

JAERI - M

87-097

JT-60全系実験データベース構築(I)

—— 基本構想とシステム設計 ——

(臨界プラズマ試験装置設計報告・195)

1987年7月

栗原 研一・伊藤 康浩*・木村 豊秋

JAERI-Mレポートは、日本原子力研究所が不定期に公刊している研究報告書です。
入手の問合わせは、日本原子力研究所技術情報部情報資料課（〒319-11茨城県那珂郡東海村）あて、お申しこしてください。なお、このほかに財団法人原子力弘済会資料センター（〒319-11茨城県那珂郡東海村日本原子力研究所内）で複写による実費頒布をおこなっております。

JAERI-M reports are issued irregularly.

Inquiries about availability of the reports should be addressed to Information Division
Department of Technical Information, Japan Atomic Energy Research Institute, Tokai-
mura, Naka-gun, Ibaraki-ken 319-11, Japan.

©Japan Atomic Energy Research Institute, 1987

編集兼発行 日本原子力研究所
印 刷 (株)高野高速印刷

J T - 6 0 全系実験データベース構築 (I)
—基本構想とシステム設計—
(臨界プラズマ試験装置設計報告・195)

那珂研究所 J T - 6 0 試験部 J T - 6 0 第1試験室
栗原 研一・伊藤 康浩*・木村 豊秋

(1987年6月11日受理)

J T - 6 0 の放電結果データを解析するには、その環境の整備が必要不可欠である。これを実現するため、放電結果データをオンライン通信により大型汎用計算機に転送し、そこでデータベース化することとした。

このデータベースは、データファイルと利用者との間の各種インターフェイスを整備する体系とし、データベースの信頼性を確保する一方、利用者からはファイルの構造を意識することなく、かつ I / O 回数・CPU 時間を極力短くすることで実運用に耐え得るツールとした。また、図形表示するための支援機能も整備し簡便な図形処理を可能にした。

本冊子では、特に基本構想とシステム設計について報告する。

JT-60 Database System (I)
— Basic Concept and System Design —

Kenichi KURIHARA, Yasuhiro ITOH* and Toyoaki KIMURA
Department of JT-60 Facility
Naka Fusion Research Establishment
Japan Atomic Energy Research Institute
Naka-machi, Naka-gun, Ibaraki-ken

(Received June 11, 1987)

Naturally, sufficient software circumstance makes it possible to analyse the discharge result data effectively. JT-60 discharge result data, collected by the supervisor, are transferred to the general purpose computer through the new linkage channel, and are converted to "database". Datafile in the database was designed to be surrounded by various interfaces. This structure is able to preserve the datafile reliability and does not expect the user's information about the datafile structure.

In addition, the support system for graphic processing was developed so that the user may easily obtain the figures with some calculations.

This paper reports on the basic concept and system design.

KEYWORDS: JT-60, Database, Basic Concept, System Design, Discharge Result

* I.B.S. Data Center Co. Ltd.

目 次

1. はじめに	1
2. 基本構想	1
2.1 全系実験データベース製作の契機	1
2.2 全系フロンテンド計算機 (FEP) の結合における問題と対策	3
3. 全系実験データベース基本設計	4
3.1 データベースの構成	5
3.2 実験データの構造	7
3.2.1 実験データのモデルとファイル構造	7
3.2.2 実験データの物理構造と検索方法	9
3.2.3 実験データの表現形式	13
3.2.4 実験データの変換, 編集処理	14
3.2.5 開発言語について	15
3.3 利用者の資格検査	15
3.4 データ取得ツールと参照支援ツール	19
3.5 運用管理支援ツール	21
4. システム設計	24
4.1 全系・FEP 通信とその周辺	24
4.1.1 全系側の送信・受信処理とその周辺	25
4.1.2 FEP 側の送信・受信処理とその周辺	28
4.2 データベース創成	31
4.2.1 創成処理の起動	31
4.2.2 データ変換・編集処理	31
4.3 汎用データハンドリングパッケージと資格検査	35
4.4 図形処理支援機能	37
4.5 運用管理機能とデータベース結合・追加・削除機能	41
4.5.1 運用管理機能	41
4.5.2 データベース結合・削除・追加機能	42
5. あとがき	44
謝 辞	45
参考文献	45
付録 1. データモデル	46
付録 2. 何故汎用的なデータベースを使用しないという結論を出したか。	48

Contents

1. Introduction	1
2. Initial Consideration	1
2.1 Needs to Build Up the Database System	1
2.2 ZENKEI [*] -FEP(Front End Processor) Linkage	3
3. Basic Concept	4
3.1 Database Structure	5
3.2 Discharge Result Data Structure	7
3.2.1 Model and File Structure	7
3.2.2 Physical Structure and Referring Method	9
3.2.3 Data Format	13
3.2.4 Creation, Conversion and Edition	14
3.2.5 Language for coding	15
3.3 User Identification Check	15
3.4 Tool for Handling the Data	19
3.5 Support Tool for Database Operator	21
4. System Design	24
4.1 ZENKEI-FEP Communication	24
4.1.1 Communication Procedure in ZENKEI	25
4.1.2 Communication Procedure in FEP	28
4.2 Database Creation	31
4.2.1 Method for Creation Start	31
4.2.2 Data Conversion and Edition	31
4.3 Data Handling Package	35
4.4 Support System for Graphic Display	37
4.5 Support Tool for Operator, and Combine, Addition and Elimination of Database	41
4.5.1 Support Tool for Operator	41
4.5.2 Combine, Addition and Elimination Database	42
5. Concluding Remarks	44
Acknowledgment	45
References	45
Appendix 1 Database Model	46
Appendix 2 Why didn't we adopt the general purpose database?	48

* ZENKEI; JT-60 Supervisory computer system

1. はじめに

JT-60が完成して半年ほど経ち、装置の製作に係わる残作業やトラブル対策も少し一時の慌しさから解放されようとしている時期に、OH(I)実験の実験結果を眺めて見る場合を想定しよう。JT-60の実験データの主要なものは、全系制御設備（以下「全系」と略す）に集められ画面に表示される。従って、それを見て必要となれば画面のハードコピーを取れば眺めることはできる。しかし、全系は記憶容量の点の制約から過去4ショット分のデータしか保持することができない。そのため、少し前のデータを見ようとするとき磁気テープを全系内に再記録して見ることになる。この際の手間は、まず磁気テープを探して、その磁気テープをかけ、再記録するまで、最短でも約20分はかかる。この20分は、実験結果を見ようとする人それぞれに対し、ショット毎に必要な時間であり、20分/人/shotである。これは大変な労力であり、毎日行うショットの数が約25ショットであることを考えると、1人の人が前日のショットを見るために8時間以上もかかることを意味し、現実的には解析作業が不可能な状態と行うことができる。そこで打策としては、大型汎用計算機へデータを移行し、それらの計算機にデータ管理をさせることを考えた。大型計算機は多くの端末を持っており多くの人が同時にデータを利用することができる。また高度なデータ処理ツールを用いることにより様々な解析を可能にする能力を持っている、等々、データの解析にとって有利な環境を大型計算機は持っていると言える。しかし、ただデータをそのまま漫然と入れておくだけでは利用する者に数多くの負担を強いることになる。そこでデータベースという概念を導入し、多くの人にとって共通の基本データであるという前提で、人間とのインターフェイスが充分とれたデータを構築することを考えた。

本報告は、全系実験データベースの構築に関わる、基本構想からシステム設計に至るまでの部分について行う。詳細設計については、全系実験データベース構築(II)³⁾において述べることにする。

2. 基本構想

大規模なソフトウェアの設計・製作に際して、まず、その基本となる考え方を明確にしておくことが必要である。特に事の発端からの論理の展開に注意を払うことによって、将来如何なるトラブルが生じようとも、直ちにこの基本に立ち戻って考え直しができるようにしておく、そうすることでシステムは常に最適な形へと向うものと思われる。本章には、1章でも簡単に触れたが、何故全系実験データベース（以下「全系DB」と略す場合がある）を作ることにしたかという点について述べるとともに、そこに内在する問題点と対策、さらに通信とデータベースの両方の基本設計について述べることにする。

2.1 全系実験データベース製作の契機

全系実験データベースを作るにあたっては、現状にデータ解析上の不満を感じたことに端を発している。それは、全系の能力と関係している。

1. はじめに

JT-60が完成して半年ほど経ち、装置の製作に係わる残作業やトラブル対策も少し一時の慌しさから解放されようとしている時期に、OH(I)実験の実験結果を眺めて見る場合を想定しよう。JT-60の実験データの主要なものは、全系制御設備（以下「全系」と略す）に集められ画面に表示される。従って、それを見て必要となれば画面のハードコピーを取れば眺めることはできる。しかし、全系は記憶容量の点の制約から過去4ショット分のデータしか保持することができない。そのため、少し前のデータを見ようとするとき磁気テープを全系内に再記録して見ることになる。この際の手間は、まず磁気テープを探して、その磁気テープをかけ、再記録するまで、最短でも約20分はかかる。この20分は、実験結果を見ようとする人それぞれに対し、ショット毎に必要な時間であり、20分/人/shotである。これは大変な労力であり、毎日行うショットの数が約25ショットであることを考えると、1人の人が前日のショットを見るために8時間以上もかかることを意味し、現実的には解析作業が不可能な状態と言うことができる。そこで打策としては、大型汎用計算機へデータを移行し、それらの計算機にデータ管理をさせることを考えた。大型計算機は多くの端末を持っており多くの人が同時にデータを利用することができる。また高度なデータ処理ツールを用いることにより様々な解析を可能にする能力を持っている、等々、データの解析にとって有利な環境を大型計算機は持っていると言える。しかし、ただデータをそのまま漫然と入れておくだけでは利用する者に数多くの負担を強いることになる。そこでデータベースという概念を導入し、多くの人にとって共通の基本データであるという前提で、人間とのインターフェイスが充分とれたデータを構築することを考えた。

本報告は、全系実験データベースの構築に関わる、基本構想からシステム設計に至るまでの部分について行う。詳細設計については、全系実験データベース構築(II)³¹において述べることにする。

2. 基本構想

大規模なソフトウェアの設計・製作に際して、まず、その基本となる考え方を明確にしておくことが必要である。特に事の発端からの論理の展開に注意を払うことによって、将来如何なるトラブルが生じようとも、直ちにこの基本に立ち戻って考え直しができるようにしておく、そうすることでシステムは常に最適な形へと向うものと思われる。本章には、1章でも簡単に触れたが、何故全系実験データベース（以下「全系DB」と略す場合がある）を作ることにしたかという点について述べるとともに、そこに内在する問題点と対策、さらに通信とデータベースの両方の基本設計について述べることにする。

2.1 全系実験データベース製作の契機

全系実験データベースを作るにあたっては、現状にデータ解析上の不満を感じたことに端を発している。それは、全系の能力と関係している。

全系制御設備計算機システム(16 bit ミニコンピュータ HIDIC-80E が7台で構成、以下「全系計算機」と略す)において収集・蓄積されている情報は、放電に関する制御情報だけにとどまらず、本体、電源など各設備からの機器情報、運転操作情報など広範囲にわたっている。この情報は、1ショットあたり3メガバイト(MB)を超え、仮に1日25ショットの放電を行なった場合、計75MBという膨大な量となる。ところが全系計算機は、制御用計算機という性格上オンライン制御を中心とした処理に関してサポートし、汎用的なデータ解析処理を実行することについては、もともと考慮されていない計算機である。

この点に関して、第1章で述べた以外の実際の具体例をとってみると以下のようなになる。

- ・計算機主記憶容量が小さく、現状でとれる最大論理コアブロック領域としては、約30Kワード(論理空間の最大値では64Kワード)しかとることができない。
- ・既存の制御用コードの保護を行なう為、磁気ディスク装置にプロテクトがハード的にかけられている。このため新規にプログラム登録を実施する場合、このプロテクトピンの解除を行なう必要が生じるが、この時の操作を誤るとシステムの一部もしくは、全体を破壊する可能性がある。
- ・計算機に対する入力手段としては、パンチカードが主体となり、手間がかかる。またカード読み取り装置の処理速度も非常に遅い。
- ・応用プログラムの開発に際し、計算機使用時間が実験終了後でないに行なうことができない。

また、上記全系計算機の機能で対応不可能な処理については、計測データ処理設備(大型汎用計算機 FACOM M-360 2台で構成)との間でオンライン通信を行ないデータ処理設備側で処理することを考え放電実績データの転送処理など全系計算機-データ処理計算機間リンケージをとってきた。

しかし、これから新たに考えられる処理方法の開発、また、本来ショット間定型処理^{*}に乗らない処理までもデータ処理計算機にのせるのは、既に負荷が高いことや運用上なじまないこともあり不可能と言える。この為、別の大型汎用計算機との結合をすることが必要である。

全系計算機で収集した情報はすべて磁気テープへ退避するという方法がとられていた。しかし、この磁気テープは6-7 Shotに1本の割合で消費され、その掛け換えに係わる労力は決して少なくない。

また最終的に磁気テープに退避するとしても、全系計算機の低密度(1600 BPI)記録より高密度(6250 BPI)にして、磁気テープの本数を減らすことも保管場所の点から重要である。

このように、実験データの統計解析処理、物理・工学的解析処理を行なうためには、それなりの環境が与えられなければならない。しかも、放電終了後、速かにデータを見るためには全系計算機とオンライン通信を行う大型汎用計算機に実験データを転送して統一的運用管理を実現することが必要となる。

一方時期を同じくして、那珂研究所にフロント・エンド・プロセッサ(以下「FEP」と略す)を設置する計画が具体化することになり、データベース構築の対象大型汎用計算機をFEPとすることとした。

実は、このような機能の必要性は、オフライン的に又はオンライン的にJT-60の解析を行うものとしてJT-60詳細設計の段階で“高レベル計算機”という名で提案されていた。この高レベル

* ショット間に行う処理方法が確立したデータ処理

計算機は、J T - 6 0 のデータを受信し、解析して、次の条件を作成するという今考えてもかなり意欲的なことを実行する計算機であった。この計算機は、予算の関係で実現しなかったが、この考え方は、計測設備に属するデータ処理設備に受け継がれた。ところが、このデータ処理設備は、計測装置のデータ操作だけで当時既に 1 0 0 % に近い負荷率を示しており、とても解析などの余裕が無いことから、高レベル計算機構想は現実とならないままとなっていた。しかし、ここに来てやはり実験上解析機能は不可欠との認識から浮かび上って来たと言うこともできる。

2.2 全系—フロントエンド計算機 (F E P) の結合における問題と対策

全系— F E P を結合することで、現状の不满はほぼ全て解消されると考えられるが、一方それを実現するのに問題が無い訳では無い。本節では、その問題点を提起し合わせて対策を示す。

まず、全系計算機から送られた放電制御実験結果データを F E P 上に展開するにあたり、全系計算機内でのデータの特殊性についての問題がある。

(1) 放電制御実績結果データのアクセスに関する特殊性

全系計算機の内部におけるデータのリンケージは、すべて P I D (Point IDentification) 番号と呼ばれる固有のキーにより行なわれているという特殊性を有している。この為、放電制御実績結果データを参照するには、P I D 番号とデータが格納されているアドレスを対応させる“管理定数テーブル”と呼ばれるテーブルが必要となり、このテーブルがなければ放電制御実績結果データの情報を一切参照することができない。

また、全系計算機のシステム構成の変更— P I D 番号の追加・削除—に伴い、“管理定数テーブル”の内容が変更されることになり、放電制御実績結果データの検索に際して本テーブルとのバージョンが一致していることが必要になる。

しかしこの点については、F E P 側での管理・運用方法及び、データ構成を合理的に構築することにより解決可能である。

(2) データ表現形式の相異

全系計算機は 1 6 ビット計算機であり、通常の内表現は 2 バイトを基準にし、固定小数点、浮動小数点などの他に、指数仮数表現、ビット表現などの特殊な表現形式がとられている。ところがこれに対して、F E P は 3 2 ビット (4 バイト) が基準となった表現になっているため、全系計算機から伝送されたままの状態では使用することができない。特に通常の演算処理においてビット形式のデータを自由に操作する機能が十分でない。またこのような環境の下で、これらの変換操作を一般のユーザーが逐一行なうことは、データの利便性や信頼性等を考慮すると決して最良の方法ではない。

この為、特殊な形式の数値情報をユーザーの扱いやすい通常の形式 (4 バイト、8 バイト整数、8 バイト実数等) に予め変換しておくことが必要になる。

次に、新規導入が予定されている F E P は、計算センター運用の汎用計算機という位置付けのため、データ処理設備と同じ考え方で計算機間通信を議論することはできないことがある。このため次のような点を考えなくてはならない。

(3) F E P が汎用機であることによる特徴

F E P においては、不特定多数のユーザーが大小さまざまな処理を行っており、通常のバッチジョブの他に、5 0 台以上の T S S 端末をサポートすることになると考えられる。ところが、全系計算

機からは FEP の負荷の大小にかかわらず、大量な情報を伝送してくる。この為、FEP の伝送情報の入力処理が滞ると、全系計算機への入力完了信号の発行が遅れ、これに伴い JT-60 の放電サイクルに遅れを生じさせてしまう可能性がある。これは、FEP 側のチャンネルの I/O に関する処理のプライオリティーが、一律に設定されており、全系計算機との I/O の部分だけを優先することができないことに起因している。

現在のところ、FEP 側の負荷がどの程度まで上がるのかを具体的な数値で表現することはできないが、実際の運用試験の段階で通信時間がオーバーするようになることになれば、全系計算機側の送信バッファ容量の拡張や実行クラスの優先割り当てなどで対応する考えである。

(4) FEP のディスク専用化

伝送されてきた情報は、先にも述べた様に 3 メガバイト/ショットと膨大な情報量となっている。この大量の情報を効率よく格納し、データの保全性を向上する為、FEP の記憶装置（磁気ディスク、大容量記憶装置（CLS^{*}））の専用割り当てなどの処置が必要になる。

3. 全系実験データベース基本設計

今まで、“データベース”という言葉を使ってきたが、本節で、その意味する内容について明確にする。

そもそも実験データを用いて我々が目指すものは、

- ① プラズマ特性の解明
- ② 制御手法の確立
- ③ 各種ハードウェアの特性の解明

等々色々有る。従ってそれら目標に応じた各種解析処理があり、それら全てに対して便利な機能を提供する必要がある。その前提としては、データがある所定の形に変換し、さらにそれらを取得するための簡便なサブルーチン群を準備する。一方、統一的なデータ管理を行うと共にデータの品質を維持するためのツールも用意することが必要となろう。最近、この種のシステムを“データベース”と言うそうなのでここでも、そのように呼ぶことにするが、一口で“データベース”と言っても様々な種類があると思われるので、我々の扱うデータに適したデータベースとはどのようなものになるか、ということを考えておかねばならない。データベースの目的としては、我々の場合、JT-60 の実験データの品質を保持し、各種解析処理を行うユーザに迅速にデータを提供することである。また、データベースが一般的に持っている特徴は、次のことが言える。

(1) データの冗長性を最小限にする

利用者がデータを各自で保管すると、複数の人が同一のデータを別々に保管することがあり、重複している分だけ余分に記憶媒体が必要になるが、データベースで1つにまとめて管理することにより、冗長性を最小限にすることができる。

(2) 保全性

データをまとめて保守管理することにより、データの紛失が防げる。また、データの退避を行うこ

* CLS : Cartridge Library System

機からは FEP の負荷の大小にかかわらず、大量な情報を伝送してくる。この為、FEP の伝送情報の入力処理が滞ると、全系計算機への入力完了信号の発行が遅れ、これに伴い JT-60 の放電サイクルに遅れを生じさせてしまう可能性がある。これは、FEP 側のチャンネルの I/O に関する処理のプライオリティーが、一律に設定されており、全系計算機との I/O の部分だけを優先することができないことに起因している。

現在のところ、FEP 側の負荷がどの程度まで上がるのかを具体的な数値で表現することはできないが、実際の運用試験の段階で通信時間がオーバーするようなことになれば、全系計算機側の送信バッファ容量の拡張や実行クラスの優先割り当てなどで対応する考えである。

(4) FEP のディスク専用化

伝送されてきた情報は、先にも述べた様に 3 メガバイト/ショットと膨大な情報量となっている。この大量の情報を効率よく格納し、データの保全性を向上する為、FEP の記憶装置（磁気ディスク、大容量記憶装置（CLS^{*}））の専用割り当てなどの処置が必要になる。

3. 全系実験データベース基本設計

今まで、“データベース”という言葉を使わずに、断わり無しに使って来たが、本節で、その意味する内容について明確にする。

そもそも実験データを用いて我々が目指すものは、

- ① プラズマ特性の解明
- ② 制御手法の確立
- ③ 各種ハードウェアの特性の解明

等々色々有る。従ってそれら目標に応じた各種解析処理があり、それら全てに対して便利な機能を提供する必要がある。その前提としては、データがある所定の形に変換し、さらにそれらを取得するための簡便なサブルーチン群を準備する。一方、統一的なデータ管理を行うと共にデータの品質を維持するためのツールも用意することが必要となろう。最近、この種のシステムを“データベース”と言うそうなのでここでも、そのように呼ぶことにするが、一口で“データベース”と言っても様々な種類があると思われるので、我々の扱うデータに適したデータベースとはどのようなものになるか、ということを考えておかねばならない。データベースの目的としては、我々の場合、JT-60 の実験データの品質を保持し、各種解析処理を行うユーザに迅速にデータを提供することである。また、データベースが一般的に持っている特徴は、次のことが言える。

(1) データの冗長性を最小限にする

利用者がデータを各自で保管すると、複数の人が同一のデータを別々に保管することがあり、重複している分だけ余分に記憶媒体が必要になるが、データベースで1つにまとめて管理することにより、冗長性を最小限にすることができる。

(2) 保全性

データをまとめて保守管理することにより、データの紛失が防げる。また、データの退避を行うこ

* CLS : Cartridge Library System

とにより、ハードウェアの故障や種々の事故に対処できる。

(3) 機密性

データは、様々な情報を含み、多大な価値を持っているため、許可されていない人によるデータの参照、修正あるいは破壊を行えないようにデータを保護する必要があるが、データベースでは、許可された人のみデータを入手できるように管理することができる。

(4) 柔軟性と迅速性

利用者は、データについて様々な角度から分析すると考えられるが、データベースではデータを柔軟かつ迅速に検索することができる。

(5) データとプログラムとの独立

データあるいはその記憶媒体は、将来、様々な変化していくと考えられるが、これに伴ってデータベースを利用している応用プログラムを変更していると、そのために法外な費用がかかる。データベースでは、応用プログラムから見たデータは、データの物理的表現とは切り離された形式になり、物理データと応用プログラムとの独立を保つことができる。

以上の目的に適い、且つ特徴を兼ね備えたデータベースとはどのようなシステムになるか？以下に述べていくことにする。

3.1 データベースの構成

J T - 6 0 のデータは、ショット番号で識別される数多くのデータ集合から成っている。さらにそれぞれのデータ集合は、数多くの計測点のデータから成っている。実験の進行に伴ないデータ集合は単調増加して行くものであるが、それらを一つ一つ適正に管理しなくてはならない。また、利用者が簡単にデータ集合にアクセスできなくてはならないし、運用管理者は、データ集合の状態把握やデータ修正などの作業が、簡単かつ間違い無くできなくてはならない。このようにデータ集合を中心にそれらを色々な形で取扱うインターフェイス群がデータベースには必要不可欠である。逆の言い方をすれば、インターフェイス群が無ければ、データベースと言うものにはなり得ず、単なるデータファイルであり、このインターフェイスこそがデータベースたる由縁と考えられる。このインターフェイスはデータ集合を完全に包み込み外部との情報授受を統括的に制御するものであることが必要となる。この各種インターフェイスは3つに大別することができる。即ち、①データベースの創成、②利用者用インターフェイス、③運用管理者用インターフェイス の3つである。

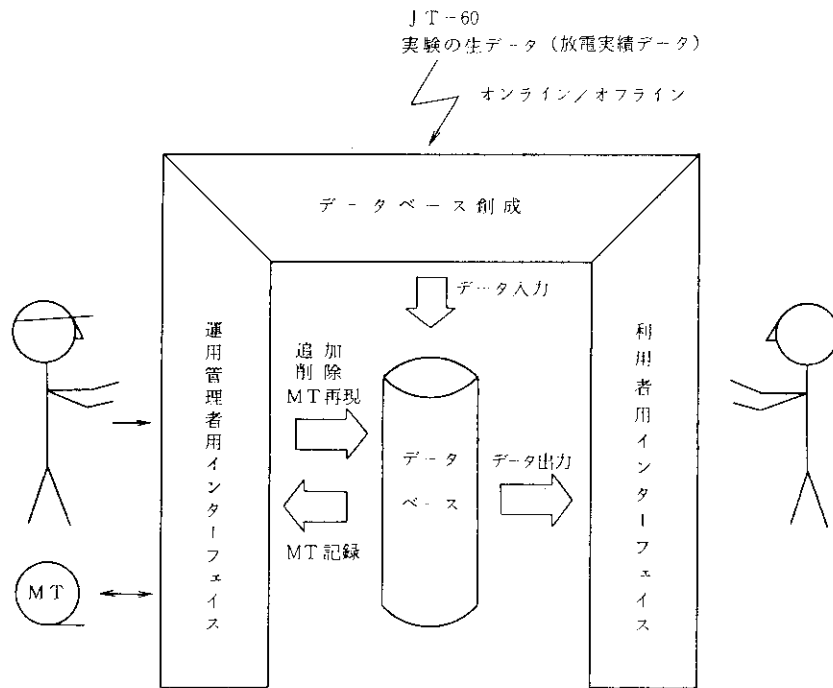
データベースの創成は、J T - 6 0 で実験した生データをオンラインで通信するか、オフライン媒体（たとえば、磁気テープ（MT））で移行するかした後、データベースとしての管理に乗るようなフォーマットへ変換する。その際、ショット毎に、実験データの内容が変わる場合もあるので、ショット単位で自己完結性（そのショットの内容を規定する各種情報（たとえば、略称）は、そのデータに付随していなければならないといったもの）がとられていなくてはならないことは注意すべき点となろう。システム設計の章でも述べるが、オンライン通信との取合いをどうするかというところは、運用ともからめて重要な点であるし、又、プログラムの起動という汎用計算機にとっては通常の使い方ではない方法も用いなければならないことも考えておく必要がある。

利用者用インターフェイスとは、データベースを利用するのに最小限の時間と労力で済むように配慮した上で、かつデータベースの機密性、安全性も保持できるものでなくてはならない。安全性とい

う意味では、サブルーチンの形で利用者に提供し、直接利用者のプログラムからデータベースにアクセスできないようにすることが良いであろう。又、機密性の観点から利用者の資格を検査することも必要となる。一方で、CPU時間や入出力回数といった計算機資源消費を最小にする必要性もある。また忘れがちではあるが極めて重要な項目として、運用管理者用インターフェイスとの競合をすることなく同時に両インターフェイスが動作することを想定しておかなくてはならない。何故なら、データの内容に不備があってデータを一部修正したり削除したりすることは通常の時間帯に十分想定される事態である一方で利用者に対し使用不可能な時間を作るのは公開されたシステムとしては望ましいことではないから、利用者の作業と運用管理者の作業とが並行して行なわれる必要があるからである。

運用管理者用インターフェイスは、データベースの番人である運用管理者の作業を円滑にするものである。運用管理者が管理しなければならないものとしては、データベースそのもの、及び周辺プログラム一式など様々であり、データベースに想定される各種変更や現状把握などの作業を支援する機能を有していなくてはならない。また、最終媒体としてのMTデータを記録することもできなくてはならない。

以上の概念図を第3.1.1図に示す。データベースを囲むインターフェースを総称してデータベース管理システム（DBMS；Data Base Management System）と呼ぶ。



第 3.1.1 図 全系実験データベースの構成概念図

3.2 実験データの構造

実験データの構造を検討するためには、その発生形態、種類、量などの特徴に留意する必要がある。実験データは、実験を実行するために必要な条件、いわゆる放電条件と実際の放電に伴ない各種計測機器で検出している計測データとに分けられる。

この実験データの特徴は、次のとおりである。

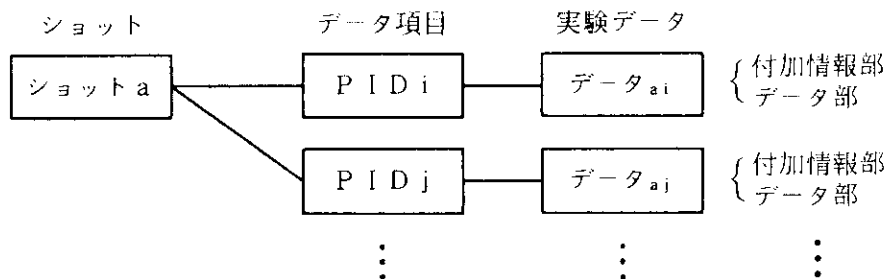
- ① 実験は、パルスのに行なわれるので実験データは、1つの放電（ショット）に従属する。
- ② 発生したデータは、複数の設備で一旦収集され、その後全系へ一括収集される。
- ③ 管理の最小単位であるデータ項目は、数千種類想定され、非常に多種類に及ぶ。
- ④ 個々のデータ項目は、発生時点では多様な形式をもつが、その性質から分類すると、一点データおよび1～4次元の時系列データになる。
- ⑤ 個々のデータ項目は、データそのものと、名称、ステータス、タイプ、単位、サンプリングピッチなどのデータの意味付けをするための付加情報をもつ。

なお、データ項目とは、入力、検索あるいは保守管理の最小単位である。データ項目を識別するための名前をデータ項目名と呼ぶ。データ項目名は、データ検索のキーとなる情報で、PIDである。PIDとは、つぎのような形式を持っている。

1	2	3 - 6	7	8	9 - 11 (Byte)
大 分 類	小 分 類	設 備 番 号	装 置 区 分	信 号 識 別	一 貫 番 号

3.2.1 実験データのモデルとファイル構造

実験データは、1つのショットに従属しており、そのショット中のデータの区別は計測点毎の名前付けが行なわれる。従って、基本的にショットとデータ項目が定めれば実験データが定まるというもので、第3.2.1図のような従属関係となる。

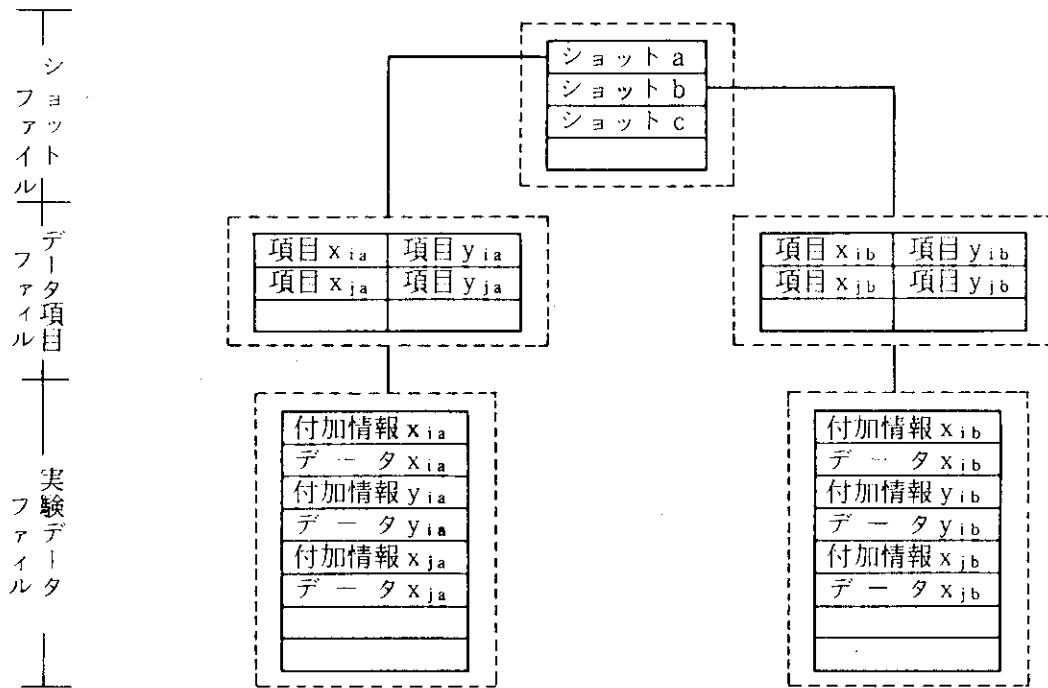


第3.2.1図 ショット従属の実験データの特徴

ここで、各レコードタイプを考えてみると、ショットは、放電方式、ショット番号を検索キーとする1つのレコードタイプとして扱うことができる。データ項目は、PIDなどのデータ項目を検索キーとする1つのレコードタイプとして扱うことができる。実験データは、データ項目に從属することからデータ項目のレコードタイプの1つの属性であると考えられるが、形式や長さが様々であるため、別のレコードタイプとして扱った方がよい。また、実験データは、付加情報部とデータ部に分けられるが、これらは、データ項目ごとに扱う情報が異なるため、それぞれがレコードタイプになると考えられる。

データモデルとしては、第3.2.1図からもわかるように、階層型データモデル^{*}が良いと思われる。というのは、ネットワーク型モデルにするほど上下関係が複雑ではないし、また、これらのレコードタイプの関係および実験データの性質から、実験データをテーブル化することにより、リレーショナル型データモデルを採用することも考えられるが、リレーショナル型データモデルは、データが表形式になる場合に適しており、時系列をはじめとする多次元で多種多様のデータを表形式に変換することは、非常に難しいと考えられるからである。この場合、たとえば、ある設備のデータということできくり出すことができる構成になっていないことに注意が必要で、設備単位というまとまりはあまり意味無く、PID番号という単位で十分と判断したからである。

この階層型モデルを採用した本DBのファイル構造は殆んど議論の余地なく第3.2.2図のとおりとなる。



第3.2.2図 ショット単位のファイル構造

* 付録1参照

** 付録2参照

この形態は、1ファイル=1ショットということであり、ファイルの構造を単純化するためには、望ましいものといえる。

この形態では、各レコードが構造化されているため、データ入力における重複データの検査のチェックが容易であるため、冗長性が防げる。また、データ検索も容易に行える。

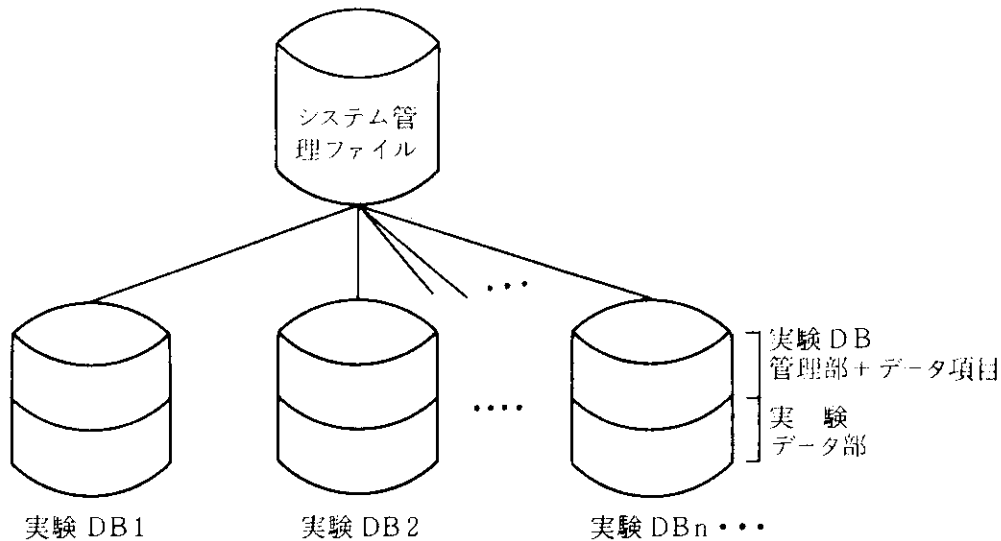
しかし、1ファイルに1ショット分の実験データが格納されるため、ファイルが大きくなり、ファイルの管理情報が破壊された場合、破壊による影響の範囲が大きいという欠点もある。

また、論理的なファイルの単位が設備などのデータ項目グループでないため、RACF^{*}などによるファイル単位の機密保持を行うときにデータ項目グループを単位とすることができない。

3.2.2 実験データの物理構造と検索方法

実験データベースを構成する論理的なファイルとして、ショット、データ項目、実験データを格納するファイル、および利用者登録ファイル、データ項目登録ファイルなどが考えられる。これらのファイルは物理的にデータセットとして分けると、検索を行うときにファイルのオープン/クローズの回数が増えるため、処理時間がかかると考えられる。そこで、ファイルのオープン/クローズの回数を減らすためにこれらのファイルをまとめることを考える。

上記のファイルで管理している情報の性格は、システムで共通に管理すべき情報と、個々のデータについて管理する情報とに分かれる。すなわち第3.2.3図のような構成になると考えられる。



第3.2.3図 ファイルの物理構成

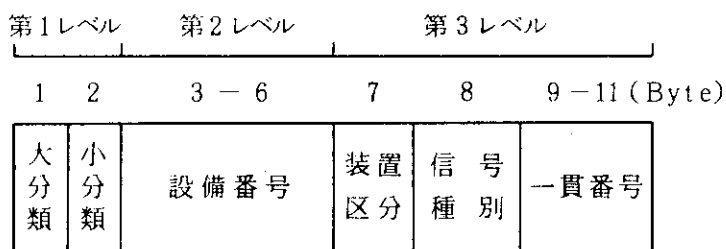
ショット単位で構築された上記実験データベースには、実験データおよびその管理情報としてデータ項目情報が格納される。

* RACF : Resource Access Control Facility ;大型汎用計算機がシステムとして持っているファイル単位に行う機密保護。

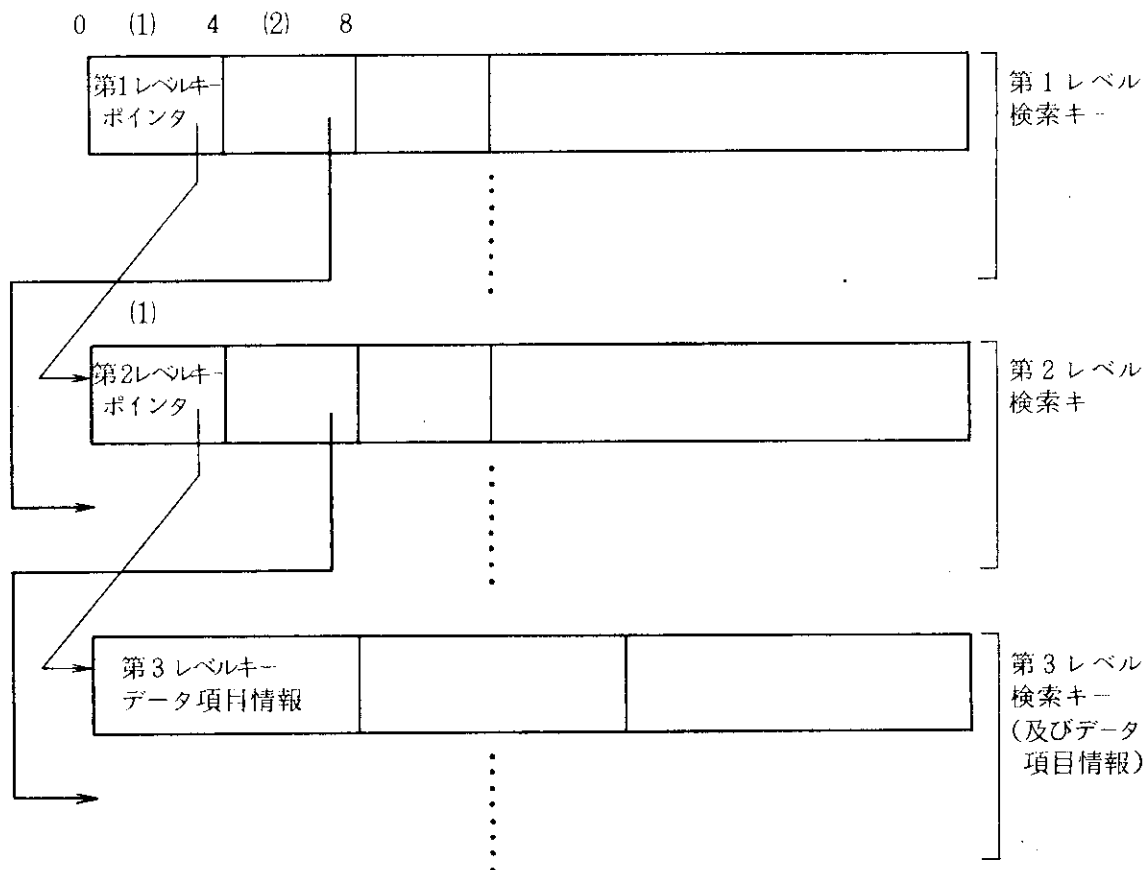
ショットが固定されれば、次は、どのようにしてデータを検索するのが、I/O 回数などの観点から必要かということについて調べていく。検索キーとしては、PID 番号を想定し話を進めるが、この想定は全く常識的なものである。ここでは、最も効率的と考えられる2つの方法について述べる。

(1) 検索を階層化して検索時間を平均化する方法

この方法は、検索キーを分割することにより、検索を階層化して検索時間を平均化する方法である。たとえば、PID 番号の長さは12バイトであるが、3つに分割して、第1レベル→第2レベル→第3レベルの順で検索を行う。PIDは、次のように分割される。

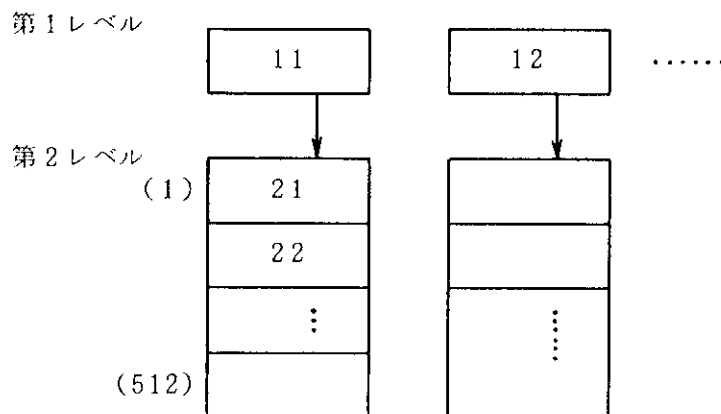


物理ブロックへは、次のようにレベルごとに分けて格納することになる。



第3.2.4図 レコード格納形式

1つのレベル検索キーとポインタとの長さを8バイトとすると、4096バイト(4KB)の物理ブロックに512個格納できる。これは、次のように第1レベル検索キー1つに対して、それに属する第2レベル検索キーが512個以下であれば1ブロックに格納できることを意味している。



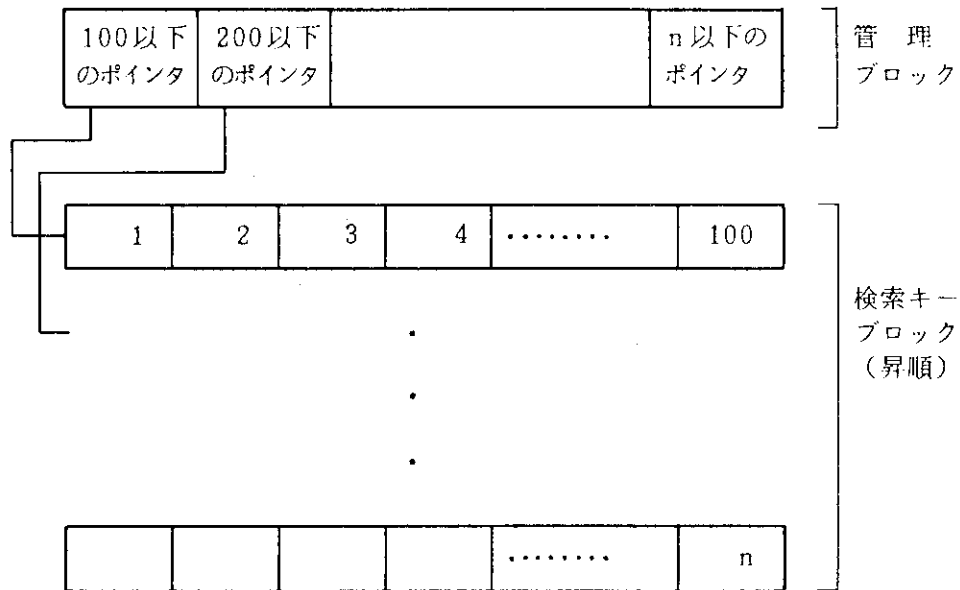
この方法では、各レベル検索キーが極端に偏っていなければ、充分迅速な処理が行えると考えられる。この場合、I/O回数が3回程度で平均化されると考えられる。

一方、レコードの追加を行う場合、検索手順と同様に、追加する検索キーを探して、既存でないことを確認した後、レコードを追加することになる。この場合、検索と同様に各レベル検索キーが極端に偏っていなければ、充分迅速であるといえる。なお、応用としてデータ項目情報を第3レベル検索キーとは別のブロックに格納する方法も考えられる。しかし、検索キーが偏っていると1つのレベルに対する検索キーが数ブロックに渡って格納されるため、I/O回数が増える。そこで、偏りを意識しなくてもよい方法として、次に述べる検索キーをソートして格納する方法が考えられる。

(2) 検索キーをソートして格納する方法

検索キーが多く、またレベルが付けられない場合の方法として、検索キーを昇順あるいは降順に並べて、順に格納する方法が考えられる。この場合、検索キーをソート(並べ替え)してから格納するため、登録するときに時間が掛かる。しかし、1度ソートすることによって、手を加えない限り検索キーの大小関係によりどこに格納されているかが分かるため、I/O回数は減る。

この方法では、次のようにどの物理ブロックにどのような検索キーが格納されているかを管理するブロックと、検索キーを格納するブロックに分かれる。また、検索キーとその属性情報とは1つのレコードとして格納することになる。



第3.2.5図 レコード格納形式

検索は、管理ブロックの情報により検索キーの格納されているブロックを見つけることになる。従って、I/O は2回必要となり、検索は迅速であるといえる。

一方、レコードの追加を行う場合、まず、同一の検索キーが既存でないことを確認した後、検索キーを昇順あるいは降順に並べ変えることになるが、この並べ替えには処理時間が掛かるため、予めレコード追加が予定されている場合は検索キーを登録しておき、データのみ格納するようにする方がよい。

この方法は、項目数が多いデータ項目登録ファイル、データ項目ファイルにおいて有効であると思われるが、検索キーとなるデータ項目名を入力時に登録する場合にはむいていない。

以上、ファイルへの格納方法を述べたが、特徴の比較から、我々は後者の方法を採用した。

これらのファイルは、すべてランダムに参照できる必要があるため、編成方法が順編成あるいは区分編成では処理できない。従って、直接編成(DAM)^{*1}あるいはVSAM^{*2}の相対レコードデータセット(RRDS)でなければならないと考えられる。

VSAMを使用すると、VSAMカタログによるデータセットの集中管理ができ、データ安全性もよく、退避、復元のユーティリティも整っている。しかし、VSAMカタログの作成などをDBMSの運用管理者が行なわなければならない、また計算センタの運用管理においてVSAM用のボリュームを用意する必要があり、他への影響が出てくる。無駄な空きスペースが出たり、I/O回数が増える点も問題と言える。

DAMは、データの安全性をデータベース管理システム(DBMS)で保証しなければならないが、

*1 DAM: Direct Access Method ;ユーザが物理空間を意識してデータ検索する方法。

*2 VSAM: Virtual Storage Access Method ;ユーザが物理空間を意識することなく各種キーを用いてデータ検索する方法で、このソフトウェアは汎用計算機のオプションとして供給されている。

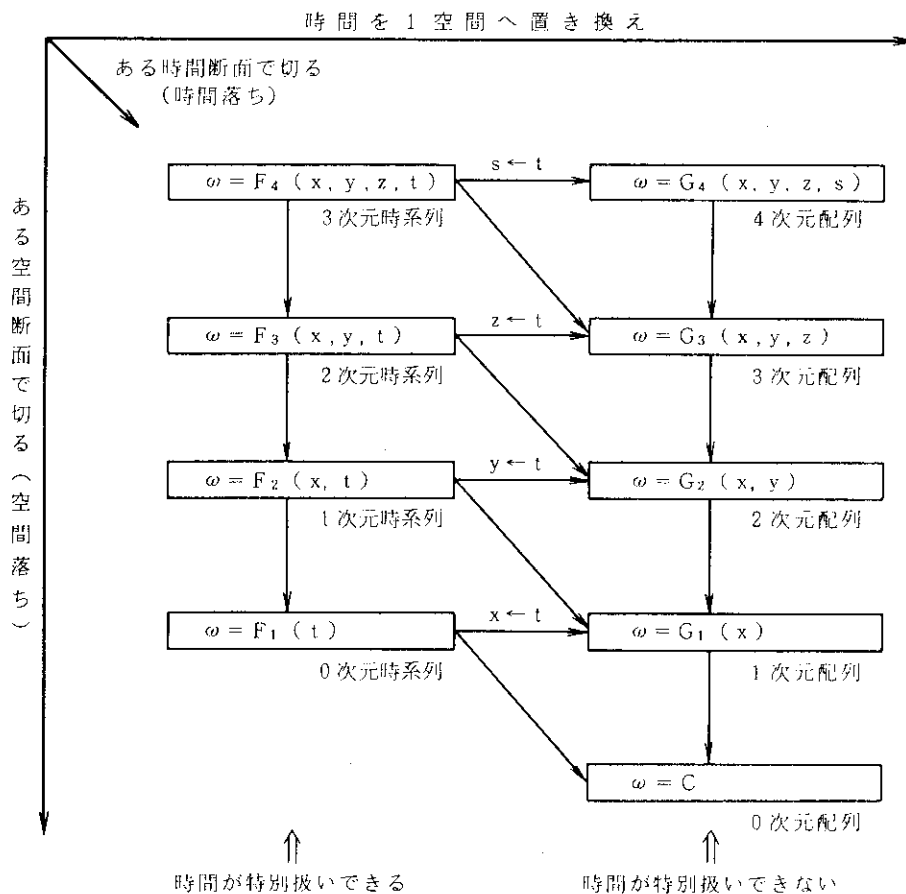
DAM 自体の処理は迅速である。

データの保全性の保証や各種ユーティリティを DBMS 固有に開発する場合は、DAM が適していると考えられる。

一方、実験データの利用にあわせて検索を迅速化するための論理を DBMS のソフトウェアに組み込むことも考えられる。たとえば、利用者が実験データの参照を行う場合、1つのショットについて複数のデータ項目を参照することが多いという特徴があるときは、検索されたショットに関する情報をつぎの検索依頼まで主記憶上に覚えておき、つぎの検索依頼が同一ショットの検索であればその情報を使用して、I/O を省略することが考えられる。また、利用者が複数のショットについて参照し交互に参照することが多いという特徴があるとき、複数のショットについて検索された情報を主記憶上に覚えておくことも考えられる。また、ファイルのオープン/クローズにも I/O を伴うため、オープンしたファイルはなるべくクローズしないように配慮する必要があると考えられる。

3.2.3 実験データの表現形式

通常実験データベースで扱うデータを分けると、時間の関数であるものと、そうでないものとなるが、時間のある1次元としてとらえれば、結局データの次元の違いという形でまとめることができる。これら次元の関係図を第3.2.6図に示す。時間を特別扱いとするかしないかは、データの容量という意味で重要である。特別扱いとは、時間情報はサンプル絶対時刻列 t_1, t_2, t_3, \dots のかわ



第 3.2.6 図 実験データの表現形式関係図

りに第1サンプル時刻 t_1 , サンプル周期 Δt , 全個数 n の3パラメータで表現することを意味している。逆に、サンプル周期が一定でない場合には、時間軸を1次元としてとる必要がある。

最もデータとして多いのは、0次元配列（1点データ）と0次元時系列（時系列データ）であるが、計測装置によっては1次元以上のデータも存在する。

実験データの表現形式は、実験データベースに格納されている形式を定義するもので、実験データを解析するためには“時間が $i \sim j$ のときの空間1のデータ”などの条件により実験データを抽出することになる。従って、実験データの抽出に先立って表現形式を定義する必要があるが、個々のデータ項目の形式が9種類のうちのいずれかに分類できるため、実験データの入力時、あるいはデータ項目の登録時に定義することができる。これにより、利用者は、“時間”や“空間”などの条件を指示することにより実験データの抽出ができる。

また、実験データの使用方法によっては、1つのデータ項目について別の表現形式で抽出する場合が考えられる。たとえば、“時間+空間”のデータについて、利用者が自分で利用しやすいように編集する場合、単なるデータ列として先頭から順に抽出することになる。このような場合、利用するときに表現形式を一時的に定義できるようになっている方が便利である。実験データのデータ部が9種類に分類できるように付加情報部も分類できると考えられるが、計測機器やデータ部の内容によって様々になり、将来的にも変更されることが予想される。従って、データ部を含め、分類された形式は、適宜登録できるようにする必要がある。

登録できる表現形式の種類として、まず、データ部が表現できるように0～5次元データが定義できる必要がある。付加情報部の表現形式は、様々なものが予想されるため、構造体をもつ一連の領域について個々の項目が定義できる必要がある。これらは、個々の次元や項目について、名称、長さ、タイプ（文字/整数/実数）などの属性の情報である。

実験データとして、コメントの類を入力可能にするため、数値データの他に文字データを考える必要もある。

3.2.4 実験データの変換、編集処理

各設備では計算機によって実験データの収集を行っているが、データの計算機上での表現は、各々の計算機が得意とする独自の形式を持っている。従って、実験データの解析を行うには、全系データの表現形式からFEPシステムの計算機が得意とする表現形式に変換する必要がある。また、実験データのディメンション（単位）は、計測器により定まるもので、解析に適するディメンションに変換する必要がある。

一方、全系の計算機の記憶容量や処理効率など収集の都合により実験データが分散したり束ねられたりしており、解析に適する形式では収集されていないと考えられる。従って、実験データの利用方法および管理方法に適する形式に編集する必要があることは既に述べた。

そこで、実験データの変換処理および編集処理をどの段階で行うか検討してみる。考えられるのは① 全系において処理、② データベース創成において処理、③ 利用者用インターフェイスにおいて処理、④ 利用者において処理 の4つである。なお、変換処理と編集処理は、同じ時点で行う必要はないが、各々について同じ特徴が多いのでまとめて行うことにする。

まず①は、全系の負荷や、実験シーケンスへの影響を考えると得策でない。③、④は、利用時の処理時間や、同一データを複数の人が見られることを考えると直ちに排除される。従って、②が最良という

ことになる。

3.2.5 開発言語について

実験データベース管理システムを開発するための言語として、FORTRAN 77, PL/I, COBOL およびアセンブラが考えられるが、これらの言語を混在させて使用すると、言語間のインタフェースを常に意識しなければならず、障害が発生しやすくなり、開発が容易ではない。そこで、このシステムの中核を成す言語を定める必要があると考えられる。

各データベースの利用として各種演算が組み込まれることが想定され、また、各種プラズマパラメータ計算を行う応用システムは、科学技術計算向けの言語で記述されると思われる。従って、多くの演算処理をサポートしている FORTRAN 77, PL/I が適していると考えられる。また、科学技術計算や表示処理には既存のパッケージが使用されることも考えられ、それらに対処しやすくなる必要もある。たとえば、科学技術計算用のパッケージである富士通の科学用サブルーチンライブラリ (SSL II) や、日本原子力研究所で開発された図形処理用のパッケージである「ARGUS」では、FORTRAN 77 インタフェースである。従って、これらに適している言語は FORTRAN 77 であるといえる。なお表示処理における画面の操作などで高級言語でサポートしていない部分はアセンブラになると考えられる。

実験データベースを構成するファイルの編成方法として、直接編成 (DAM) を用いる場合、COBOL はサポートされていないため適さない。VSAMで行う場合、どの言語でもサポートされているため、特に問題はないと思われる。しかし、高級言語ではファイルのアクセスには様々な制限があるため、アセンブラが最も適していると考えられる。

以上のことから、開発言語の中核を FORTRAN 77 とした。この補助として適宜 PL/I とアセンブラを使用することとした。

3.3 利用者の資格検査

本項目は、データの機密保護を目的に、データ毎に見れる利用者と、そうでない利用者が作れるようにする機能である。この機能をどのように実現するかについては次に示す方法が考えられる。

- (1) 一般のファイル保護に関する検討
- (2) ユーザ別、ショット別、PID 別のアクセス可/否テーブルによる資格検査
- (3) ユーザ別、(ショット + PID) 別アクセスレベル設定による資格検査
- (4) ユーザ別、PID アクセスレベル設定による資格検査

以下、それぞれについて検討する。

(1) 一般のファイル保護に関する検討

FEP (FACOM M-380) に標準的に設定されているデータファイルの機密保護について検討を行なう。

機密保護の設定方法としては、

1) RACF (Resource Access Control Facility) を使用した機密保護
TSS 端末における “PERMIT” 処理等^{*}。

2) パスワードを使用した機密保護

ファイル別に特定のパスワードと保護レベルをファイル創成時に設定する方法で、バッチジョブでは JCL (Job Control Language) の中の DD 文 (ファイル定義文) でこの 2 つのパラメータを指定することで可能となる。

の 2 種類があり、次の 4 つの設定が可能になっている。

- 1) NON : 一切の読み込み, 書き込み不可
- 2) READ : 読み込みのみ可能, 書き込み不可
- 3) UPDATE : 更新, 参照可能
- 4) ALTER : すべての操作可能

ところが、機密保護の単位が 1 データセット単位でしか行なわれず、1 PID 単位での機密保護を行なうことができない。

(2) ユーザ別、ショット別、PID 別のアクセス可/否テーブルによる資格検査

ユーザ別、ショット別、PID 別の機密保護を行なう方法として、資格検査用のテーブルを用意する必要があり、このテーブルは、次の 3 つのパラメータのマトリックスで表現されることになる。

- a) ユーザ ID (計算機登録番号)
- b) ショット番号
- c) PID 番号

例えば、次を仮定してデータ容量を計算してみる。

ユーザ総数 (一般職員, 外来研究員等) : 200 人
PID 総数 : 5000 PID

ここで、1 PID あたりの検索キーとして、大小分類、設備番号、装置区分、信号区分、一貫番号、及びアクセス識別用フラグで構成された 12 バイトの情報が必要であり、同様にユーザ登録番号に 4 バイトの情報が必要になるため、1 ショットあたり—

$$200 \text{ 人} \times 4 \text{ Byte} + 5000 \text{ PID} \times 12 \text{ Byte} = 60.8 \text{ kByte} / \text{ショット}$$

の容量が必要になる。

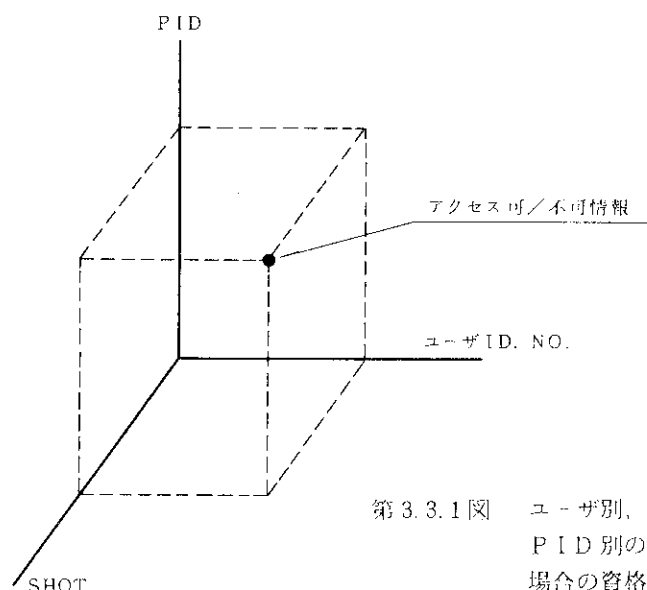
またこれらの資格検査が、過去のすべてのデータに対して適用されることになると、1 ショットあたりにショット番号識別のための情報として 7 バイト必要となる。

この為、1 日 25 ショット、1 月 18 日間の放電実験を実施すると仮定すると、

$$60.8 \text{ kByte} \times 25 \text{ ショット} / \text{日} \times 7 \text{ バイト} \times 18 \text{ 日} / \text{月} = 192 \text{ MByte} / \text{月}$$

の大容量の資格検査テーブルが必要となる (第 3.3.1 図参照)。しかも、このテーブル中をデータ検索行なうこととなり、計算機資源の制約 (I/O 回数, 実行リージョンサイズ等) から考えて現実的でないことが明らかである。

* PERMIT 処理: ファイルをどのユーザに対して参照可能にするかを PERMIT というコマンドで指定する方法。



第3.3.1図 ユーザ別、ショット別、PID別の検査を行う場合の資格検査テーブル概念図

(3) ユーザ別、(ショット+PID)別アクセスレベル設定による資格検査

(2)の議論で、全ての情報に対して資余検査用のフラグを有するのは、現実的でないことが明らかになったが、先の提案の条件をゆるめ、ショットが異なっても1ユーザがアクセス可能なデータは変わらないものとし、ユーザ単位の情報には、アクセス・レベルだけが設定されているものとして先のマトリックスから次元を1つ落としたもので考えてみる(第3.3.2図参照)。

この場合、資格検査テーブルには各ユーザに関する情報がないため、マトリックスの容量を先の条件で試算してみると、

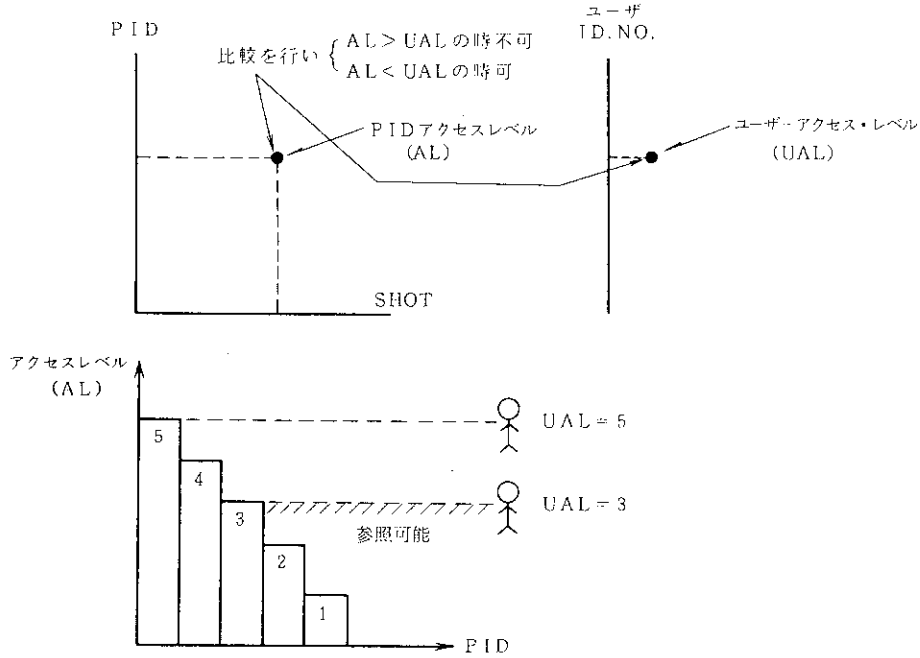
$$12 \text{ Byte} \times 5000 \text{ PID/ショット} = 60 \text{ kByte/ショット}$$

となり、1月あたりにしてみると、

$$60 \text{ kByte} \times 25 \text{ ショット/日} \times 18 \text{ 日/月} = 27 \text{ MByte/月}$$

となる。

たしかに、(2)に比べると容量としては1/7になるが、(2)と同様資格検査テーブルの容量が Mega 単位でありショットに伴って増えていくため実用的とは思えない。



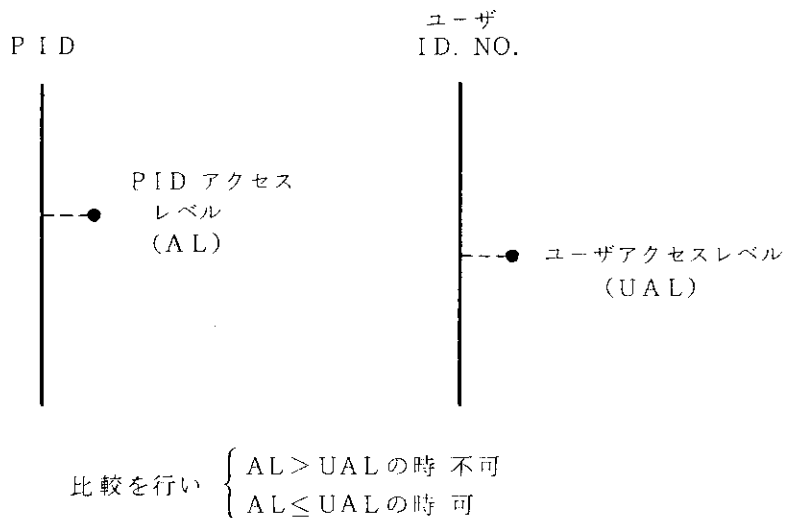
第 3.3.2 図 ユーザ別、(ショット+PID)別アクセスレベル設定による資格検査

(4) ユーザ別、PID別アクセスレベル設定による資格検査

(3)の方法でも、資格検査テーブルはショットに伴なって増えていくため、さらに、ショットの次元を落として考える(第 3.3.3 図参照)。この場合の容量を計算してみると、

$$4 \text{ Byte} \times 200 \text{ 人} + 12 \text{ Byte} \times 5000 \text{ PID} / \text{ショット} = 60.8 \text{ KByte}$$

この方法であると容量の点では問題が無くなったものの、ある時ある個人のアクセスレベルが上がった場合に、昔は見る事が許されなかったデータがある時点から見れてしまうという問題が生ずる。この点については締らめることが前提となる。



第 3.3.3 図 ユーザ別、PID別アクセスレベル設定による資格検査

以上の(4)つを表にまとめると、第3.3.1表のようになる。

第3.3.1表 資格検査方法の比較一覧表

保護方法 項目	RACF PASSWORD保護	USER. PID. SHOT 可/不可検査	USER(PID+SHOT) アクセスレベル検査	USER. PID アクセス レベル検査
テーブル容量	—	192 MByte/Month	27 MByte/Month	60.8 KByte
検索回数	—	(104+n)回	$\approx 1^{(13+n)}$ 回	≈ 1 13回
保護の確実性	○	○	○	○
項目の多様性	△	×	○	○
設定の容易性	○	×	△	○
メンテナンス性	△	×	△	○
ショット毎の独立性	○	○	○	×
総合評価	—	×	△	○

≈ 1 nの値はショット単位に増加し、 $n = \lceil \log_2(\text{ショット数}) \rceil$ の関係を有する。

これより、ユーザ別、PID別アクセスレベル設定による資格検査が最も良いと判断できる。

3.4 データ取得ツールと参照支援ツール

実験データベースの利用の目的は、様々であろうが、人間が認識するのだから最終的には、グラフや数値であろう。これらは、実験データをそのまま表示する場合や何らかの演算処理の後に表示することもあるであろう。後者の演算は、利用者によって色々なことを考えるわけであるから、そういう自由度を全く狭めることなくデータ取得することに加え、より進んだデータ加工も行なうサブルーチン群の必要性を意味し、一方、前者の表示についても簡単にグラフ出力可能なようなツールも必要となる。さらに様々なデータ解析を、プログラムを組むことなく可能になれば、かなりの労力の軽減となることは間違い無い。

頭に思い描いた処理（グラフ出力も含む）を実現するには、通常プログラミング、コーディング、デバッグといった作業プロセスを通ることになる。このプロセスは、時間的にも、労力的にもかなりの作業になり、必ずしも本質的とは思えない作業を利用者として強いられるのは不合理な話とも言える。この理由は、第1に計算機が“何でもできる”ようになっている一方、良く言われるように“そのままでは何もしない”ようになっているからである。第2に、第1の理由の裏返しになるが、利用者がソフトウェア全体に対し詳細な要求をするからである。以上の2点が大きな理由と考えられる。すると、作業を簡単にするためには、利用者がある範囲の要求しかしなければ良いわけである。言い換えれば利用範囲を限定する一方使い易さに注意したシステムを開発し、それでは満足できない利用者には、従来の手順を踏んでもらうということにすれば良いのである。勿論、利用範囲を限定するとは言っても普通に使うと思われることは全て取り込むことは言うまでも無い。

このような考え方に立ってツールの分類を行うと次の通りである。

① 汎用ハンドリングパッケージ

生データをデータベースから取得してくるサブルーチン群で、単純にデータを持って来るものや、簡単なデータ加工として、最大、最小、平均の各値を行うもの等が考えられる。

② 拡張ルーチン

簡単なデータ加工から一歩進んだデータ処理を行うサブルーチン群で、データ補間処理を行うもの、微分・差分・積分を行うもの、さらに FFT（高速フーリエ変換）を行うもの等が考えられる。

③ 図形処理支援ツール

図形表示を行う前に何らかの演算を行うのに、①、②を使ってプログラムを組むのも良いが、先の実験からもっと簡単にするために、会話形式で計算できるツールを製作することを考える。例えば、

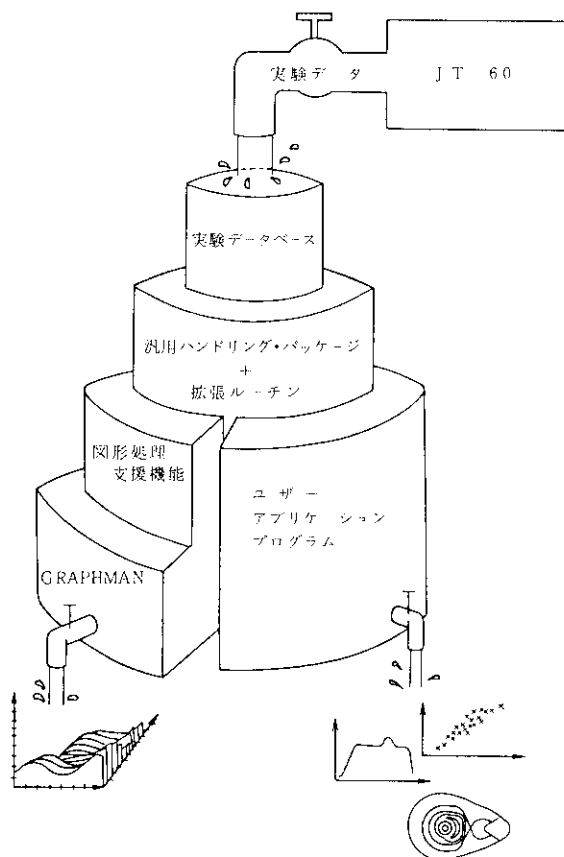
- (i) あるショットを指定し、いくつかの PID のデータを変数に割り当てる。
- (ii) それらの変数を用いてある演算を行い、演算結果を新しい変数に定義する。
- (iii) 演算結果をグラフ表示する。勿論、軸のパラメータ等を変更して適切なグラフが出せるようにする。

というような具合である。

当然の順序として、世の中に(i), (ii), (iii)の機能を持ったソフトウェアが無いか調べた結果、(iii)としては原研にて開発した ARGUS というツールが有り、さらにそれを包含した上、(ii)の機能を組み込んだ形のソフトウェアを富士通（株）が提供していることが分った。このソフトウェアは、GRAPHMAN というもので機能は、かなり充実しており、特に GRAPHMAN 内でコマンドプロシージャ（命令語の列）を組むことができる為、(ii)の機能をも含んでいると言える。

そこで、GRAPHMAN とインターフェイスをとるプログラム（これを図形処理支援機能と呼ぶ）を必要とするが、その中に(i)の機能を入れた形で開発することにすれば、データベースシステムとして必要な機能を一応完備することになる。

以上の利用に対するツール体系を模式図を、第 3.4.1 図に示す。



第 3.4.1 図 データ解析の模式図

3.5 運用管理支援ツール

データベースに限らずデータというものに求められる条件は、正確な値である。勿論、検出器系からの生データは誤差を含むものであり、全く正確などというのは不可能である。しかし、逆に誤差を正しく評価さえすれば、その誤差の範囲で正しいと言える訳であり、そのようなデータは、正常なデータと呼ぶことができる。一方、検出器系のトラブルなどからデータが異常になる場合がありその異常も何種類かに分類できる。このようにデータ毎に“質”が違うわけであり、データを使う人はその“質”でデータの信頼性を判断することが必要となる。この“質”をデータステータスと呼び、全てのデータに付加することを JT-60 では標準化している。さてこのデータステータスは、データが発生した時に付加されるのが最も一般的で、引き続いて作成されるデータベースにも、そのまま登録される。しかし、データ発生時では、判定のできないデータである場合や、判定はしたが間違っただけの場合などは、一度データベースが作成完了の後であっても変更される必要がある。さらに、演算結果データの場合などは、演算式が間違っていた場合には、再度演算して正しい演算結果を再登録しておかねばならないだろう。このように、データベース内のデータを常に最新で信頼性の高いデータに保つためには、一度作成したデータベースと言えども適宜、修正や変更が加えられるようにしておかなくてはならない。また、そうでないならデータベースという考えを採り入れ、冗長度を最小限にする一

方、維持性（保全性、信頼性）を高めた意味が無くなってしまふ訳である。

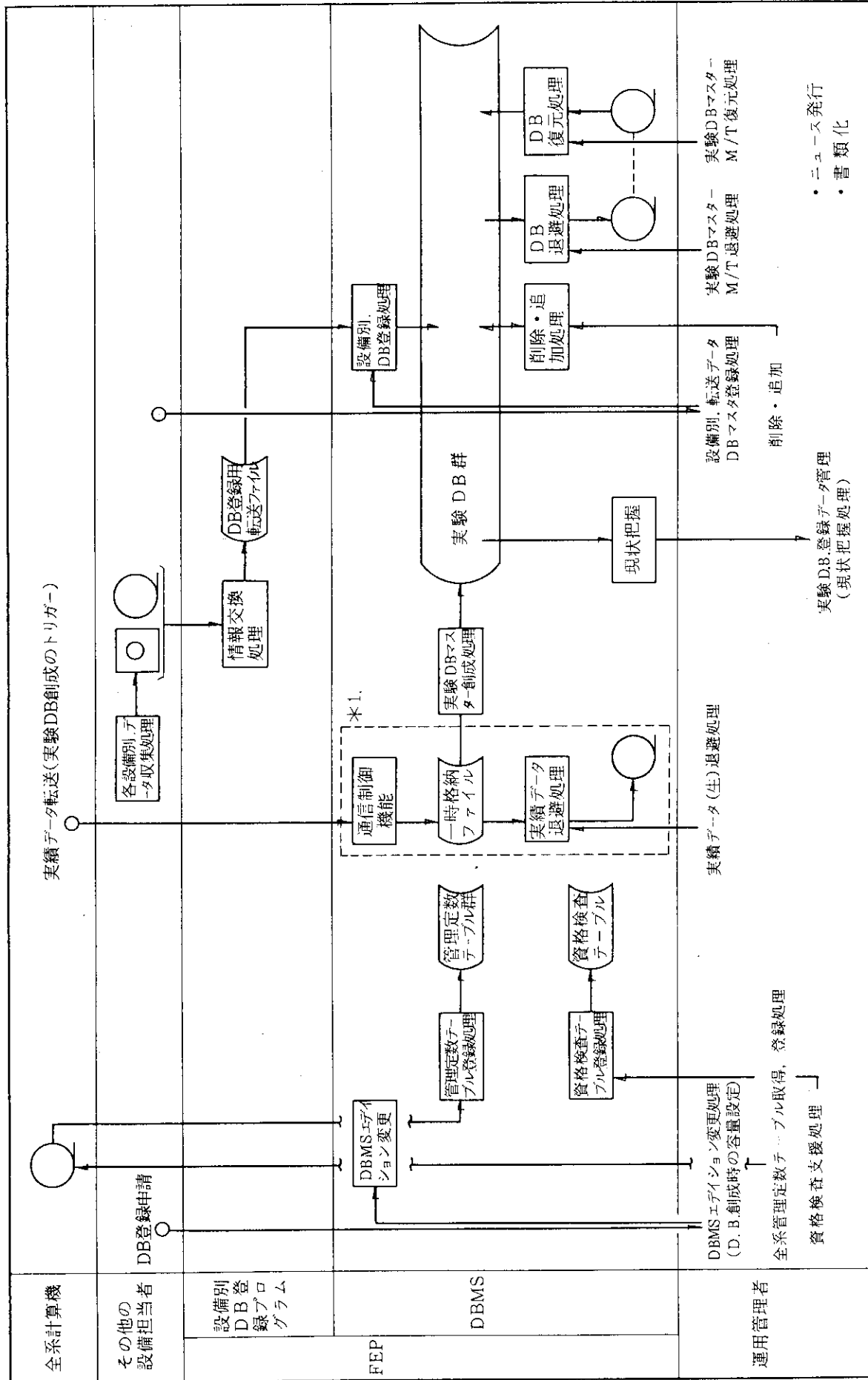
データベースとは、最良の状態に維持されるべきものであることは今述べたとおりであるが、そう言った作業を実行する人（即ち運用管理者）の仕事の具体内容は、少し考えただけでも多岐に渡っていることがわかる。先のデータ修正、変更は、データベースの内容の管理という意味で、本質的な作業の一つである。また、J T - 6 0 の改造に伴い追加、削除データをユーザへ知らせることも重要な事であるし、計算機の資源（例えば、記憶容量）の点での制約からくる各種作業（例えば、毎日磁気テープにデータを退避する）や、データ取得ツール、参照支援ツールの修正、利用者の変更・追加に伴う資格検査表の修正も必要な作業として必要となる。このように考え、運用管理者の作業内容を大別すると次のようになる。

- ① データ内容を最良の状態に維持するために必要な作業。
- ② 計算機の記憶容量の制約又は安全の観点からくる磁気記憶媒体への退避、及び復元に係わる作業。
- ③ 各種ツールの修正作業。
- ④ ①～③の作業に伴う各種書類の作成とユーザへの周知徹底作業。

一日当たり、25ショット放電すると仮定すると芒大なデータ（5MB×25）が発生する一方で大勢のユーザがデータ解析をするという、言わば生きたシステムを運用管理していくことが必要となるのだから、作業の種類、量ともに多くなるのは当然と言えよう。多くなれば間違いを犯す確率が増えることになり、これを回避し、より効率的に運用管理作業を行なうための各種支援ツールの必要性が自然に求められるという訳である。

運用管理支援ツールの具体的な処理内容は、運用の手順を考えることで、把握することができる。第3.5.1図に運用フローの概略を示す。図に示された処理が支援ツールとして必要な処理ということができる。以下に箇条書きすると、

- (i) データベース退避・復元処理
- (ii) ユーザ資格検査支援処理
- (iii) データベース削除・追加処理
- (iv) データベース現状把握処理



*1. 全系-FEP間結合に伴う通信制御機能に含まれる。

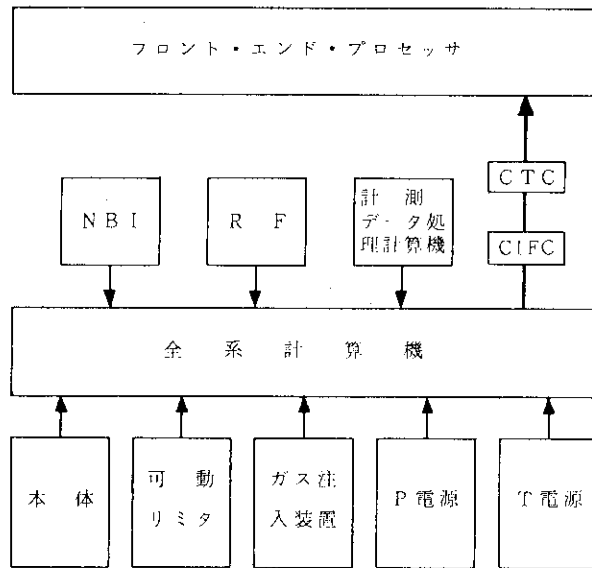
第 3.5.1 図 実験データベース運用フロー

4. システム設計

本章では、FEP内における全系実験データベース及び全系・FEP間の通信の部分のシステム設計について述べることにする。システム設計とは具体的に作る一步手前の設計であり、実際の完成状態をイメージとして思い描きながら外部仕様を決定していくものである。

4.1 全系・FEP通信とその周辺

全系は、JT-60の総括制御部であり、実験のシーケンスに従って装置全体を制御している。1回の放電（ショット）が終了するとその放電に係わる各種データ（放電条件、放電前後取得データ、放電結果データ、など）を全系は一括して収集する（第4.1.1図参照）。全系計算機内部では、4ショット分のデータバッファしか保有しておらず、磁気テープにデータ退避しない限り、これを越える放電を行うことができないという制約がある。



第4.1.1図 計算機ネットワーク概略図

FEP と接続することにより、ショット直後に全系から FEP へデータ転送を開始する。ここで簡単に計算機間通信について触れておく。一般にデータ転送を行う場合ハードウェアの接続が保証されれば、メッセージをやりとりするための約束（この約束を「通信プロトコル」と呼ぶ）に従ってデータを送受する。

ハードウェアの接続は、計算機の機能に依存するが、汎用大型計算機の計算機間接続としては IBM が規格化している“IBM チャンネルインターフェイス”が一般的である。従って市販のミニコンピュータ、大型コンピュータは、それぞれの内部バスインターフェイスをこの IBM チャンネルインターフェイスに変換するアダプタを大抵は持っている。全系の核となる（株）日立製作所製ミニコンピュータ HIDIC-80E も、自らの直接制御（DMA）インターフェイスを IBM チャンネルインターフェイスに変換する装置（CIFC）を提供している。一方、（株）富士通製 FACOM-M380 も自らの I/O チャンネルと IBM チャンネルインターフェイスとの変換装置（CTC）を提供しており、第 4.1.1 図に示すような接続が実現する。

ハードウェア的に接続されれば、そのやりとりの約束である通信プロトコルを決め、そのプロトコルを実行するソフトウェアを製作しておかなくてはならない。通信プロトコルは、全系とデータ処理計算機（FACOM-M360）との結合の際に用いたものをそのまま流用することにした。ただし流用と一口に言っても、FEP 側は、機種、OS（オペレーティング・システム）共に異なるものへの移行であるので若干の改造は必要となる。

さて JT-60 の実験シーケンスに従って、又はオペレータの命令により伝送メッセージやデータを通信プロトコルに受渡すことにより、通信が成立する訳であるが、その部分の処理（送/受信処理やその起動処理）は、それぞれの計算機の運用と密接に関係しており使い易いか否かは、この部分の良し悪しで決まると言っても良い。そこで以下に両計算機のそれぞれについて説明をする。

4.1.1 全系側の送信・受信処理とその周辺

データ転送の方向は、概ね、全系から FEP へ 1 文字分の空白というものであるが、転送の信頼性を高めるために通常双方向でメッセージのやりとりを行う。放電実績データの転送については、データ量が～3 MB にもなるので、汎用側の準備処理のために転送予告信号を放電実績データの転送に先だっで行うこととした。第 4.1.2 図に転送メッセージとそのフローを示すが、1 ショット分のデータ転送はこのフローに従って行なう。

次に、全系のデータ転送に係わる制約条件からくる転送時の約束について、特に異常時の処理について明確にしておこう。全系計算機は、大きく3つに分類することができ、運転制御系、放電制御系、実時間制御系がそれらである。このうち、放電実績データをシーケンスに従って収集するのは、放電制御系であるから、放電制御計算機が最も適しているように見える。しかし、この計算機は、プログラムの負荷が異常に高く、又プログラムのローディングエリアも殆んど無いという状態で FEP 転送のソフトウェアを組み込むことは、難しい。また、連続シーケンス（複数ショット連続）を流している間に転送を行うとすれば、シーケンス監視機能と同時に転送プログラムが流れるという状態が起こり JT-60 のシーケンスに影響を及ぼす可能性が高くなって来る。そこで、放電制御計算機を諦め、実時間制御計算機に行なわせることを考える。この計算機は、放電制御計算機とディスクを共有しており、結果データを取り扱うことは可能であると共に、ショットとショットとの間は、全く動かない計算機であることが負荷の点からも望ましいと言える。一方、ショットを含めたある時間（放電1分前～結果データ編集）は、実時間プラズマ制御以外のタスクは動かさないという点が問題として挙げられる。これは、FEP へのデータ転送が中断される可能性を意味し、FEP 側の対応とも関係してくる訳である。

この問題点の解決策として次の各項が考えられる。

- ① 中断された場合は、全系は、メッセージ送信を中止し、放電終了後は、先の放電実績データを、転送予告メッセージから再度送信する。一方、FEP は、メッセージタイムアウト検出をすること、及び転送予告メッセージの受信時には、受信途中のデータセットを消去し、新たなデータセットとして認識する。
 - ② 中断された後の放電終了後、全系は、中断されたところからデータ転送を再開する。一方、FEP は、メッセージタイムアウト検出をしないようにする。
- ①は、折角、途中まで送ったデータが無駄になってしまい、時間的にも効率的でない。一方、②も FEP 側のタイムアウト検出機能を無くしてしまうことになり、通信上の異常時にも自動修復を期待できなくなる。ただ、回線を全系から切ることにより、メッセージの終結を行うようにすること、又は、転送予告メッセージを FEP が受信した時には、如何なる状態でも、新たなデータ転送として受信することのどちらかが施されれば運用で対処可能と言える。

上記の判断のうち、②の方法を通信方法として採用することとした。タイムアウトによる自動修復機能を断念する一方、回線の切/入の操作による再転送を可能にすることにした。

転送開始（又は再開）タイミングと中断タイミングについては、放電から約3分前（放電条件を JT-60 各装置にプリセットする時）に中断をすることにし、開始又は再開を放電終了後2分（放電後、全装置の高エネルギー状態が解除された時）、及び、次のシーケンス開始待ち状態の時、の2時期に行うことにした。

また、シーケンスとのインターロックは、必要最小限の条件、即ち、全系内の4ショット分のバッファ全てのデータが FEP 転送未完又は MT 退避未完の状態であった場合のみ次のショットのシーケンスを起動することができないものとした。これに伴ない、先のデータ転送開始（又は再開）タイミング及び1ショットのデータ転送終了時において FEP 転送未完のバッファを探し、もし有れば転送プログラムを起動するようにして、全系バッファ内に FEP 転送未完状態のデータをできるだけ残さないように考慮した。

最後に、試験や、何らかのトラブル時を想定してオペレータのマニュアル起動でデータ転送を開始

する機能をつけ加えた。中断／再開のシーケンスには拘束されるが FEP 転送完のショットであっても再度転送可能とした。

4.1.2 FEP 側の送信・受信処理とその周辺

前節で述べたように FEP は、回線切り離し時のみ処理を初期化するが、回線接続中は、全系からの指令に忠実に従えば良いという単純な構造にした。

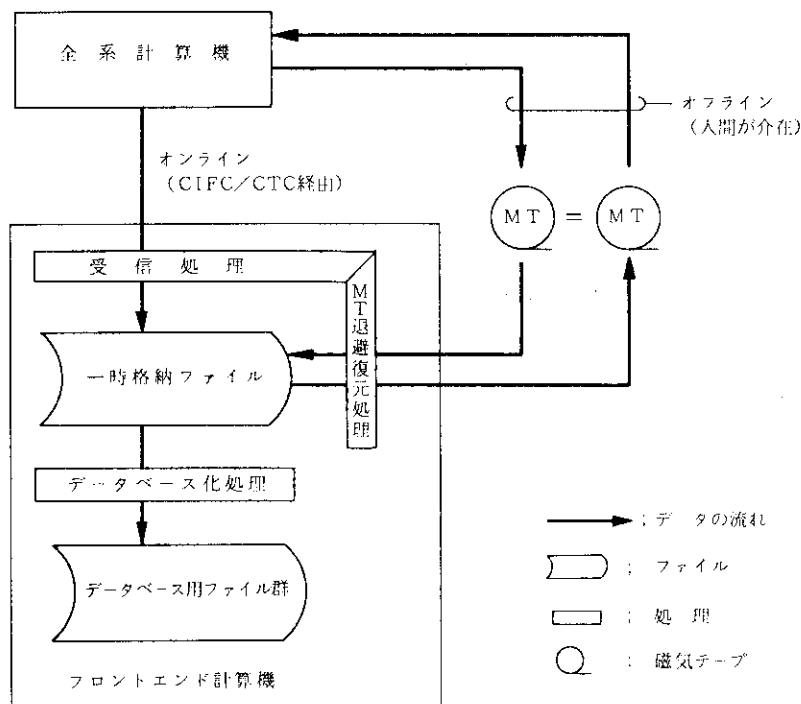
一方、受信したデータを最終的にはデータベース化（DB化）する訳であり、そこへの過程のどこまでを FEP 側受信処理で行うかということ、言わば、受信処理と DB 化処理との取合いを明確にしなければならない。例えば計算機によって異なる数値の表現形式を変換する処理などをどの段階やるかといった事該を明確にしなければならないということである。

これを決めるために、試験や運用の単純化を念頭に次のような条件を置いた。試験の単純化はシステムの信頼性を高めるという認識は重要である。

- ① FEP が全系から受信したデータは、そのまま何の加工もしない状態で一旦格納する。そのデータは FEP 側退避 MT と全系側退避 MT とが同一の形式になるようにする。
- ② DB 化との取り合いも、全系の加工しない状態のデータとする。

これらの条件によって、どのような運用が可能になるかと言うと、第 4.1.3 図に示すように、CIFC / CTC を介して転送されるルートとは独立にオフライン的に MT を介したルートが登場する訳で具体的には、次のような利点が生ずる。

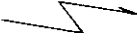















- ④ 全系計算機で退避した MT を直接 FEP に格納しデータベース化することができる。
- ⑤ オンラインで FEP に転送した放電実績データを MT に退避しておけば、その MT を直接全系計算機に格納することができる。これにより、過去の放電条件を全系計算機内部に再現することが可能となる。



第 4.1.3 図 FEP - 全系間データの流れ

- ◎ 一時格納ファイルを設定したことにより、受信処理とデータベース化処理との分離が可能となり、全系-FEP間の転送の渋滞を防止でき、延いては、JT-60のシーケンスへの影響を最小限に留めることが可能となった。

このような利点を確かなものとするためには、ソフトウェア的な考慮が必要となるが、特に、データ保全のための一時格納ファイルのインターロックは重要である。それには、1放電毎にフラグを持ち、MTに退避を行なわない限りはデータを消さないようにする、データベース化されないうちも同様である、受信途中でトラブルが起った場合には、ファイル未完として上書き可能とする、などの制御を行なわなくてはならない。このようなことを検討する場合のポイントは、一時格納ファイルが保証すべき事該、即ちデータの保全（MT退避・データベース化の完璧性、一時格納ファイルの有効利用）を明確にし、これを前提として入出力（受信処理、MT退避復元処理、データベース化処理）の競合を防ぎ運用面での便を計ることである。このようにして決めたフラグとその遷移は、第4.1.4図のとおりである。フラグは、ファイル情報管理テーブルの中に2種作った。（STSとDBF）

データ発生パターン	タイミング	データの動き	左の動作完了時のフラグ	
			S T S	D B F
全系からの オンライン 受信	全系通信	 	80 ↓ 40	8000 ↓ 0000
	データベース 創 成	 → 	↓ 40	↓ 8000
	一時格納 ファイル退避	 → 	↓ 80	↓ 8000
全系MTか らの復元	全系MTから 復 元	 → 	80 ↓ 40	8000 ↓ 0000
	データベース 創 成	 → 	↓ 40	↓ 8000
	一時格納 ファイル退避	 → 	↓ 80	↓ 8000
FEP MT からの復元	FEP MT から復元	 → 	80 ↓ 80	8000 ↓ 0000
	データベース 創 成	 → 	↓ 80	↓ 8000

ファイル情報管理テーブル

S T S	D B F
00 (Hex) : 格納未定	0000 (Hex) : DB未定
20 : 不 定	8000 : DB創成完
40 : 格納完	
80 : 退避完	

一時格納ファイルの書き換えの条件は (STS = 00又は20又は80) かつDBF = 8000とする。

第 4.1.4 図 データの動きとフラグの遷移

4.2 データベース創成

一時格納ファイルに放電実績データが格納された後、データベース化の処理を起動することによりデータを最終的な形に変換する訳である。どう創成するかを考えるには、むしろ、どう使うかを検討してその使い易い形に創成するというのが当然であろうが、より重要なこととして、創成ソフトウェアをどういう構造にしておくかを検討することが挙げられる。この構造を散えて議論するのは、このデータベースにある程度の柔軟性を持たせ、将来に渡って想定される改造、追加作業に対応するためである。

第3章において、データベースの構造について基本的な検討を行ったが、ここではそれを踏まえた上に上記検討や、実際の境界条件を加味して具体的にどう作るかを明確にする。

4.2.1 創成処理の起動

前節で述べた一時格納ファイルに全系放電実績データが格納されると、引き続きデータベース化処理が速やかに起動されなくてはならない。一方、ジョブの起動というのは、バッチジョブを主体とする大型汎用計算機にとって普通の使い方ではない。あるジョブから別のジョブを起動する方法として FACOM M-380 で考えられるのは、次の2つである。

① AOF (Advanced Operation Facility) を用いる方法

計算機のオペレーションをするためのコンソール CRT にメッセージを出力すると、そのメッセージを自動的に計算機が読んで、もし予め登録しているメッセージであれば、予め登録しているジョブを起動する機能 (AOF) を利用し、一時格納ファイルへの登録プログラムの中でコンソール CRT にメッセージを与え、それを介してデータベース化ソフトウェアを起動する。

② インターナルリーダを用いる方法

これは、親のジョブ制御文の中に、子のジョブを実行するための SUBMIT 文を挿入するものである。SUBMIT されるのは、子のジョブ制御文が格納されたファイルである。この方法は、一時格納ファイルへの登録を実行後、データベース化ソフトウェアのジョブ制御文を起動して、自らのジョブを終結されるもので、仮にデータベース化ソフトウェアがトラブルを起しても、何ら親のジョブに影響を及ぼさない。

これら2つのうち AOF を用いる方は、計算機オペレータや計算機運用への影響が大きいため排除した。従って、受信処理と創成処理との取合はインターナルリーダが実行するジョブ制御文が格納されているファイルを決定すれば済むという単純なものとなる。

全系 MT から復元する場合には、データベース化処理をする場合もしない場合もあり、従ってこの起動は端末から行えるようにすれば良いのである。

4.2.2 データ変換・編集処理

一時格納ファイルに入れられた放電実績データは、全系計算機に復元可能な形で格納されていることは、既に述べたが、これは、全系計算機にとって都合の良い形のフォーマットになっていることを意味しているにすぎない。FEP で利用することを考えれば当然それ相応のフォーマットに変換する必要がある。

ここでまず、全系フォーマットの特異性を理由と合せて以下に示す。理由を付記するのは、同様の

境遇のシステムへの示唆を含めるためである。

① 結果データエリアの有効利用から、各データの名称などのデータの属性を共通に持っている^{*1}。従って、オンライン/オフライン共に放電結果データ中には、生データが詰っているだけである。

② PID番号（データ識別番号）と生データとの対応も、PID番号を分解して検索テーブル^{*2}を探していくことで初めてとられるという状態である。特に、この検索テーブルは、メモリの制約から、極めて複雑な構造をとっている。

③ 計算機の違いからくる特殊性として、全系計算機は16ビットが1ワードである。また、浮動小数点データの2進数形式が特殊である。

さて、データベースへの移行時には、特殊性をそのままにして変換することも可能であるが、3.2節での検討から、データベース創成時に行うことを念頭に置くことにする。さらに、

④ FEPには極端な容量上の制約は無い。また、データ属性が複数のショットで共通であっても、その単位で属性を格納したファイルとの紐付けを記憶しておかねばならず、構造上簡単で無く、最適と言えない。ショット毎に独立にすることが必要である。

⑤ PIDの番号と生データ格納エリアとの対応は、実験の進行と伴に変っている。の2つのことから、全系の特殊性の①、②を解消するには、「PIDファイル管理定数テーブル」の内容を全てデータベース化の際に含めて変換することが必要となる。即ち、1ショット1ショットが互いに情動的に独立するように作らねばならないという訳である。

今まで述べた全系の特殊性を創成時に解消すること以外のデータベース創成に対する要求を以下に挙げ、変換・編集処理を明確にすることとする。

④ PID番号毎に、3.3節で述べた資格検査用のアクセスレベルの値を入れなくてはならない。

⑤ ショット終了後、放電実績データを用いて演算したデータを再度データベースに登録することが考えられるので、その分のエリアを創成時に確保しなければならない。これを放電後演算処理（ショットサマリ創成処理）と言う。

⑥ 1次元データ、2次元データというようにデータ構造という意味では、数種のデータが存在し、(第4.2.1表参照)、それらのデータを読むプログラムも、データ構造の種類の方は最低でも用意する方が、効率良いデータ取得には必要となろう。従ってPID番号毎にデータ構造（即ち検索方法）を表す情報をデータベース上に持たなくてはならない。

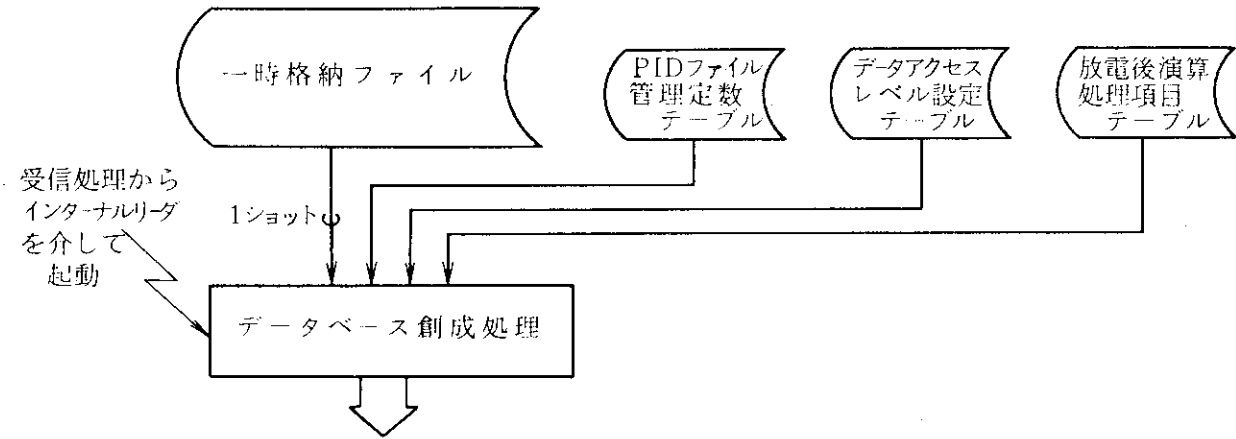
⑦ データ検索を迅速に行うために、PID番号とデータ格納エリアとを対応づける表が必要となる。

さて以上を考慮したデータベース創成処理とデータベース概略構造を第4.2.1図に示す。

*1, *2は「PIDファイル管理定数テーブル」に格納されている。

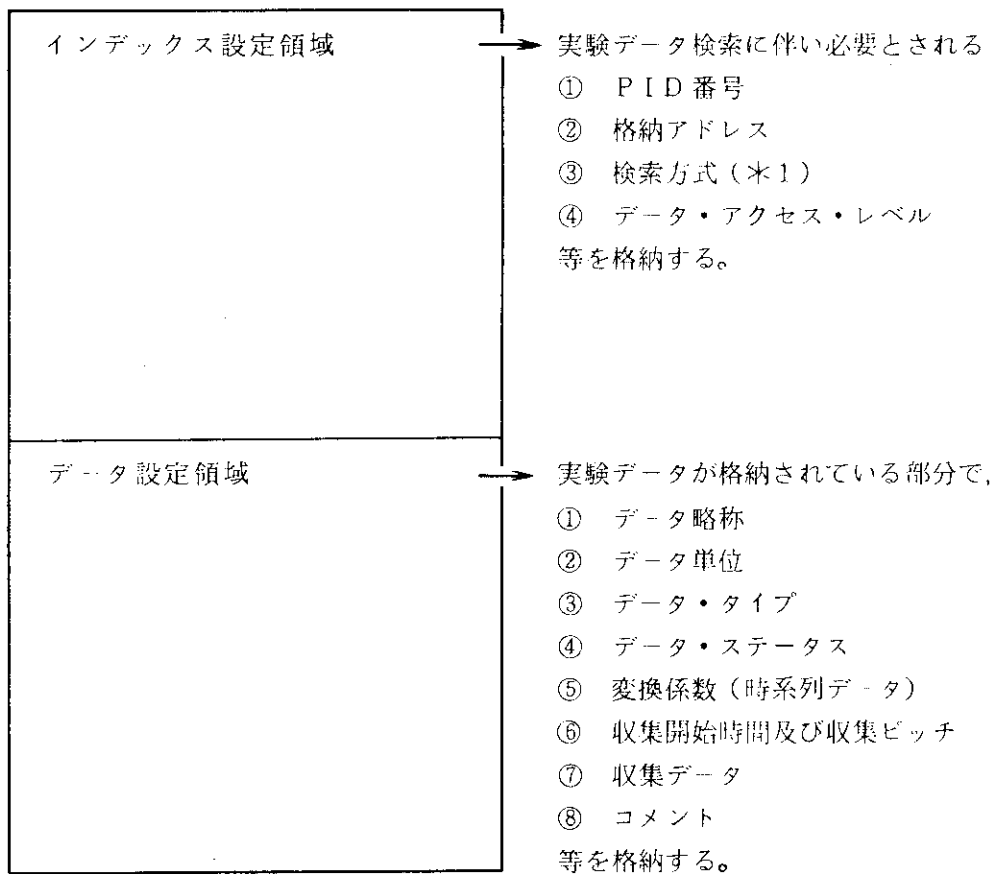
第4.2.1表 データの種類

No.	データの種類	データ構造の特徴
1	1点データ	1 PID に1つの数値が対応しているデータ (例: フィードバック開始時期)
2	時系列データ	1 PID に対してサンプル周期が一定の数値列が格納される。
3	放電条件データ (プレプログラム以外)	1 PID に1つの数値(設定値)が対応しているデータ。但し, 1点データと異なってデータステータスを持たない。(例: トロイダルコイル電流設定値)
4	プレプログラムデータ	波形で与える放電条件で, 最大19本の折線等で設定する条件。従って, データとしては(x, y)の20組から成るデータである。 (例: プラズマ電流目標値)
5	マスフィルタデータ	時系列データが横軸時間であるのに対し, マスフィルタデータは, 質量数を横軸にとったスペクトルである点に特徴がある。
6	コメントデータ	1 PID に対しキャラクタで登録されたコメントのデータ。(例: ショットのコメント)
7	放電前後取得データ	放電そのものの有無にかかわらず, 存在する1点データ。(例: 放電前後の真空容器温度データ)



ショット番号 = E000123

1ショット = 1 DAM



*1 種々のデータ構造の異なるデータが混在しているため、それらを識別するための情報である。

第 4.2.1 図 データベース創成処理とデータベース概略構造

4.3 汎用データハンドリングパッケージと資格検査

一般ユーザがデータベースのデータを読み出したり、また、特定の管理者がデータベースへデータの書き込みを行ったりするときのインタフェースをどのように提供するかは、データベースシステムと利用者との取合い、延いては利用者の自由度を決めるという意味で重要である。本データベースでは、データそのものを取り出すサブルーチン群を整備して公開し、さらにそれを用いて簡便に結果を見ることができるツール（第4.4節で述べる「図形処理支援機能」）を準備することとした。細かなデータの取扱いをするユーザに対しては、処理内容に応じたサブルーチンを使うことを想定し、ディスク入出力回数が少なく、CPU時間を短くするなど、使い勝手の良いものを製作することとした。手軽にデータを見たり加工したりするユーザに対しては、図形処理支援機能により端末との会話形式でほとんどのことが可能になることを目指した。（第4.4節参照）

さて、このような考え方で、作成する汎用ハンドリングパッケージを大きく次の4つに分類した。

① 基本アクセスルーチン群

実験データを一部又は全部取得する際に用いるサブルーチン群である。

② 拡張アクセスルーチン群

基本アクセスルーチンにて取得した各種実験データを統計処理、微積分処理、等をするサブルーチン群である。

③ データ登録アクセスルーチン

データベースにデータを登録、書き込むサブルーチンである。但し、データベースのインデックス部には、あらかじめ登録され、データエリアも確保されていることを前提とする。一方、第4.5節で述べる追加機能は、インデックスもエリアも確保されていないところにデータの追加を行うものである。

④ サービスルーチン

一般データ処理用に提供するサブルーチン群で、科学計算処理用ライブラリ（SSL II）の拡張機能とすることができる。

以上のサブルーチンの一覧表を第4.3.1表に示す。

これらのサブルーチンによりデータを取得する場合、データとそのデータを取得しようとしているユーザとの対応検査を行い、合格した場合にのみデータ取得可能とする。これにより未成熟データの使用やデータの不正利用を未然に防ぐことができるようになる。

ユーザの使い勝手の良いことが前提となるが、CPU時間、I/O回数などの技術的問題の最適化は勿論のこと、利用マニュアルの整備が極めて重要な仕事となる。

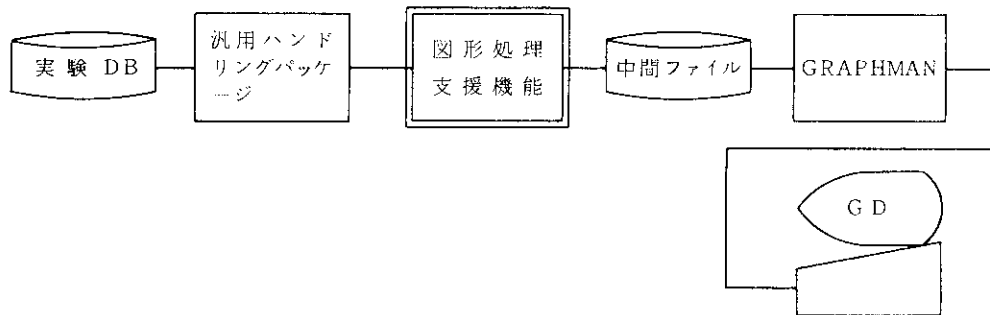
第 4.3.1 表 サブルーチン・処理内容一覧

	データ項目	処 理 内 容
基本 ア ク セ ス ル ー チ ン	[共通ルーチン]	① 指定ショットの実験DB有無確認 ② 指定PIDのインデックス部取得 ③ 指定PID生データ取得(キャラクターイメージにて取得)
	[一点データ]	① 指定PID(一点データ)のデータ取得
	[時系列データ]	① 指定時間帯(Ts~Te)でのデータ取得 ② " 最大値とその時刻 ③ " 最小値とその時刻 ④ " 平均値
	[プレプロデータ]	① プレプロデータの展開結果取得 ② プレプロ時間軸パラメータ取得 ③ 展開結果の最大値とその時間 ④ " 最小値とその時間
	[ショットサマリ]	① キャラクタ・データ(運転員名, 目的等)取得 ② 数値データ(Case 1~512)取得
	[マスフィルター]	① 区間(m/e)s~(m/e)eでのデータ取得 ② " 最大値とその(m/e) ③ " 最小値とその(m/e) ④ " 平均値
	[放電条件]	① 放電条件略称取得 ② 放電条件データ取得
	[その他]	① ショット毎でのコメント取得 ② 指定PIDのコメント取得 ③ ユーザーID資格検査
拡張 ア ク セ ス ル ー チ ン	[時系列データ, プレプロデータ]	① 指定時間帯Ts~Teでの積分 ② " 差分 ③ " 微分 ④ " フーリエ変換(FFT)→パワースペクトル ⑤ " 一次遅れフィルター ⑥ " 直線補間(内挿) ⑦ " スプライン補間(内挿)
*	[全データ形式]	① 指定されたデータを該当領域に格納
サー ビ ス		① 複数時系列データ間の相関計算 ② 時系列データの回帰計算

* データ登録アクセスルーチン

4.4 図形処理支援機能

J T - 6 0 のデータをどのように使うかは、ユーザによって千差万別であろうが、色々な演算を行った後、最終的には人間が目で見えて評価する訳である。そこで、簡便に演算できること、簡便にグラフ表示できること、という2つの条件を満たすツールを製作することを考えたことは既に3.4節で述べた通りである。その際に“GRAPHMAN”という富士通製のコードが強力なツールになることを述べたが、そのGRAPHMANで取扱えるデータ形式に、データベース内のデータを変換する必要がある。このデータ変換の部分に、演算機能を組込んで、より簡便で能率良く演算・表示を行うものとして位置付けた。(第4.4.1図参照)



第4.4.1図 図形処理支援機能の位置付け

このような位置付けの本機能の具備すべき仕様として、以下の事項が満足されていることとした。

- ① 実験 DB から指定 PID データの取得が行なえること。
- ② 取得された実験データに対する積分、微分、フィルタリング等の汎用ハンドリング・パッケージに含まれている各種演算処理に対応できること。
- ③ 四則演算処理が行なえるようにすること。
- ④ ②及び③に示した各種の演算処理は会話型のコマンド入力を行なうことで実行できるような構造とする。
- ⑤ 図形処理支援機能の出力としては、GRAPHMAN で取扱えるデータファイルを作成することとする。
- ⑥ 本処理で使用可能な変数名としては、英文字 A 0 ~ Z 9 の 2 6 0 文字を対象とし、名変数の属性は“FORTRAN”の暗黙の型宣言に従ったものとする。
- ⑦ 一点データ、時系列データの区分は指定されたデータの属性により自動的に設定されるものとする。
- ⑧ 変換結果データ(中間ファイル)をユーザーファイルとして登録する機能を設ける。
- ⑨ データの演算処理に際しては、関数型の処理コマンドを用いて行なうものとし、次の2種類のコマンドを用意する。

*演算コマンド

変数名 = 主関数名 (パラメータ 1, パラメータ 2, ...)

*制御コマンド

制御コマンド名 パラメータ 1, パラメータ 2, ...

- ⑩ 各種コマンドの設定パラメータは数値だけでなく変数名も併せて使用できること。
 - ⑪ 予めユーザーが作成登録してあるコマンド群を実行できるマクロコマンドを備えていること。
 - ⑫ 通常のプログラムのように配列を意識しなくても、複数の時系列データを用いた演算が可能なこと。(自動的に同一絶対時刻のデータを用いて演算する)
- さらに、⑨のコマンドとしては、第 4.4.1 表に示すものが必要と考えられる。

表 4.4.1(a)表 主要演算コマンド一覧

No.	主関数名	制 御 パ ラ メ ー タ						機 能 概 要
		1	2	3	4	5	6	
1	PPNT	PID No.	{次元}					指定PIDデータ取得
2	PMAX	"	次元	{ts}	{te}			指定PIDの最大値検索
3	PMIN	"	次元	{ts}	{te}			指定PIDの最小値検索
4	PSMO	"	次元	種別*1	dt	{ts}	{te}	指定PIDの補間
5	PFIL	"	次元	τ	{ts}	{te}		指定PIDの一次遅れフィルタ
6	PINT	"	次元	{dt}	{ts}	{te}		指定PIDの積分
7	PSBN	"	次元	{ts}	{te}		-	指定PIDの差分
8	PDIF	"	次元	{dt}	{ts}	{te}		指定PIDの微分
9	PSPEC	PID No.	次元	種別*2				指定PIDのパワースペクトル
10	PMAS	PID No.	{区間}					Yield区間のカウント取得
11	PTSR	PID No.	{区間}	{ts}	{te}			指定PIDデータ取得
12	PPRE	"						プレプロ時間軸取得
13	SPEC	変数	種別*2					指定変数のパワースペクトル
14	PDSC	PID No.						指定PID放電条件取得
15	PSS	PID No.	{Case No.}					指定PIDショットサマリ取得
16	PDT	"						指定PIDの収集間隔時間取得
17	PTS	"						指定PIDの収集開始時間取得
18	PTE	"						指定PIDの収集終了時間取得
19	PNUM	"						指定PIDのデータ数取得
20	MAX	変数	{ts}	{te}				指定変数の最大値検索
21	MIN	変数	{ts}	{te}				指定変数の最小値検索
22	SMO	"	種別*1	dt	{ts}	{te}		指定変数の補間
23	FIL	"	τ	{ts}	{te}			指定変数の一次遅れフィルタ
24	INT	"	{dt}	{ts}	{te}			指定変数の積分
25	SBN	"	{dt}	{ts}	{te}			指定変数の差分
26	DIF	"	{dt}	{ts}	{te}			指定変数の微分
27	**	変数1	変数2					変数1の(変数2)乗
28	+	変数1	変数2					変数1と変数2の加算
29	-	"	"					変数1から変数2を減算
30	*	"	"					変数1と変数2の乗算
31	/	変数1	変数2					変数1を変数2で除算
32	=	変数	{ts}	{te}				変数の代入
33	MACH	変数1	変数2	種別*1				変数(2)を変数(1)で展開
34	IMG	変数1	変数2					変数1:実数, 変数2:虚数

*1) 種別: 1=直線補間, 2=スプライン補間, *2) 種別: 1=FFT(sin), 2=FFT(cos)
表以外に基本関数は使用可能である。

第4.4.1(b)表 制御コマンド一覧

No.	主関数名	制御パラメータ	機能概要
1	SHOT	C mmmmmm - C nnnnnn (ショット番号)	ショット番号指定
2	BASE	PID番号	時間軸の基準となるPID指定
3	FCOPY	変数(A0~Z9), 新規作成ファイル名	変数格納ファイルを複写
4	LISTV	[変数(A0~Z9)]	変数一覧の表示
5	MACRO	変数(A0), ……変数(Z9)	マクロコマンド上の引数の定義
6	FREE	変数(A0), ……変数(Z9)	変数領域の解散
7	MALOC	図形処理マクロ登録ファイル名(メンバー名)	マクロコマンドファイル指定
8	EXEC		マクロコマンドの実行
9	LOGLST		入力コマンドのロギング表示
10	HELP	[主関数名]	指定演算のガイダンス
11	ENDST		機能の終了
12	SET	変数1, 変数2	時間軸パラメータの値の代入

4.5 運用管理機能とデータベース結合・追加・削除機能

運用管理者がデータベースを維持管理していくことで始めて、その保全が確実なものとなることとして、その作業を円滑・確実に行うために支援ツールを準備する必要があることは、前にも述べたが、本章では、その処理の具体内容を示すことにする。また特に結合・追加・削除機能は所定のフォーマットで作られているデータをデータベースの構造に創成することまでも含んでおり、任意のデータをデータベース化するという自由度を持っているものである。

4.5.1 運用管理機能

運用管理者（一部一般ユーザ）向けの支援ツールの処理項目と内容は以下の通りである。

(1) 実験データベース退避処理

本処理は、実験データベース創成機能にて磁気ディスク（もしくはCLS*）上に作成された実験データベースを磁気テープに退避する際の運用管理者のマンマシン処理を支援するもので、TSS漢字端末上にてオペレーションが行なえるものである。端末でオペレーションができる点は以下も同様である。

(2) 実験データベース復元処理

本処理は、実験データベース退避処理にて退避された実験データベースを磁気ディスク上に再現する際の運用管理者のマンマシン処理を支援するものである。

(3) 実験データベース削除処理

本処理は、既に作成済みの実験データベースの削除を行なう際の運用管理者のマンマシン処理を支援するものである。

(4) 実験データベース現状把握処理

本処理は、現在実験データベースに登録されているショットに関する各種データの一覧を把握するためのマンマシン処理を支援するもので、運用管理者及び一般ユーザーの両者を対象とする処理である。

本処理は、TSS端末上での会話形式でのオペレーションだけでなく、NLP（日本語ラインプリンター）上にも出力ができる構成とし、次に示すような内容が出力できる処理である。

- ① 全てのショット番号一覧表
- ② DASD上に存在するショット番号一覧表
- ③ 磁気テープ格納済みショット番号一覧表
- ④ 指定されたショット番号一覧表
- ⑤ 指定された作成年月日YYMMDD(1)～YYMMDD(2)のショット番号一覧表
- ⑥ 磁気テープ未格納ショット番号一覧表
- ⑦ 放電方式毎でのショット一覧表

(5) 資格検査支援処理

本処理は、実験データベースのデータ保護を行なうために設けたデータ・アクセス・レベル設定テーブル及びユーザー・アクセス・レベル設定テーブル登録、修正に係わる以下の作業を行う際の運用

* CLS : Cartridge Library System ; ディスクと接続される大容量記憶装置。

管理者のマシン処理を支援する処理である。

- ① 新規利用者登録
 - ② 新規 P I D 追加登録
 - ③ ユーザーアクセスレベル登録, 修正
 - ④ データアクセスレベル登録, 修正
 - ⑤ 実験 DB登録済データのアクセスレベル修正処理
 - ⑥ ユーザー, データアクセスレベル表示
- (6) ショット毎のコメント入力処理支援処理

本処理は、既に作成された実験データベースの予め設定されているショット毎コメント入力領域に対して、コメントデータ設定を行なう際の運用管理者のマシン処理を支援するものである。

4.5.2 データベース結合・削除・追加機能

本機能は、既に作成されている実験データベースに対して、新規にデータ（各設備計算機からオンライン/オフラインにて取得, 変換, 編集したデータ）を結合追加登録したり, 既に登録されているデータの削除を行なう場合に用いる運用管理者向けの機能で, 通常のショット番号+ P I D 番号付きのデータを対象とする。

結合・削除・追加機能は, 次の2処理から成る。

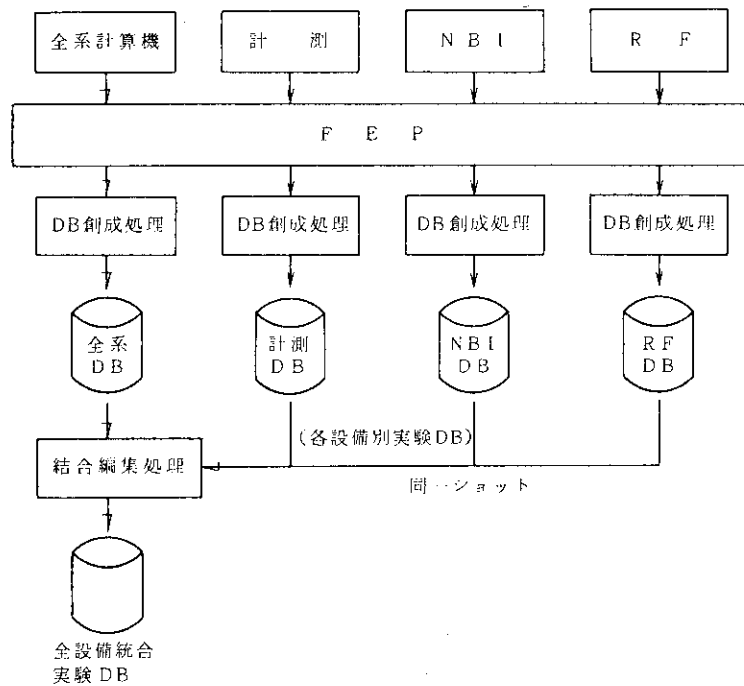
(1) 実験データベース結合編集処理

親実験データベースと各設備にて作成したデータベースとを結合編集するための機能である。尚, 各設備にて作成したデータベースは本実験データベースと同一の形式を有するものとする。(第4.5.1図参照)汎用ハンドリングパッケージは, このような複数設備のデータベースの存在を前提にして作ることにしており, ユーザはこの構造を意識する必要は無い。また, 記憶媒体の特殊性から, 小分割されていることが都合の良い場合もあることを述べておく。F E P における記憶媒体としては, ディスクと C L S (大容量記憶装置)とがあり, ディスク上のファイルに対しては常時, 書き込み/読み出し可能であるが, C L S 上のファイルは, まずその C L S に対応したディスクの空きエリアに一度転送 (recall) してから書き込み/読み出しが可能となる。この recall やディスクから C L S への転送 (migrate) は, ファイルの容量が小さいほど身動きが楽なことは言うまでもなく, その意味で小分割されている方が有利と言える。但し, 小ファイル数が大すぎても管理が大変であるし, また, 良く一緒に使うデータが異なるファイルに存在しているのでは, 使い勝手が悪い訳であるから, その分割の単位は, 発生元や使い方の異なるデータの集合であることが必要である。

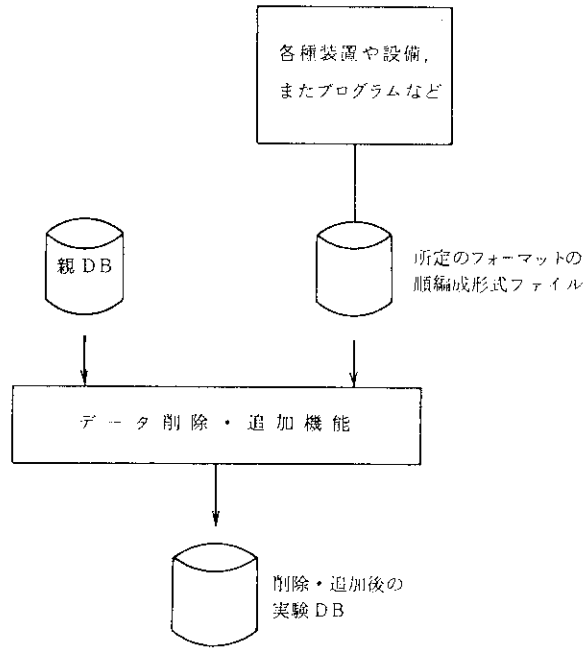
(2) データ削除・追加処理

所定のフォーマットで作成された順編成形式ファイルで行う追加登録及び削除処理で、親実験データベースに登録済みのデータの修正を行うことを第1目的とした処理である。削除の場合は、削除の指示と削除するデータのショット番号・PID番号とを所定のフォーマットで作成して処理を起動することにした。追加の場合には、追加の指示と追加するデータのショット番号・PID番号、さらにデータを所定のフォーマットで作成し処理を起動するという具合である。(第4.5.2図参照)

また、本機能の第2番目の使い方は、第4.5.2図において親データベースが無いとした場合の追加処理である。この場合は、所定のフォーマットのファイルを実験データベースに変換すること、即ち、データベース創成を行うことである。この使い方は如何なるデータもフォーマットさえ整えばデータベース化できることを意味し、極めて利用価値の高い処理である。一旦、データベースができてしまえば、結合編集処理を使って統合することも、又そのままデータベースとして利用することもできる訳である。



第4.5.1図 データベース結合編集処理概略図



第 4.5.2 図 データ削除・追加処理概略図

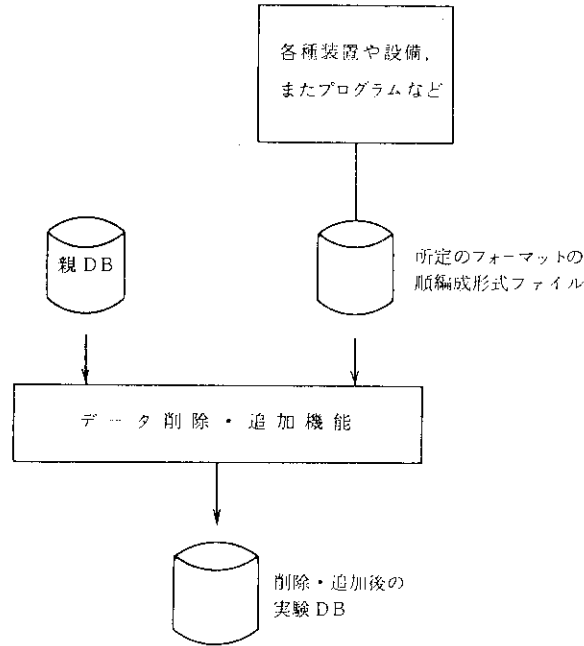
5. あとがき

本データベースシステムは、JT-60の実験データの効率の良い利用を目指して、昭和61年6月から仮運用、昭和62年1月から本運用を開始した。この製作に際し様々な書類を作り、またその運用に先立って、利用のための各種書類が利用者へ配られたことは言うまでも無いが、あとがきとして、これら書類の重要性について一言述べることとする。

プログラムの製作などに於いて良く見受けられるのは、コーディングが先行し、それに辻褄を合わせるが如くフローチャートや機能設計書が作られるというやり方である。これは、コードと書類とで確実に対応がとれていれば問題無いが、最終生産物を先に直してしまえば、その担当者にとって見た時、書類を修正するのは後戻りの作業になるため、なおざりになり易い。書類がなおざりになると、保守の能率が悪くなるばかりか、場合によってはコードの品質すら下げってしまうことにもなる。我々はこの教訓から書類を全てに優先させるという原則を徹底させた。

特に利用者向けの書類には、より一層の注意を払って作成した。如何に良いコードを作ったとしてもその利用マニュアルが杜撰だとその良さが発揮されないことになる場合が往々にしてあるからである。我々は、このような考えから、利用マニュアル、データ一覧などの書類は、全て活字化し、利用の便を考えると伴に今後加えられるであろう改造等に対応できるよう改訂番号管理を行うこととした。

通常軽んじられる書類化が、実は最も重要な仕事の一つであることを繰り返して本稿を終る。



第4.5.2図 データ削除・追加処理概略図

5. あとがき

本データベースシステムは、JT-60の実験データの効率の良い利用を目指して、昭和61年6月から仮運用、昭和62年1月から本運用を開始した。この製作に際し様々な書類を作り、またその運用に先立って、利用のための各種書類が利用者へ配られたことは言うまでも無いが、あとがきとして、これら書類の重要性について一言述べることとする。

プログラムの製作などに於いて良く見受けられるのは、コーディングが先行し、それに辻褄を合わせるが如くフローチャートや機能設計書が作られるというやり方である。これは、コードと書類とで確実に対応がとれていれば問題無いが、最終生産物を先に直してしまえば、その担当者にとって見た時、書類を修正するのは後戻りの作業になるため、なおざりになり易い。書類がなおざりになると、保守の能率が悪くなるばかりか、場合によってはコードの品質すら下げってしまうことにもなる。我々はこの教訓から書類を全てに優先させるという原則を徹底させた。

特に利用者向けの書類には、より一層の注意を払って作成した。如何に良いコードを作ったとしてもその利用マニュアルが杜撰だとその良さが発揮されないことになる場合が往々にしてあるからである。我々は、このような考えから、利用マニュアル、データ一覧などの書類は、全て活字化し、利用の便を考えると伴に今後加えられるであろう改造等に対応できるよう改訂番号管理を行うこととした。

通常軽んじられる書類化が、実は最も重要な仕事の一つであることを繰り返して本稿を終る。

謝 辞

全系実験データベース構築の作業を積極的に進めることを提言し各種の助言を下された原研那珂研究所 JT-60 試験部第 2 試験室長 近藤育朗氏, またデータベースの運用に便宜を図って下さったデータベース運用管理委員会前座長 臨界プラズマ研究部計画室長 岸本浩氏, 現座長同室計画調整グループリーダー 関省吾氏に感謝の意を表します。また, 本作業の全般の便宜を賜わった JT-60 試験部長 飯島勉氏に御礼申し上げます。最後に, 各種議論に参加して下さいました制御グループの諸氏に感謝します。

参 考 文 献

- (1) D. C. Tschritzis, F. H. Lochovsky (齊藤忠夫訳)
「コンピュータサイエンス研究書シリーズ — データベース管理システム」 日本コンピュータ協会出版
- (2) G. J. Myers (松尾正信訳) 「ソフトウェアテストの技法」 近代科学社
- (3) 伊藤康浩, 栗原研一, 木村豊秋 JAERI-M 87-098 「JT-60 全系実験 データベース構築 (II) — 機能マニュアル —

謝 辞

全系実験データベース構築の作業を積極的に進めることを提言し各種の助言を下された原研那珂研究所 JT-60 試験部第 2 試験室長 近藤育朗氏, またデータベースの運用に便宜を図って下さったデータベース運用管理委員会前座長 臨界プラズマ研究部計画室長 岸本浩氏, 現座長同室計画調整グループリーダー 関省吾氏に感謝の意を表します。また, 本作業の全般の便宜を賜わった JT-60 試験部長 飯島勉氏に御礼申し上げます。最後に, 各種議論に参加して下さいました制御グループの諸氏に感謝します。

参 考 文 献

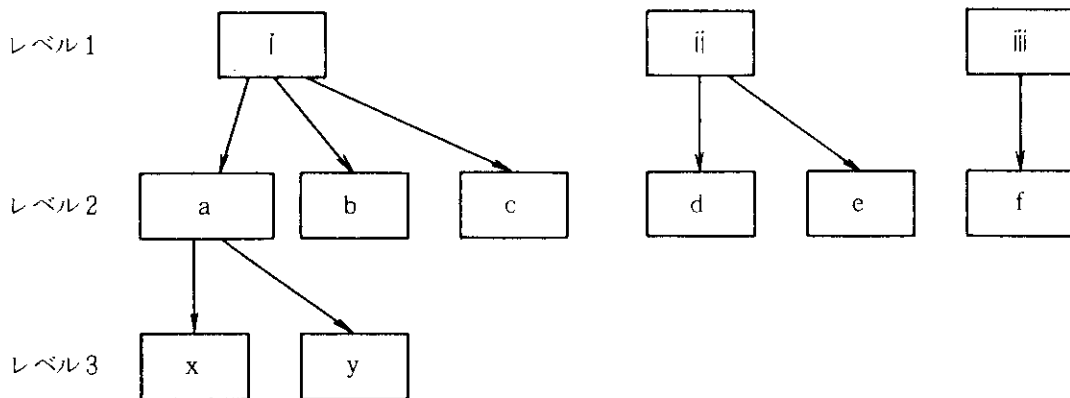
- (1) D. C. Tschritzis, F. H. Lochovsky (齊藤忠夫訳)
「コンピュータサイエンス研究書シリーズ —— データベース管理システム」 日本コンピュータ協会出版
- (2) G. J. Myers (松尾正信訳) 「ソフトウェアテストの技法」 近代科学社
- (3) 伊藤康浩, 栗原研一, 木村豊秋 JAERI-M 87-098 「JT-60 全系実験データベース構築 (II) —— 機能マニュアル ——

付録1 データモデル

本付録では、通常使用されるデータベースのデータモデルについて触れておく。詳細は、数多く出版されているデータベースの参考書を参照のこと。

付1.1 階層型データモデル

本構造（ツリー構造）とも呼ばれ、付第1.1図のような構造をもつ。データの関係として親（上位レベル）が複数の子（下位レベル）をもつことが可能で、子は複数の親をもたない。このモデルは非常に理解しやすく、比較的少数の覚えやすいコマンドで操作することができ、許される関係のタイプには制限があるため他のモデルより実現が容易であるが、逆に課せられる制限からデータの構造が不自然になってしまう場合がある。



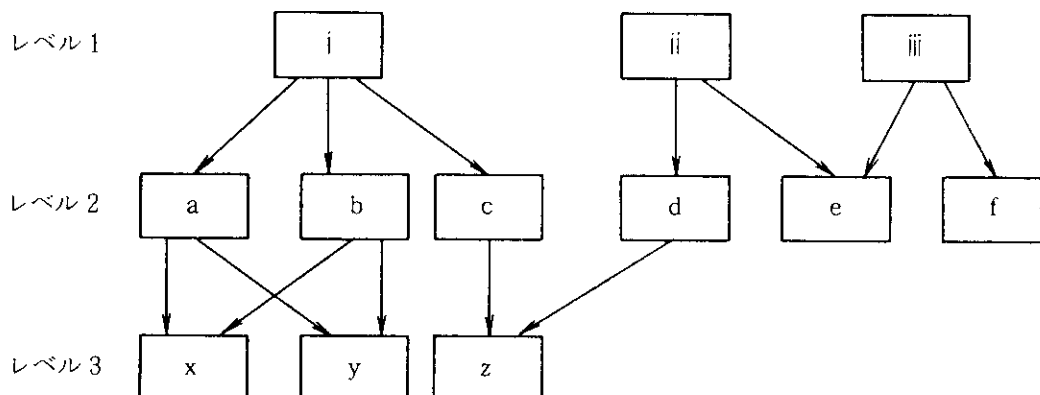
付第1.1図 階層型データモデル

付1.2 ネットワーク型データモデル

メッシュ型にデータ関係をとらえたもので、付第1.2図のような構造をもつ。親子関係がクロスすることができる。

階層型モデルでは親子関係が1:Nであり、構造に制限があるが、ネットワーク型データモデルではデータごとの関係には制限がなく、N:Mの関係を持っている。従って、それぞれの関係が複雑になるため、更新や削除に弱い。

このモデルは定型的な処理に向いている。また、データベース設計においては、N:Mの関係を1:Mの関係に置き換えて考えることが多い。

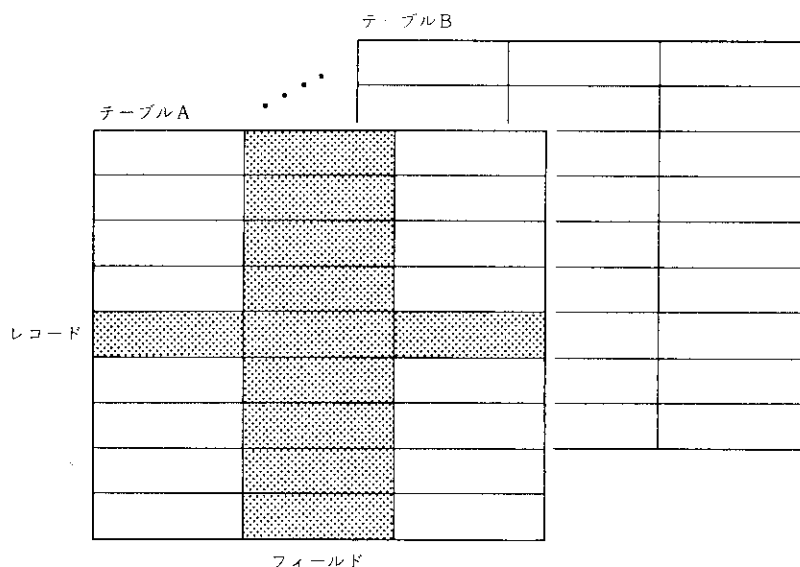


付 第 1.2 図 ネットワーク型データモデル

付 1.3 リレーショナル型データモデル

数字における関係（リレーション）という概念を応用したものであり、表現方法はテーブルと呼ばれる二次元の表形式をとっている（付第 1.3 図）。データ間の関係付けはデータの値によって行う単純データ構造である。削除・更新が容易である。

適用上の指針を比較してみると、階層型データモデルやネットワーク型データモデルが定型業務に向いているのに対し、リレーショナル型データモデルは、非定型業務に向いている。



付 第 1.3 図 リレーショナル型データモデル

付録2 何故汎用的なデータベースを使用しないという結論を出したか (PLANNER, AXEL IIの比較)

世の中にデータベースソフトと言われているものが多くあるが、(株)富士通製のそれについて本節で比較を行う。(尚 VSAMについては、第3.2.2項で述べてある。)FACOM M-380に乗る汎用的なデータベースシステムとしては、計画管理情報システム(PLANNER)と会話型データ解析システム(AXEL II)がある。

計画管理情報システム(PLANNER: PLANNing and management information system based on Easy RDB)は、経営情報管理、品質管理解析、環境データ管理など幅広い分野の業務に適し、時系列データなど各種の形式のデータを統合的に管理できる。会話型データ解析システム(AXEL II: An Extensible Engineers Language II)は、技術分野における研究者・技術者のデータ解析作業などの効率向上を目的としており、実験データ、技術管理データなどを統合管理し、検索・比較・解析あるいは日本語報告書の作成などが容易に行える。

これらのデータベース管理システムの比較を付第2.1表にまとめる。

付第 2.1(a)表 PLANNERとAXEL IIとの比較

項 目		PLANNER	AXEL II
特 徴		<ul style="list-style-type: none"> ・経営情報管理、品質管理解析など幅広い分野の業務に適す ・テーブル表現による柔軟な操作ができる。 ・ANALYSTなどの他のDBMSと連携が可能で、統計処理や図形処理が充実している。 ・パソコンとの連携が可能である 	<ul style="list-style-type: none"> ・技術分野におけるデータ解析に適す ・テーブル表現による柔軟な操作ができる ・三次元データの曲面表示が可能である ・会話型のきめ細かい図形処理ができる ・定型化した処理をマクロ化できる
データベースモデル		リレーショナル型	リレーショナル型
設 計 と 運 用	利用形態	<ul style="list-style-type: none"> ・TSS ・バッチ 	<ul style="list-style-type: none"> ・TSS ・バッチ
	保守管理	<ul style="list-style-type: none"> ・ユーティリティ完備 退避/復元/診断情報印刷 	<ul style="list-style-type: none"> ・ユーティリティ完備 退避/復元/登録情報印刷
	排他制御	<ul style="list-style-type: none"> ・コマンドごとに、データベースのファイル単位で排他制御 	なし
	機密保護	<ul style="list-style-type: none"> ・パスワード方式 (MASTER/READ&WRITE/READ) ・データベース/テーブル単位 	なし
	拡張性	<ul style="list-style-type: none"> ・ユーザプログラムにより、機能拡張 	<ul style="list-style-type: none"> ・ユーザプログラムにより、機能拡張
	他システムとのインターフェース	<ul style="list-style-type: none"> ・ANALYST (統計データ処理パッケージ) ・MDS-IV (経営意思決定サポートシステム) 	なし
	必要資源	<ul style="list-style-type: none"> ・CPU・メモリともに大 (メモリ1MB以上) 	<ul style="list-style-type: none"> ・CPU・メモリともに大 (メモリ2MB以上)
デ ー タ 操 作	コマンド	<ul style="list-style-type: none"> ・テーブル創成 (ファイル、端末からのデータ入力) ・検索 ・更新・削除 ・データ値の表示 ・加工編集 ・データ保存 ・図形処理 ・統計解析 ・地図出力 ・帳票出力 	<ul style="list-style-type: none"> ・テーブル創成 (ファイル、端末からのデータ入力) ・検索 ・更新・削除 ・データ値の表示 ・加工編集 ・データ保存 ・図形処理 ・統計解析 ・画面情報操作 ・レポート編集

付第2.1(b)表 PLANNERとAXEL IIとの比較

項 目		PLANNER	AXEL II
デ ィ タ 操 作	コマンド	<ul style="list-style-type: none"> 画面情報操作 ビュー定義と利用 メニュー処理 など 	<ul style="list-style-type: none"> マクロ作成 プログラミング言語 など
	検索・加工	<ul style="list-style-type: none"> 検索結果の絞り込み テーブルの併合 合計値の算出 など 	<ul style="list-style-type: none"> 検索結果の絞り込み テーブルの併合 合計値の算出 など
	統計処理	<ul style="list-style-type: none"> 最大、最小など基本統計量の算出 相関係数の算出など ANALYSTによる統計処理 	<ul style="list-style-type: none"> 最大、最小など基本統計量の算出 度数集計など
	報告書	<ul style="list-style-type: none"> 画面情報の保存・表示・印刷 	<ul style="list-style-type: none"> 画面情報の保存・表示・印刷
	日本語処理	<ul style="list-style-type: none"> メッセージ・コメント 問合せ検索 更新・表示 	<ul style="list-style-type: none"> メッセージ 文書表示
	メニュー処理	有 り	な し
	グラフィック機能	<ul style="list-style-type: none"> 折れ線グラフ、円グラフなどの基本グラフをサポート グラフィック端末 富士通 { F9430 シリーズ F6653 	<ul style="list-style-type: none"> グラフ作成コマンドがきめ細かく用意されている グラフィック端末 F6240 シリーズ
応用システム インターフェース	<ul style="list-style-type: none"> COBOL, PL/I, FORTRAN インターフェース 	<ul style="list-style-type: none"> FORTRAN インターフェース 	
デ ィ タ ベ ィ ス 定 義	入力媒体	<ul style="list-style-type: none"> カード 磁気ディスク 磁気テープ TSS 端末 	<ul style="list-style-type: none"> カード 磁気ディスク 磁気テープ TSS 端末
	取扱うデータの性質	<ul style="list-style-type: none"> 数字データ 16桁まで (整数, 実数) 文字データ 64文字まで 日本語データ 32文字まで 	<ul style="list-style-type: none"> 数字データ (整数, 実数) } FORTRANでサポートされる範囲 文字データ
	単 位	<ul style="list-style-type: none"> 入力時にフィールドごとに単位指定 	<ul style="list-style-type: none"> 入力時にフィールドごとに単位指定

PLANNER および AXEL II はリレーショナル型データベースなので実験データのリレーショナル型データベースへの適用とそれによる特徴を調べてみる。リレーショナル型データベースでは、データはテーブル形式で表現される。従って、リレーショナル型データベースを適用するために、データをテーブル化しなければならない。ここでは、次のような1点データと時系列データについて考える。

〔1点データ〕

略称	単位 略称	フォーマット	データ ステータス	データ タイプ	データ
				データ部	
付加情報部					

〔時系列データ〕

略称	単位 略称	データ ステータス	データ タイプ	データ点数	収集開始 時間	サンプリ ングピッチ	付加情報部
データ (1, 1)				データ (1, 10)	データ部	
データ (1000, 1)				データ (1000, 10)		

(この例では時系列データは、10個の計測器において1000回サンプリングした場合のデータで(1000×10)の2次元配列とする。)

付第2.1図 実験データの例

リレーショナル型データベースで表現されるテーブルは、テーブル名が付けられる。テーブルの縦方向(列)は、フィールドと呼ばれ、フィールド名が付けられる。フィールドは属性(単位)をもち、同一の意味を持つデータが格納される。

テーブルの横方向(行)は、レコードと呼ばれ、ある時点におけるデータあるいは一連の関係があるデータなど横方向に関係を持っている。

(1) 適用するためには

1点データと時系列データとはデータ部が異なるため、まとめて1つのテーブルにすることは難しい。そこで1点データと時系列データを分けてテーブル化することにする。

(a) 1点データの場合

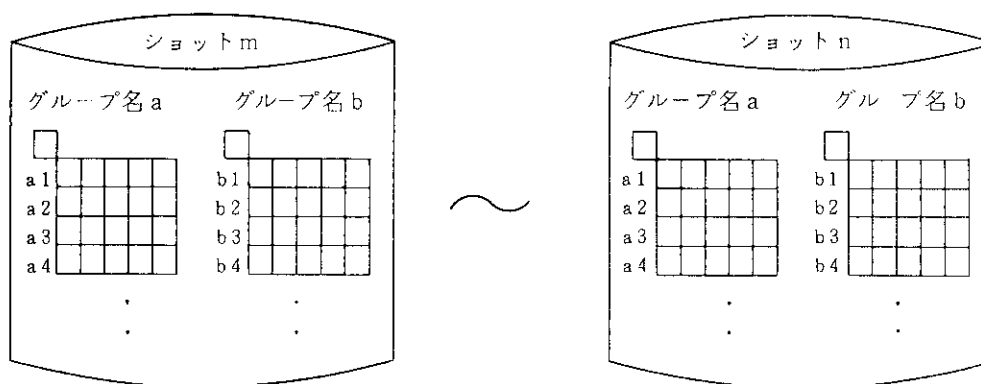
1点データをテーブル化するためには、1ショットについて同一の形式あるいは意味を持つものをグループ化してグループごとにテーブルを作成する方法と1データ項目ずつテーブルを作成してショットが発生する都度追加する方法が考えられる。しかし、実験データは、ショットごとに発生することやデータ項目数が極端に多いことなどの特徴から、1データ項目ずつテーブル化する方法では、テーブルが極端に多くなることや過去データの復元を考えると、利用および運用管理が難しいと考えられる。従って、1ショットごとにグループ化されたテーブルを作成する方法が適してい

ると考えられる。この方法では、ショットごとにデータベースを構築することになると考えられる。

ショットごとに実験 DBを作成する場合、データベース名はショットが区別できるように付ける必要がある。これは、PLANNERあるいはAXEL IIを利用するとき、検索するデータベースを指定する必要があり、検索するショットがどのデータベースに格納されているかを分かり易くするためである。

データベースに格納するテーブルは、付加情報部およびデータ部の形式が一致し、相互に関連があるものをグループ化し、データベース内で一意になるようにグループ名を付ける。このグループ名は、利用するとき指定する必要があるため、利用者が自分で覚えておかなければならない。また、テーブルには、データ項目名が分かるようなフィールドを定義し、検索キーとなるように工夫する必要がある。

ショット単位で実験 DBを作成する場合、付第 2.2 図のようになる。



付第 2.2 図 一点データのテーブル例 - ショット単位で実験 DB作成

(b) 時系列データの場合

付加情報部のレコードは、1項目について1つずつであるが、データ部はサンプリング数分できるため、これらを1つのテーブルにすることは好ましくない。従って付加情報のテーブルは、一点データのようにグループ化してテーブルを作成し、データ部は、データ項目ごとにテーブル化することが考えられる。

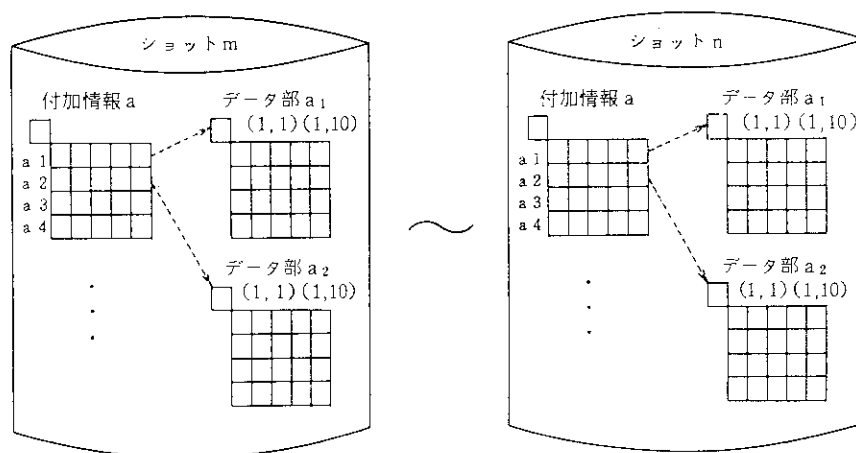
時系列データの付加情報部の形式は、あるまとまりで統一されていると考えられ、一点データのようにグループ化してテーブルを作成できる。しかし、データ部は、サンプリングピッチによりデータ量が変わると考えられるが、計測点の個数は変わらないと考えられ、計測点をフィールドとして配列をテーブル化することができる。

従って、時系列データについては、付加情報部とデータ部とを分けてテーブルを作成することになると考えられる。

この場合、データベースの構築単位は、ショットごと、あるいはショットごとかつ設備ごとになると考えられる。データベース名の付け方は、一点データと同様である。付加情報部のテーブル名は、グループ名となり、また、テーブルには、PID番号などのデータ項目名が分かるフィールドを設ける必要がある。データ部のテーブル名は、データ項目名となるが、PID番号のような数値のものは、

英数字の名前を付ける必要があり、利用者は、その名前により検索を行うため、覚えておく必要がある。また、フィールド名は、各配列となるが、これにも名前を付ける必要がある。

時系列データについてショットごとに実験 DBを作成する場合、付第 2.3 図のようになる。



付第 2.3 図 時系列データのテーブル例 ショット単位で実験 DB 作成

(2) 適用したときの特徴 (PLANNER および AXEL II)

(a) 入力

1点データおよび時系列データの付加情報部については、グループごとにまとめてテーブルになるように編集する必要がある。また、時系列データのデータ部については、データ項目ごとに時系列になるように編集する必要がある。時系列データのデータ部については、PIDなどのデータ項目ごとに8文字以内のテーブル名をきめる必要があり、データ項目の数が極端に多いと管理が大変であり、データベース内で一意にするのも容易ではないと考えられる。一方、AXEL IIでは、テーブルごとにファイルが作成されるが、時系列データの数が極端に多いと、計算機システムのカタログ管理などに負荷がかかるため、適さないと考えられる。

(b) 利用

ショットの検索は、作成されているデータベースを探すことを意味し、会話形式で端末に表示することになる。また、応用ソフトウェアからのショット（作成されているデータベース）の検索をする手段は、サポートされていない。

データ項目ごとの付加情報の検索は、どのテーブルに格納されているかを利用者が覚えていなければならない。一点データのデータ部についても同様となるであろう。

時系列データのデータ部の検索は、データ項目ごとに定められたテーブル名を利用者が覚えておく必要があり、検索するデータ項目が多数ある場合、データ項目とテーブルとの対応がとれなくなると思われる。検索は、各テーブルのフィールドごとに行うことになるため、入力時のフィールドの定義には十分な検討が必要である。

図形表示処理は、AXEL IIでは三次元図の表示をサポートしている他に、XY軸作画などのきめ細かい処理ができるようになっている。PLANNERでは、定型的な図形を広範囲にサポート

し、地図などの出力もできる。

機密保護については、AXEL IIでは特に無いため、RACFによりテーブルごとに検査することになる。PLANNERでは、RACFの他にパスワード方式でテーブルごとに機密保護ができる。排他制御については、AXEL IIではサポートされていないため、入力作業は入力作業担当者が時間帯を選んで更新処理を行うことになる。

(c) 運用

PLANNERおよびAXEL IIは、データベースの退避/復元は、テーブルごとに行うことができ、また、利用状況を印刷するツールもあり、ともに運用のユーティリティは充実しているといえる。

このように見てくると、データの大多数を占める時系列データの場合、データ構造の面で極めて複雑になると共に、機密保護などでは管理が充分できないといった点で、両ソフト伴に敢えて導入に踏み切る要因が無いと判断した。