どのような言語仕様の拡充・改良を計ろうとも手続型であることの影を消し去ることは難かしい。一方、データベースではデータ構造・データ転送方式といった内

部操作に関わることなく必要な情報取出しが行えることを利用者は望む。このことをつきつめると、利用者が計算機に対し「何をすべきか」 (what is to be done) を指示する言語のほうに適していることになる。「いかに」を指示する言語を手続型と呼んだのに対し、「何を」を指示する言語を非手続型と呼ぶ。そしてこの非手続型言語の仕様作成において、既存手続型言語にとらわれない独自方式を用いるという意味で独立言語という。

誤解を避けるため注意をつけ加えると、手続・非手続の区分は言語の表記法のような外部的形式により断絶的に行うものではなく、よりどちらの度合いが強いかの程度で判断すべきものということである。従って、両型式間には連続的に言語が存在し得ると考えたほうが柔軟性に富む。

本報告で何回も繰返し述べるように、数値シミュレータ・データベースは在来型の文字主体と異なり、数値・図形など多彩なデータを取り扱う。そのため親言語型のように標準化の進んだものではかえってそれが制約となりうることも考えられる。かと言って独立言語でかつ非手続き一辺倒で済ませられないことも明瞭である。研究という非定形な利用形態では「何を」の全てを予測し固定化することが不可能なためである。このためまず独立言語であることで自由度を確保し、手続部分を最少或いは露わに要求しない中間形態の操作言語を考えることにした。

言語に関して常につきまとうもう一つの問題は、自然言語か人工言語の何れを選ぶかのそれである。この古くて新しく、かつ永遠とも思える問題に対し十分な根拠を示して選択理由を述べることは不可能である。

自然言語を採択すれば一見誰にでもデータベース（と限らず一般に計算機）システムの操作が容易であるように思える。しかし自然言語の持つ柔軟性を考えると、それに対応する機能をシステムに持たせることは容易でない。これらの機能はシステムが自然言語の操作者つまり人間並の知的水準に近づかなければ実現しないので、自然言語によることは今日的手法として適切でない。もちろん計算機科学分野でこの方面の研究が全世界的規模で進められており、次世代と言われる知識ベース・システムで可成りの

ことが実現しそうな見込はある。それ迄の間は在来型の人工言語で我慢するしかない。

人工言語でも自然言語風にみせかけることは可能である。特に非手続言語によればその可能性をより大きくできる。だがそれらはあく迄も見せかけだけのもので、現在の計算機技術では能率を下げるだけにしか役立たない。自然言語風にみせかけるほど計算機上の実際の命令群（手続型）に展開する手間が増える。そこでこのような細工に労力をかけることも放棄した。結局、ある規約のもとに所要条件を入力すれば目的が達成でき、かつ前に述べた簡易・迅速の両目標も実現できようということでコマンド列型の操作言語とする道を選んだ。今日会話型端末の普及により、この種の操作言語に経験を持つ利用者が多くなってきており頭から拒絶される危険性は少ないであろうとの予測も入っている。

### 3.2 操作コマンド

本節では操作コマンドの機能について概要を述べる。機能概要から数値シミュレータ・データベースの利用法が見えてくると思う。

コマンドは、機能面から系統づけて、

- (1) 検索
- (2) 利用
- (3) 支援
- (4) 運用

とわけることができる。(4)を除いたものが一般利用者向けのコマンドである。以下、(1)～(3)について簡単な説明を追加する。表 3.2 に各系統別に分類したコマンドの一覧を示す。

#### (1) 検索系

検索系は文字通りデータの検索を行なうためのものである。利用者はこれらのコマンドを駆使して所要データの有無を探し出す。この検索作業を効率化させるため

- (a) キーワード
- (b) 試験環境
- (c) 模型形状

などの検索パラメタ使用を取り入れた。キーワードは一般的にも使用されているので特長とはいえないが、現象や実験・解析手法を指定することで検

索経路を大巾に短縮できる場合が多くなる。

試験環境は風洞名・試験期間などを検索パラメタに使用するもので、かなり漠然とした既知情報（記憶）からでも検索したい場合を考慮したものである。ある種の実験があつた風洞で行われた筈であるとか、この模型の試験が何年何月ごろ行われた筈などの記憶から必要データを探したい場合が意外と多い。模型についても同様なことがある。これらの検索パラメタは本データベースにとっての特長と言える。なお、風洞名の代りに数値シミュレーション・プログラムを指定すると同様な検索が可能であることを特記しておく。

#### (2) 利用系

利用系は検索によって探しあてたデータを取出すためのものである。取出先として

- (a) 利用者が指定するファイル
- (b) 出力機器

一般文字型端末

グラフィック端末

ライン・プリンタ

グラフィック・プロッタ

#### (c) 利用者のプログラム

#### (d) リモート・ステーション

を考えた。(a)～(b)はセンタ直結型のもの、(c)はセンタ・システム内で走行中のものを指す。(d)の場合、その配下の機器がセンタ直結型として動作できれば、(a)～(b)と同じであるが、その他の場合はステーション側の処理に任せられる。当然のことながら取出先のハードウェアが処理できないコマンドは使用不可である。文字端末から図形出力コマンドは投入できない。

#### (3) 支援系

支援系には次のような機能をもつコマンドが含まれている。

- (a) ヘルプ
- (b) 既入力コマンド再使用
- (c) マクロ化
- (d) 標準値設定

ヘルプ機能を用いて、コマンド使用法、キーワード一覧表などを見ることができる。

既入力コマンド再使用は、既に入力済のコマンド

表 3. 2

## 検索系コマンド一覧表

コマンド名	コマンド修飾子名	機 能 概 要
ABSTRACT	MODEL NS PROG WT	指定されたデータセットの要旨情報だけを出力させる。 模型データ NS データ NS コード 風試データ
DB	NAME	データベースの総合情報を出力させる。 データベースのグループ別分類名
KEYWORD	MODEL NS PROG WT	指定されたデータセットが持つキーワードの一覧を出力させる。 模型データ NS データ NS コード 風試データ
MODEL	GROUP HISTORY NAME NAME/GROUP NAME/KEYWORD	模型データベースに関する情報を出力させる。 グループ単位での模型情報 指定された模型の試験履歴 指定された文字列が含まれる 模型グループ名 指定された文字列が含まれる 模型名、但しグループを限定する。 指定されたキーワードを持つ模型情報
PROG	CONDITIONS INSTRUCTION NAME/KEYWORD NAME/MODEL NAME/NS	指定された NS コードに関する情報を出力させる。 計算条件の表示 使用法の表示 指定されたキーワードを持つ NS コード名 指定された模型名に対する NS 試験のコード名とデータセット名 指定された文字列を含むコードの概要
TEST	DATE FACILITY HISTORY/NS HISTORY/WT ITEM KEYWORD/NS KEYWORD/WT MODEL/NS MODEL/NS TNUMBER WTTEST/NS	風試又はNS試験に関する情報を出力させる。 指定された期間内における風試情報 試験設備の一覧 指定された NS コードの使用履歴 指定された風洞の風試履歴 指定された項目の風試情報 指定されたキーワードをもつ NS コードの使用履歴 指定されたキーワードをもつ風試情報 指定された模型の NS 試験情報 指定された模型の風試情報 指定された試験番号範囲での風試情報 指定された NS コードに対応する風試情報

表 3.2

## 利用系コマンド一覧表

コマンド名	コマンド修飾子名	機 能 概 要
DATA	COPY/MODEL COPY/NS COPY/PROG COPY/WT DISPLAY DISPLAY/MODEL/NS DISPLAY/MODEL/WT DISPLAY/NS DISPLAY/WT GDISP/NS/FIELD MODEL/GENERATE PLOT/COMPARE=NS_WT PLOT/NS PLOT/NS/COMPARE PLOT/NS/CONVERGE PLOT/NS/FIELD PLOT/NS/GRID PLOT/WT PLOT/WT/COMPARE	指定されたデータセットに転送を行わせる。 模型データの取出し NS 試験結果の取出し NS コード (ロードモジュール) の取出し 風試データの取出し 模型データの表出力 NS 試験 模型データの表出力 風試模型データの表出力 NS 試験データの表出力 風試データの表出力 NS 試験データの場情報の図形出力 指定条件による 模型データの生成 NS・風試間の比較図形出力 NS 試験結果の図形出力 NS 試験間の比較図形出力 NS 試験の収束状況を図形出力 NS 試験データの場情報の図形出力 NS コードで使用した格子又はパネルの図形出力 風試結果の図形出力 異なる風試結果間の比較図形出力
MODEL	/PLOT /PLOT/COMPARE /PLOT/GENERATE /PLOT/SURFACE	模型に関する情報の図形出力を行わせる。 指定された模型の形状 模型間の形状比較図 指定条件により生成された模型形状 3次元模型の面表示

## 支援系コマンド一覧表

コマンド名	コマンド修飾子名	機 能 概 要
COMMAND	COPY INSTRUCTION NAME OBJECT PREVIOUS	コマンド操作に関する支援 打込済コマンドの再利用 コマンドの使用法 ある文字 用途別に分類されたコマンド名の出力 打込済コマンドの再表示
INDICES	WT NS MODEL PROG	指定されたデータセット・グループに含まれる索引名を出力させる。 風試データセット・グループに限定 NS データセット・グループに限定 模型データセット・グループに限定 NS コード・データセット・グループに限定
KEYWORD		ある文字列を含むキーワードの出力



表 3. 2

コマンド名	コマンド修飾子名	機 能 概 要
PARAMETER	DEFINE LIST	コマンド・オペランドのパラメタに対し偽名を与える。 偽名の定義を行なう。 定義済偽名の表示
USE	MACRO	マクロ命令使用の宣言

の中から必要なものを単に複写することにより再使用を可能とする。操作性向上を狙った機能である。

マクロ化は既に会話型処理で馴じみが多いものと同じである。定形又は準定形（パラメタ可変としそのつど入力する）的使用法における効率化を目的とする。

標準値設定は、作業時に出力形式などのパラメタが一定でよい場合、コマンドごとに設定せず一回の宣言で固定化を行なうためのものである。やはり作業能率の向上を目指したものである。

#### 4. ファイル・システム設計

ファイル・システムに関してのハードウェア構成上の留意点を 2.2 で簡単に触れた。本章では論理上の構成について概略を述べる。この構成を与えられたファイル・システム（ハードウェア）上でどのように展開実現するかは物理設計は、数値シミュレータ・データベース用計算機システムの導入過程の中で具体化される。従って現時点でこのことには触れない。

数値シミュレータ・データベースの情報量は 100 G バイトのオーダーと想定される。これだけの情報量をもつものは、現時点で大規模データベースに分類される。大規模データベースは現在も開発段階であり、極めて限られた分野においてのみ実用化が進んでいるに過ぎない。現状で大型かつ実用と言えるデータベース規模は 30 G バイト程度であるが、この規模を桁上りに増大させたとき以前のままで実用データベースとして存続し得るかの問題に現状技術では疑問のあることが適用分野の限定理由である。このように、情報量規模一つとっても、数値シミュレータ・データベースは特異である。

さらに、風洞及び数値シミュレーション試験結果

の大量数値情報、試験用模型の形態情報などのデータを含んでいる。これらのデータは、在来型データベースに無いような情報構造表現法、処理法を必要とする。これらのための技術は 1980 年代の課題と言われるアドバンスト・データベース・システムのそれに属し、評価が確定したものばかりではない。

上述のとおり、数値シミュレータ・データベース構築のための支援技術には今後の研究・開発をまたなければならないものが多い。従って、それらの支援技術を利用する側としては将来的な変更の発生を予期し、それに対応し得ることを設計前提としておかなければならない。

以上に述べてきた環境条件を考慮した結果、数値シミュレータ・データベースを以下のサブ・データベースに分割することとした。

- (1) 管理用サブ・データベース
- (2) 風洞試験サブ・データベース
- (3) 数値シミュレータ試験・サブ・データベース
- (4) 模型サブ・データベース
- (5) コード・サブ・データベース

上記のサブ・データベース中、(2)以下については格納されるべきデータ内容がそれらの名称から容易に推定できる内容には特に言及しない。

(1)の管理用サブ・データベースには、(2)以下の各データベース内ファイルの管理情報のほか、それらファイル中から抽出した情報を含めておく。このような階層構造をとることにより検索効率向上が望める。もし階層構造でないとすると、ごく単純な検索条件に対しても常に全ファイル読出しを伴うこととなり不利は明らかである。また、利用者に関係する管理情報、たとえば利用者ごとのアクセス権などもこのサブ・データベースに含まれる。

(4)の模型サブ・データベースを特に区分した理由

は、風洞試験と数値シミュレータ試験の相方で同一データによる模型形成を狙ったものであり、IPEG<sup>\*)</sup>の考え方と規を一にする。従って、将来的には3次元CADシステム機能を取込むことにより風洞試験模型製作図面の出力と言った展開を想定している。

## 5. 安全対策

計算機が人間の日常活動に広くかつ深い関わりを持つにつれ、それをあらゆる意味での災害から保護する対策の確立が大きな課題となっている。これらの保護対策は抽象的・観念的なものから学理に基いた技術に至るまで極めて幅広いものがある。それらを総称して安全(security)対策と言う。特に計算機のそれであることを明示するため、コンピュータ、データ、情報などの言葉と結合させることもある。ここでは単に安全対策と呼ぶ。

いわゆるコンピュータ犯罪が世間を騒がせたためもあってか、安全対策と言うと計算機システムに対しての或いはそれを利用しての意図的不法行為を防止する面が強調され勝であるが、もっと基礎的な面もあることを忘れてはならない。本章では、まず後者について簡単に触れた後、前者に関連した安全対策を述べることにする。

### 5.1 物理的破壊に対する安全確保

データベース・システムと言えども計算機システムであることに変わりはない。一般的な計算機システムにおける安全対策は当然として、データベース・システムに固有の安全対策も考慮する必要がある。

火災、地震、或いは人為などによるハードウェア直撃型の破壊、また電氣的・機械的障害が原因となって派生するような、いわゆる物理的破壊に対しての安全対策がある。このうち計算機システム内部に原因する物理的破壊に対しては信頼性技術の向上に伴って安全性が高くなっている。また適切な運用技術を用いることにより安全性を確実にさせることができる。計算機システム自体が原因でない物理的破壊に対する安全対策については、次に述べること

を除き、ここでの議論を超えることが多いので省略する。

データベース・システム内の最重要資源は言うまでも無くデータである。データはシステム内のファイル機構に収納されているものだけでは安全上無力である。通常バックアップ・ファイルと言う形で全く同一のデータをシステム外部で保持しておく必要がある。このバックアップ・ファイルが計算機システムと同一の場所に保管されていたのでは安全対策として完全では無い。計算機システムと隔離されており、かつ天災等にも十分な安全性が望める保管場所を確保する必要がある。この保管場所を当研究所独自で確保するか、既にこのような安全対策を採用している機構を利用するか、方法論は別として絶対に必要な安全対策であることを確認しておかなければならない。

物理的破壊の多くは対策手法がわかっており、その意味では特に議論すべきものは少なく、要はどの程度まで実施して効率と有効性の均衡をはかるかの問題になる。

物理的破壊にまで至らないが、物理的に損害が認識可能な不法行為なども上記と同列と見て対策を講ずることができる。

ここまで述べてきたような意味での安全対策の具体的手法については市販出版物により詳細を知ることができる。なお、最近の情報によれば、計算機システムを対象とした安全基準の法制化が計画されている模様である。これが実現すれば、より明確な指針のもとで安全対策の実施が可能になると思われる。

### 5.2 データベースの安全確保

前節で述べたような安全対策はどちらかと言うと眼に見える損害に対するものであった。計算機システムに独自のものとして、眼にも見えずかつ損害の発生がはっきり把握できない無形財物に対する安全確保の問題がある。よく話題になるコンピュータ犯罪は計算機システム情報の窃取・書換・破壊を行ない利益を得ようとするものである。これは計算機内の情報が無形であるけれども財物価値の高いものであることを示している。計算機システムでは更に犯罪とは言い難いけれども、結果的には情報の財物価

\*) 文献[1] 5.

値を失わせる点で犯罪と大差のない問題がある。過失が原因となって発生する情報の書換・破壊、誤情報入力などは犯罪ときめつけられないが、システムに与える損害の結果は犯罪のそれと同じになる。このような意味での損害防止も安全対策の中に含めておく。

### 5.2.1 システムの完全性

データベース管理システム一般に要求される機能は、安全性 (integrity) と機密保護 (security) に大別される。完全性は更に

- (1) 正当性制約 (integrity constraint)
- (2) 同時制約 (concurrency control)
- (3) 障害回復 (recovery)

の機能に細分化される。

正当性制約は意味的に不当なデータを排除する機能である。数値であるべきところに文字が入ったりすることを排除するのがその例であるが、通常のデータベース管理システムの一般的機能としては限られたものしか持っていない。そのデータベースに個々の正当性制約を完全なものとするには、データ検証用プログラムの作成、人力による確認などの手段を併用しなければならない。翼形入力データの数はプログラムで簡単に検証できるが、データの値がその翼形を正確に表示しているかの確認は人力でなければ難しい。このように人力に負うところが大きく、ソフトウェアだけでは解決できない問題を含んでいる。正当性がデータベースの価値を支配する影響を持つため十分な考慮を必要とする。

同時制約は、複数利用者が同時にアクセスするときデータや検索結果にシステム全体としての矛盾を生じさせない機能である。複数利用者が同時にあるデータを参照するとき、実際にはシステムでの受付順の処理になるため、その間に行なわれたデータ変更により利用者間の結果に差異を生ずることを防止するものである。数値シミュレータ・データベースでは一般利用者にデータ変更権を与えないため、利用者と直接的な関係は少ない。

障害回復は、データベース・システムのハードウェア及びソフトウェアに起因する障害発生があった場合、早期に発生前の状態に復帰させる機能である。障害が及ぼす影響の大小により復帰手法が異なる。

最悪時にはシステム停止・再立上げの措置を必要とする。さらに媒体装置に発生した障害ではバックアップ・ファイルからの再書込など長時間を要する場合も考えられる。何れも運用技術・体制と関係を持つのでそれらの整備も含めて考えるべき問題となる。

完全性はどちらかと言うとシステム作成・運用者側に重要な事項を含む。利用者・データ提供者には機密保護のほうにより関心があると思われる。これについて以下に述べる。

### 5.2.2 機密保護

本来公開を前提とするデータベースにおいて機密保護を論ずるのは何か矛盾があるような気がするかも知れない。しかし、データベース中の情報が不当な目的のために流出したり、情報の消失・変更等により不利益を生ずるような事態を避ける必要は十分にある。そのための安全対策と考えれば、機密保護と言う用語の幾分不当めいた感じは柔らぐ。この意味ではセキュリティとそのままカナ書するほうが当を得ているかも知れないが、本報告では上記の断り書を前提として機密保護とする。

機密保護のためのシステム制御機能としては

- (a) アクセス制御
- (b) 情報フロー制御
- (c) 推論制御
- (d) 暗号制御

があげられる。これらの全てが実在のデータベースにおいて実現している訳では無く研究段階或いは検討段階にしか過ぎないものもある。以下これらを順次説明するが、問題点も同時に指摘しておく。

#### (1) アクセス制御

アクセス制御においては

- (a) ユーザ識別 (user identification)
- (b) ユーザ認証 (user authentication)

が要求される機能である。

ユーザ識別は通常の計算機システムを利用する場合のそれと同じである。これは利用資格の簡単な検査と利用者のシステム内における行動記録の取得を兼ねたものである。通常 ID と呼ばれるシステム内で一意に定まる識別符の登録簿と照合され、不一致ならば以降の入力は拒否される。ユーザ認証は識別関門通過後さらに正当利用者か否かの判定を行なう。

通常パスワードと呼ばれる情報をシステムに入力し、既登録のものとの一致が検査される。パスワード自体はシステム内で一意である必要は無く、利用者自身又はシステム管理者により変更可能な場合もある。何れにしろ両方の検査を通過できなければ以後の操作には入れない。

アクセス制御は、計算機システムへのエントリー、データベースへのエントリー、及びデータベース・ファイルへのアクセス時のそれぞれについて行なうことが出来る。ファイルへのアクセス制御はファイル、レコードの細分化単位ごとに正当な利用者とその権限を規定して行なう。また、アクセス制御は会話型と一括処理の両形態に対して同等に機能させる。

現在のアクセス制御機能は意外に脆いということが内外に発生した幾つかの事件報道により一般にも知られるようになった。IDの附与は、計算機利用統計とか課金を対象とした情報取得を目的に始められた。そのためか処理上の便宜性指向が強く、比較的単純な規則で附与するケースが多いため、機密保護能力は低い。またパスワードも一定の制限下でしか附与できないため保護能力はそれ程強力でない。

IDやパスワードはしょせん利用者の人間個体特性に基いた情報では無いため、機密保護のほんの一部を担っているに過ぎない。

上にちょっと触れたように個体特性情報を基にするアクセス制御として、音声・筆跡・指紋・掌紋などの利用が検討されている。これらの識別技術は計算機の出現以前から研究されており、一部は既に計算機による自動化にも成功し実用に供されている（但し、アクセス制御ではなく別の目的である）。この識別技術の使用により計算機システムの機密保護は非常に強力になると思われるが、コスト面での解決が無い限り一般計算機システムのアクセス制御に利用することは難かしい。また仮にそれが解決したとしても人間の心理面の問題がらみで適用が難かしいものもある。

## (2) 情報フロー制御

アクセス制御が一人の利用者とシステム間の関係を一対一で規制するものに対し、情報フロー制御は複数の利用者間にもそれを延長しようとするものである。間接的手段による情報流出の防止を狙いとする。

ある利用者が参照禁止されている情報を、その情報参照可能な別の利用者に依頼して入手するような手段が考えられる。後者の利用者が前者の所有するファイルへ直接複写するような場合を除き、システムがこのような行為の防止を行なうことは不可能である。考え得る不法行為であるが利用者各人の自覚に待つしか言いようが無い。

## (3) 推論制御

推論制御は参照可能な群情報を用いて、自分が直接参照不能な情報を推論するような利用に対する防御機能である。よく引合に出される例として、個人情報に基に作成された統計データベースを使用しての推論がある。個人情報の直接参照が禁止されていても、検索条件の組合せや既知情報との結合による個人情報推定の可能性が知られている。この例では防御機能の必然性は高度に要求される。しかし数値シミュレータ・データベースでは研究・技術開発支援を目的とするので、それらの進行に必要な推論の有意性を否定できない。この意味での推論のためか、参照禁止情報の推論のためなのかを確証をもって判定することは困難がある。システムで推論を行なっている利用者を捕捉することは可能であるが、その意図を判別することはできない。

## (4) 暗号制御

暗号制御は格納されている情報或いは伝送中のそれを入手したとしても、その内容の機密が保たれるよう暗号技術を使用するものである。暗号は本来、伝送中の情報が正当な受信者以外に洩れ不利益が起るのを防止するため古代より使用されている。計算機システムにおいては、システム内で稼動中及びバックアップ保管中のファイルからの情報流出も考慮に入れなければならない。媒体自体の盗難、媒体間の複写機能を使用しての情報窃取などの犯罪例が既に発生している。情報が暗号化され、その強度が高い程解読のコスト高や情報品質の低下（たとえば陳腐化）による犯行側の損失が増す。暗号化は、こうした不法行為の収支不均衡を莫大化させることを狙ったもので、絶対解読不能を目的とするものではない。

現実のデータベースでは記憶媒体情報の暗号化を標準仕様に持つものは無い。その理由は復号（解読）

時のソフトウェア・オーバーヘッドが大きいのである。従って稼働中の媒体内情報の暗号化は当面見送らねばならないが、バックアップ用のそれらには適用しておくことが望まれる。ダンプ時に暗号化、復帰時に復号化を行なうことにより機密保護が達成できる。

通信回線経由の端末に対しては、ハードウェアによる暗号伝送が実用になっている。送受信端のそれぞれにおいて通信制御装置とモデム間に回線暗号装置を挿入する。DES(Data Encryption Standard, 米国連邦情報処理標準)に準拠した製品が内外メーカの数社から発売されている。もちろん標準であるから暗号アルゴリズムは公開されており、秘密は暗号鍵に限定される。DESの場合、暗号化・復号化は同一鍵だから鍵の管理は厳重でなければならない。ただし端末ごとに鍵が異なるので、窃取した鍵はその端末でしか使用できない。

通信回線経由の端末における問題は盗聴のほかにも存在し、むしろこの方がより危険と言えることがある。近年のようにハードウェアの著しい価格低下によって、フロッピーディスク付のマイクロ・コンピュータによる端末構成が容易になった。受信情報をフロッピーディスクに格納したほうが受信効率の良い場合も稀ではない。この段階で情報は復号されているので、フロッピーディスクを窃取又は復写するほうが盗聴より遥かに安価な情報入手法となる。この防止策にシステム側は全く関与できず、端末利用者の管理に頼るのみである。

以上、機密保護に関して種々のことを述べたが、機密維持の究極はシステム管理・利用の両者の責任に負うところが大きいと言える。

## 6. む す び

数値シミュレータ計画立案の過程の中で、データベース構築の必要性が強く認識された。当初より構築の困難さは予想されたが何分素人集団の中ではその実体が十分に把握・認識できていない点が多々あった。数値シミュレータ実現の機運が進展するのに併せて、データベース構築を本格化させるため事前調査と設計作業の実行動に入った。本報告では「データベース概念設計書、富士通㈱、昭和59年3月」

に終結した作業に関して、結果よりも背景に重点を置いて検討経過を述べた。検討範囲が広大なため詳細にわたって記述し切れなかった点が多いが、今後構築作業の進展に伴ないその補完が実現してゆくものと思う。本報告では、上述のデータベース構築における困難の実体把握とその対処手法についての見通しを得たということをもって結論とする。

本報告の作業は冒頭「はじめがき」中に記した通り、現計算センタ長 三好 甫技官の手により始められた。その後続作業を現筆者が行ない上述の仕様書にまとめた。困難ではあろうが将来像の豊かな数値シミュレータ・データベース構築への参加と、全面委任による作業進行を認めてもらったことにより現筆者には新世界を望見する興味と昂奮の引続いた作業であった。

なお、設計書まとめ段階ではファコム・ハイタック㈱・ファコム本部システム三部の猪狩研一・小西秀之・村田総一郎各氏に参加してもらった。

本報告をとじるにあたり、上記をもって謝意に代える。

## 7. 参考文献

数値シミュレータ・データベースの企画から設計までの段階で種々の文献を参照した。本報告記述に際しそれらの引用を個別に示さなかった。致るところで交叉的にその影響を受けたため、どの部分がどの文献からの恩恵かを指摘することがもはや難しくなってしまったのがその理由である。よって通常の報告書形式とは異なるが、それらを列記して謝意に代える。

### [1] データベース技術

1. 「情報処理」；情報処理学会刊（毎月）  
同学会誌の第23巻第10号「大特集：データベース技術」を始めとし、その他の号に掲載の解説記事、個別分野の特集記事から得たものは非常に多い。
2. 上林弥彦, Kim, W., 酒井博敬編；「最近のデータベース・システムとその応用」bit別冊, 共立出版, 1984.
3. Various, "Development and Use of Numerical and Factual Data Bases", AGARD-

- LS-130, 1983.
4. Various, "NASA Administrative Data Base Management Systems", NASA CP 2254, 1983.
  5. See, M.K; and Cozzolongo, J.V., "A Common Geometric Data-Base Approach for Computer-Aided Manufacturing of Wind-Tunnel Models and Theoretical Aerodynamics Analysis", NASA TP 2151, 1983.
  6. Heerema, F.J. and van Hedel, H., "An Engineering Data Management System for Computer Aided Design", NLR MP 82050 U, 1982.
- [2] 航空・宇宙飛しょう体の研究・設計・製造システム
1. George, M.W., Weden, B.W., and Johnson, R.R., "Computational Wing Design for an Advanced Trainer", AIAA-83-0093, 1983.
  2. Kubasak, I.J., "Design Engineering", AIAA-80-1880, 1980.
  3. Narramore, J.C. and Yeary, R.D., "Airfoil Design and Analysis Using an Information Systems, Approach", AIAA-80-1444, 1980.
  4. Feder, A., "Test Results on Computer Graphics Productivity for Aircraft Design and Fabrication", AIAA-75-0967, 1975.
  5. Middleton, W.D. and Lundy, J.L., "A System for Aerodynamic Design and Analysis of Supersonic Aircraft", Part 1 — General Description and Theoretical Development, NASA CR 3351, 1980. Part 2 — User's Manual, NASA CR 3352, 1980. Part 3 — Computer Program Description, NASA CR 3353, 1980. Part 4 — Test Cases, NASA CR 3354, 1980.
  6. Morrison, R.A., "Design for Quick Reaction Aircraft Modification", AIAA-80-1887, 1980.
  7. Smyth, S.J., "CADAM Data Handling from Conceptual Design Through Product Support", AIAA-79-1846, 1979.
  8. Smyth, S.J., "CAD-The Designer's New Tool", AIAA-81-0850, 1981.
  9. Stack, S.H., "A Computer-Aided Design System Geared Toward Conceptual Design in a Research Environment", AIAA-81-0372, 1981.
  10. De Jarnett, Kamà, F.A., and Chitty, A., "Aerodynamic Heating and Surface Temperatures on Vehicles for Computer-Aided Design Studies", AIAA-83-0411, 1983.
  11. Billig, F.S., White, M.E., and Van Wie, D.M., "Application of CAE and CFD Techniques to a Complete Tactical Missile Design", AIAA-84-0387, 1984.
  12. Bergsten, F.C., "CAD/CAM and Analysis: A Production Tool for Spacecraft Design", AIAA-83-0218, 1983.
  13. Wehrman, M.D., "CAD Produced Aircraft Drawings", AIAA-80-0732, 1980.
  14. Harris, D.H.W., "Applying Computer Aided Design (CAD) to the 767", *Astrodynamics & Aeronautics*, Vol. 18, 1980, pp. 44-49.
  15. Dror, B., "Computer-Aided Design at Israel Aircraft Industries", *Computer and Graphics*, Vol. 3, 1978, pp. 93-105.
  16. Clark, R.H., "Some Aspects of Computer Aided Aircraft Design", *The South African Mech. Eng.*, Vol. 31, 1981, pp. 8-12.
  17. Folck, J.L., "Interactive Graphics in Preliminary Design of Air Vehicles", *SAWE Paper No. 1243*, 1978.
  18. Ridenour, R.W., "Computer Aided Technology Interface with Weights Engineering", *SAWE Paper No. 1346*, 1980.
  19. Stack, S.H., Edwards, C.L.W., and Small, W.J., "GEMPAK: An Arbitrary Aircraft Geometry Generator", NASA TP 1022, 1977.
- [3] コンピュータ・グラフィックス
1. Gregory, T.J. and Ashbaugh, J., "Workshop on Aircraft Surface Representation

- for Aerodynamic Computation”, NASA TM 81170, 1980.
2. Purser, K., “Interactive Computer Graphics”, AIAA-80-1889, 1980.
  3. Talcott, Jr. W.A., “Computer Graphic Display Technique for the Examination of Aircraft Design Data”, AIAA-81-0370, 1981.
  4. Edwards, C.L.W., Meissner, F.T., and Hall, J.B., “The Use of Computer-Generated Color Graphic Images for Transient Thermal Analysis”, NASA TP 1455, 1979.
  5. Various, “Applications of Computer Graphics in Engineering”, NASA SP-390, 1975.
  6. Shoosmith, J.N. and Fulton, R.E. (Compiled), “Computer-Aided Geometry Modeling”, NASA CP 2272, 1984.
  7. Newman, W.M. and Sproull, R.F., “Principle of Interactive Computer Graphics”, 2nd Ed., McGraw-Hill, 1979.
  8. 山口富士男; “コンピュータディスプレイによる図形処理工学”, 日刊工業新聞社, 1981.
- [4] 安全対策
1. 通産省監修; 「電子計算機システム安全対策基準解説書」, (社) 情報サービス産業協会, 1984.
  2. 情報処理学会; 「特集: 情報セキュリティ」, 情報処理, 第 25 巻, 第 6 号, 1984.
- [5] その他
1. 計算センタ内部資料; 「データベース概念設計書」, 富士通株式会社, 昭和 59 年 3 月。  
なおこの資料には, 「データベースコマンド解説書」が別冊として含まれている。

---

## 航空宇宙技術研究所報告 869号

昭和60年8月発行

発行所 航空宇宙技術研究所  
東京都調布市深大寺東町7丁目44番地1  
電話武蔵野三鷹(0422)47-5911(大代表)〒182

印刷所 株式会社 東京プレス  
東京都板橋区桜川2-27-12

---



**Printed in Japan**