Vol. 30 特1　　電波研究所季報　　November 1984
pp. 223—235

IV-2 データベースシステムの構造

吉野泰造* 小池国正*

(昭和59年7月3日受理)

1. はじめに

最近のデータベース（以下“DB”と略記する）技術は長足の進歩を見せており，普及の度合いも急速である．さて，DB は1960年代に始まり，データ管理の方法，通信と結合した分散化等，さまざまな形態の発展を見せており，図書検索，在庫管理等に重宝がられている．これらを可能にしたのは計算機自身の進歩である．データアクセスの方法が，プログラム内埋め込み型（DATA 文）から，シーケンシャル・ファイル型，ランダム・アクセス・ファイル型をへて，DB に到っているが，この流れは，メモリ，磁気テープ，ディスクへと発展してきた計算機技術の発展の影響を受けており，生物学の「個体発生（データ利用技術）は系統発生（計算機技術）を繰り返す」という法則を想い起こさせる．

しかし，科学技術用に DB が適用された例はまだ少く，この方面でも化学薬品の検索のように数値演算用には使われていないのが現状である．理由として，小規模演算やデータの利用期間の短いケースでは，DB 構築の手間の方がはるかに多くなるからであろう．しかし VLBI では関連する分野が地球，天文等であり，データの価値が永続性をもつこと，データ処理解析システムが大規模となることなどにより，何らかの強力なデータ管理が必要となる．そして，これを DB 技術に求め K-3 では VLBI データの統合化を図った．このような例は解析専用に作られた米国 Mark-Ⅲ DB[1] に先例が見られるが，VLBI データを総合的に DB 化したのは今回が初めてである．これは電波研究所において，スケジューリングから実験データ処理・解析まですべての作業を一貫して行えるという環境に基づいている[2]．従って，K-3 システムのハードが「IEEE-IB」を柱としているように，ソフトの柱は「データベース」であると言えよう．

2. VLBI データベースのあり方[3]

DB はしばしば図書館にたとえられる．さまざまな図書検索サービス，迅速なアクセスのための図書分類，新着図書の登録等の図書管理業務をデータの維持管理になぞらえたものである．性質もよく似ており，敷地（ディスク・スペース）は広いほど良く，利用規定はルーズな方がユーザには喜こばれるが管理は危うくなる．逆に，ガードの固い規定では図書（データ）が死蔵化する．また，いずれも設計・製作とその維持・運用は大変で多数のユーザをかかえる時，長い目で見た時，その価値が評価されるのである．

VLBI におけるデータ利用は多人数によって行われ，応用ソフトも積極的に改訂が加えられていくため，データに関する無用の混乱を最小化し，データの取扱いを容易にするため DB 導入の必要があると考えた．即ち「長期的に見て発展性をもったものにする」という認識である．しかし，しばしば誤解されるように DB さえ導入すればあらゆるデータを格納し，あらゆる要求に対応できる，という幻想を抱きがちである．DB という言葉にある種の完璧性を期待してしまうのであるが，現実の DB はさまざまな制約を持ち，サイズの上からも使い勝手からも今後使用が考えられるすべてのデータに対応することは不可能であり，これを知らずに導入を図ると幻滅を感ずることになりかねない．そのため VLBI におけるデータ利用形態を正しく把握しておく必要がある．更に利用が想定される計算機およびデータ管理ソフトに関し，ユーティリティが便利なものであるほど，アクセスは遅くなり，逆に機械レベルに近ずいたサービスであるほど速いレスポンスが期待されることを想起しておく必要がある．

VLBI で DB に登録するデータ項目の種類は2つあり，一方は観測量や物理定数のように恒久的で物理的意味のあるもの，他方は大規模ソフトを動作させる時，特に必要となるフラグ，処理経過等のデータ識別用のものである．前者と後者は経済学でいう「ストック」と「フロー」の価値になぞらえることができると思われる．VLBI のデータ処理・解析を進める上で必ずしも前者だけのデータに重きを置けるものではない．

数の上で圧倒的に多い恒久的データの一例として，「2局間の遅延量」を考える．ある1つの観測量を指定するために，「実験コード」，「周波数バンド」，「観測番号」，「基線」，場合によってはさらに，「相関処理回数」，

* 鹿島支所　第三宇宙通信研究室

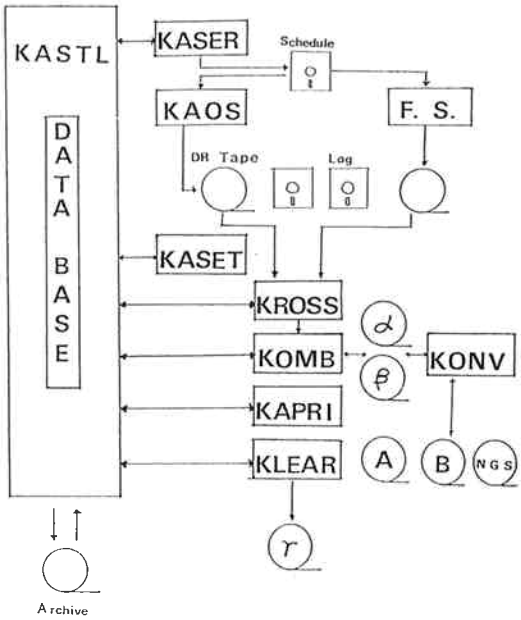
「バンド幅合成処理回数」，の合計6つの指定が必要となる．従ってVLBIの場合もデータを管理する際の考え方として『重複データを持ちやすく，取り違いを起こし易いデータ，また将来に亘ってシステム的データ管理が必要なデータをファイル形式による人的管理から脱脚させる』ことにより，データ管理の手間に十分見合うプログラム開発時の負担軽減を図ろうというものである．これによって，データの更新とプログラムの更新も全く独立に行われることになる．

さて，我々はいかにVLBIのDB構造を設計するか，という問題に直面する．DBは現実世界の計算機上への写像であるといわれる．これを3.に述べる「スキーマ」で定義する必要がある．設計上で注意すべきことは，VLBI演算において，またはユーザが頻繁にアクセスするデータを物理的性質や生成単位で分類することである．DBがそもそも集合論的に扱われるものであることから，この作業はどのようなDBでも必要なことである．

3. データベース設計

3.1 IMAGE/1000[4]

現在，急成長しているDBに関する商用パッケージの数は数十に上ると思われる[5]．これらは米国で発達したものであり，国産品はすべてその流れをくんでいると言える．しかし，通常これらの相互変換は困難であり，使用計算機の機種により大幅に限定されてしまう．各種DBの統一化もまだ遠いと言えよう．K-3で使用しているのはHewlett Packard社のHP-1000であるため，我々は同社のIMAGE/1000と呼ばれるDBソフトウェアを利用できた．これはネットワーク型と呼ばれるタイプで，ハッシング（後述）によるキーアクセスが特徴である．IMAGE/1000には会話型データアクセス・ユーティリティQUERY，プログラミング用データアクセス・ライブラリ，メインテナンス用ユーティリティ等のツールが豊富に用意されているが，これらについては本特集号「IV-4データベース運用ユーティリティ」において詳述される．K-3ではこれらIMAGE/1000のツールをふんだんに活用してDBを設計することにした．K-3ソフトウェア・システムは第1図に見るようにすべてのソフトからアクセスされるものであるため（KAOSだけは機動的に各局で運用されるものであるため，また，その性質によりDBから切り離されている），全ユーザが容易にDBを利用できることが望ましい．また，DBの何らかの変更に対し応用プログラムが影響を受けにくいこともデータ独立性の点から重要である．このため第2図に見るように（文献(6)参照）K-3ソフトは



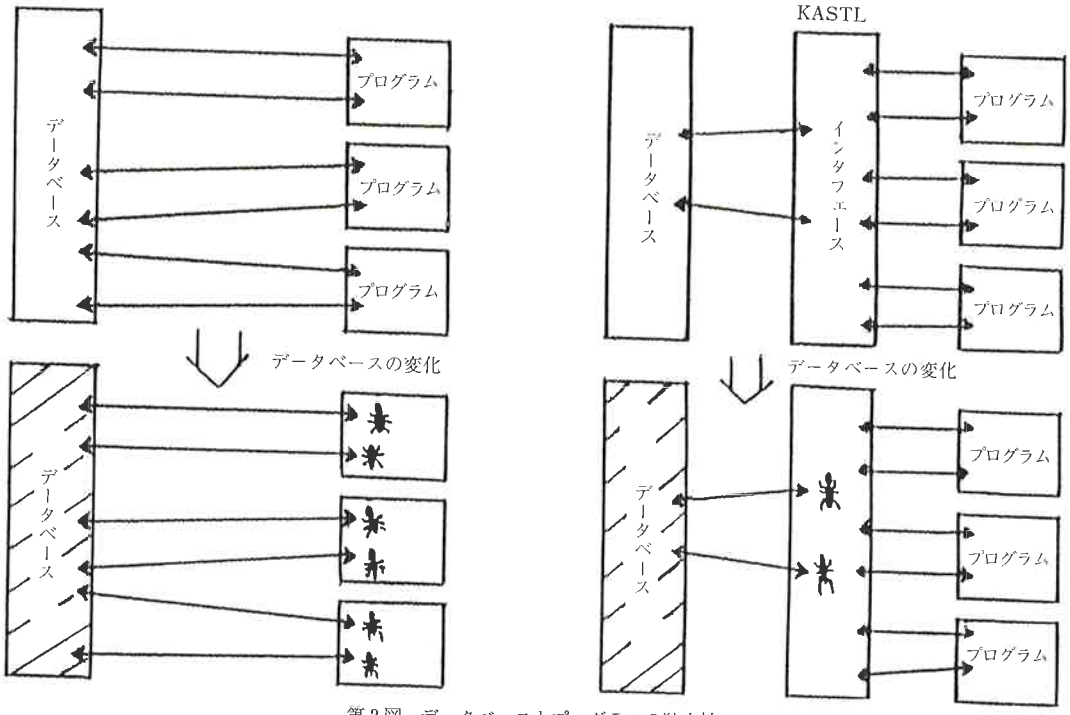
第1図 K-3 VLBIソフトウェア・システム

IMAGE/1000を直接アクセスするのではなく，利用するデータの集合をKASTLと呼ばれる仲介のソフトに伝えることによりDBへの登録，読出，抹消等ができるようにした．

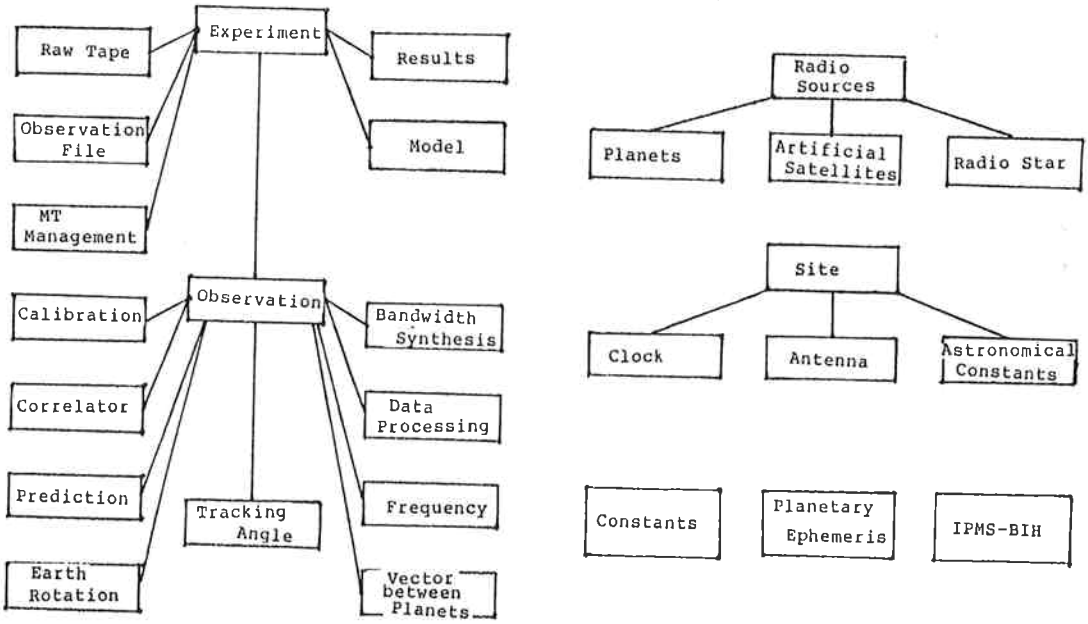
このようなインタフェースを設けることによってデータベースの何らかの変化に基く各々のプログラム中のバグ発生をこのインタフェースの中に局在化できる．従って，バグの発見が容易になり，かつ修正箇所も少くてすむ．この時，データの集合を指定するのが「アクセスキー」と呼ばれる5桁の数値である．整理すると，①IMAGE/1000の規約に基いてDBの中核を設計する，②本構造を計算機上にロードする（DBDS），③必要データをファイルから流し込んでおく(DBBLD)，④K-3ソフトはKASTLに番号指定（アクセスキー）を行い，KASTLは内部でIMAGE/1000のライブラリを活用して必要なデータ操作（登録，読出，抹消等）を行う，という手順である．IMAGE/1000は商用パッケージとして入手しているので，開発部分はDBの中核の構造を設計することと，KASTLのアクセスキー方式を確立することである．

3.2 スキーマ設計

DBの中核の構造設計は通常「スキーマ設計」と呼ばれ，採用すべきDB管理システムの選択についで重大な部分である．この段階の設計ミスは，全システムにとって死活問題となりかねないからである．この設計方法に関してのアプローチはさまざまな方向から試みられているが決定的なものはまだない[7]．目的はある属性の集



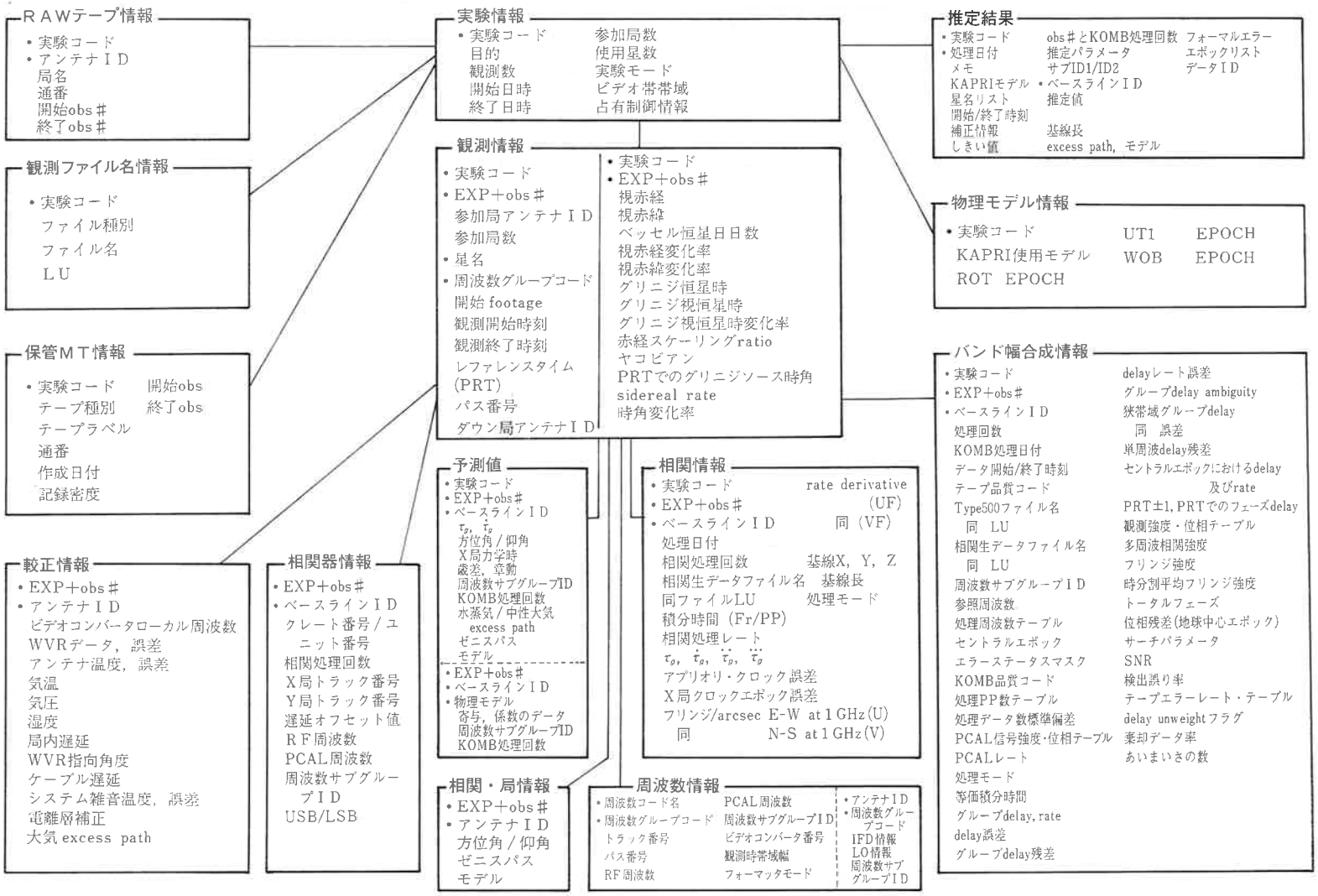
第2図 データベースとプログラムの独立性



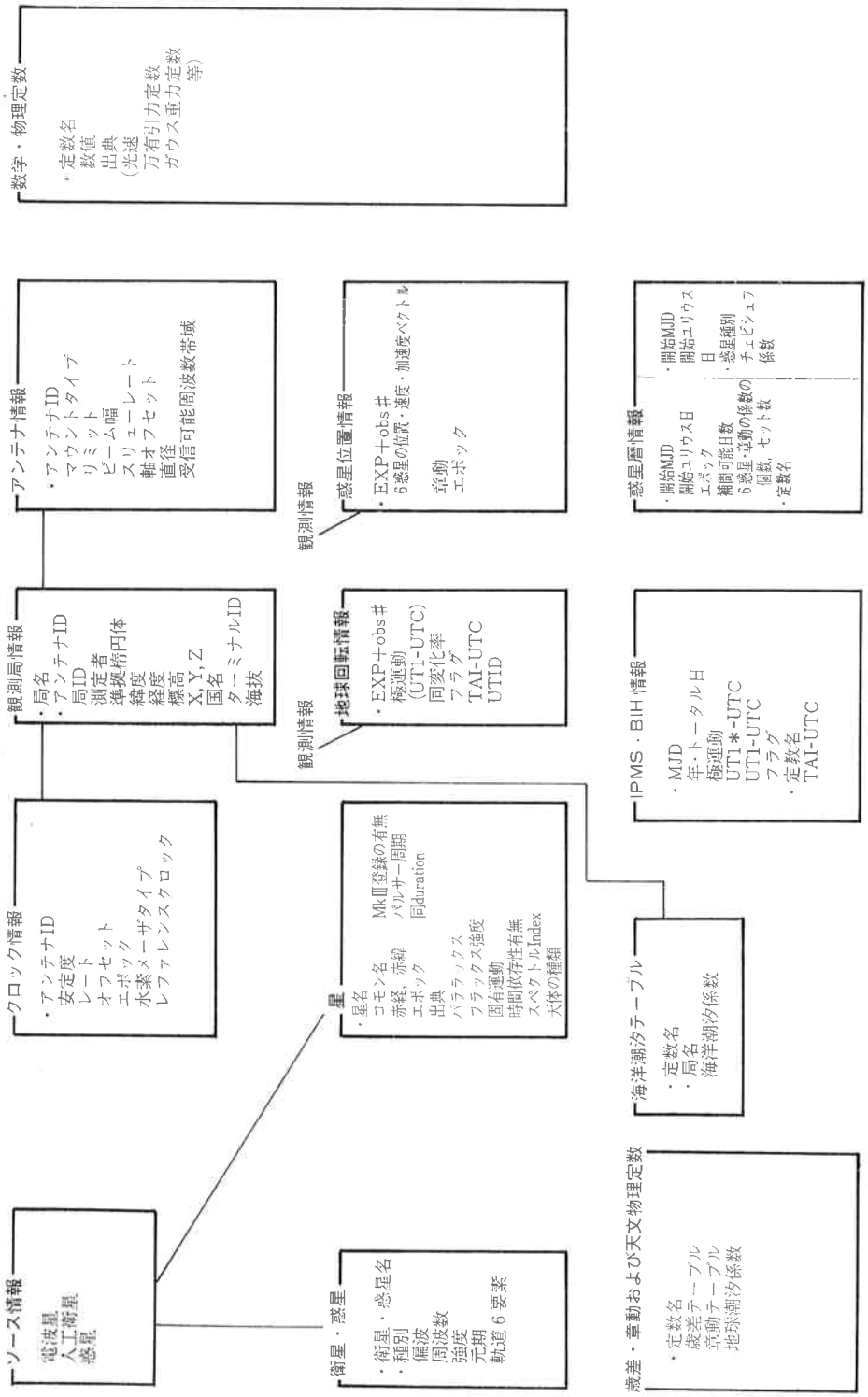
第3図 (a) 概念スキーマのデータセット

合と集合の間の結合をモデル化することである．我々はここで K-J 法[8] として知られているカードを用いたデータ間の関係づけ操作を行った．本来の K-J 法は新たな発想を求めるという色彩が強いと思われるが，ここではむしろデータ間の関係をとらえ適確に分類，結合を行うことに主眼がおかれた．

数人で VLBI に関するキーワードを出し合い，更に Mark-Ⅲ DB の項目を参考に追加，修正を行った．次に何百というキーワードをグルーピングした．分類した結果，観測局，電波源に固有な情報と定数を除き，ほぼ処理・解析の流れに沿ったものとなった．これを整理して概念スキーマを作成した（第3図）．各項目の追加，



(b) 概念スキーマ (データ項目)



を含む. ●印はキーアイテム)

第1表 VLBI データの生成単位

生成単位	例
実験	実験コード，観測数
観測	遅延量，観測開始時刻
観測局	観測局名，アンテナ座標
基線	遅延量，電離層補正量
1個	天文・物理定数
解析回数	解析日付，基線長
周波数	周波数バンド，観測バンド幅
日数	UT 1 情報，極運動情報
生テープ数	テープ通番，観測局名
KROSS ラン	相関処理回数，相関生データファイル名
KOMB ラン	遅延量，多周波相関強度
書込指定	基線長，使用物理モデル
種別	惑星暦のチェビシェフ係数，極運動 (IPMS/BIH)
登録数	惑星暦管理の開始ユリウス日

第2表 IMAGE/1000 使用変数

タイプ	ワード数	精度
実数 (R2)	2 W	6～7 桁
整数 (I1)	1 W	4～5 桁
文字 (Xn)	n/2 W	1文字 ASCII

(1 W=16 bit)

削除また，データ集合の発生単位に基づく再分割等は DB 完成に到るまで，K-3 ソフトの要求とからみ，さらに Mark-Ⅲ DB を参考に作業が継続された．データの生成単位は何種類かある．それを第1表に例とともに示した．ただし，生成単位は複合して現れるのが通例であり，例の中にも他の生成単位と組み合わさっているものがある（例えば，遅延量は「実験」，「観測」，「基線」，「処理回数」等の複合した生成単位となる）．

グルーピングの後，各項目の計算機上表現における型と必要精度を洗い出した．IMAGE/1000 は整数型，実数型，文字型の3種類を扱える．使用ワード数，精度を第2表にまとめた．しかし，第2表からもわかるように IMAGE/1000 は倍精度変数の取扱い機能を持っていないため，VLBI で必要とされる多くの有効桁数が表現しにくい．このため我々は必要な桁数だけ文字型でデータを格納することに決め，倍精度変数をも IMAGE/1000 で扱えるようにした．結果として，表現できる変数の桁数に上限がなくなったが若干のディスク容量とアクセス速度に負担がかかることになった．

さて，計算機上にのるスキーマを完成するには先のデータ集合を IMAGE/1000 の規約に従い記述していく必要がある．またユーザの持っているディスク領域の大きさと使用目的からくる要求領域の調整をとるため各部の物理的領域の定義も同時に行う必要がある．IMAGE/1000はデータ集合（データセット）を3種類に分けている．それらは

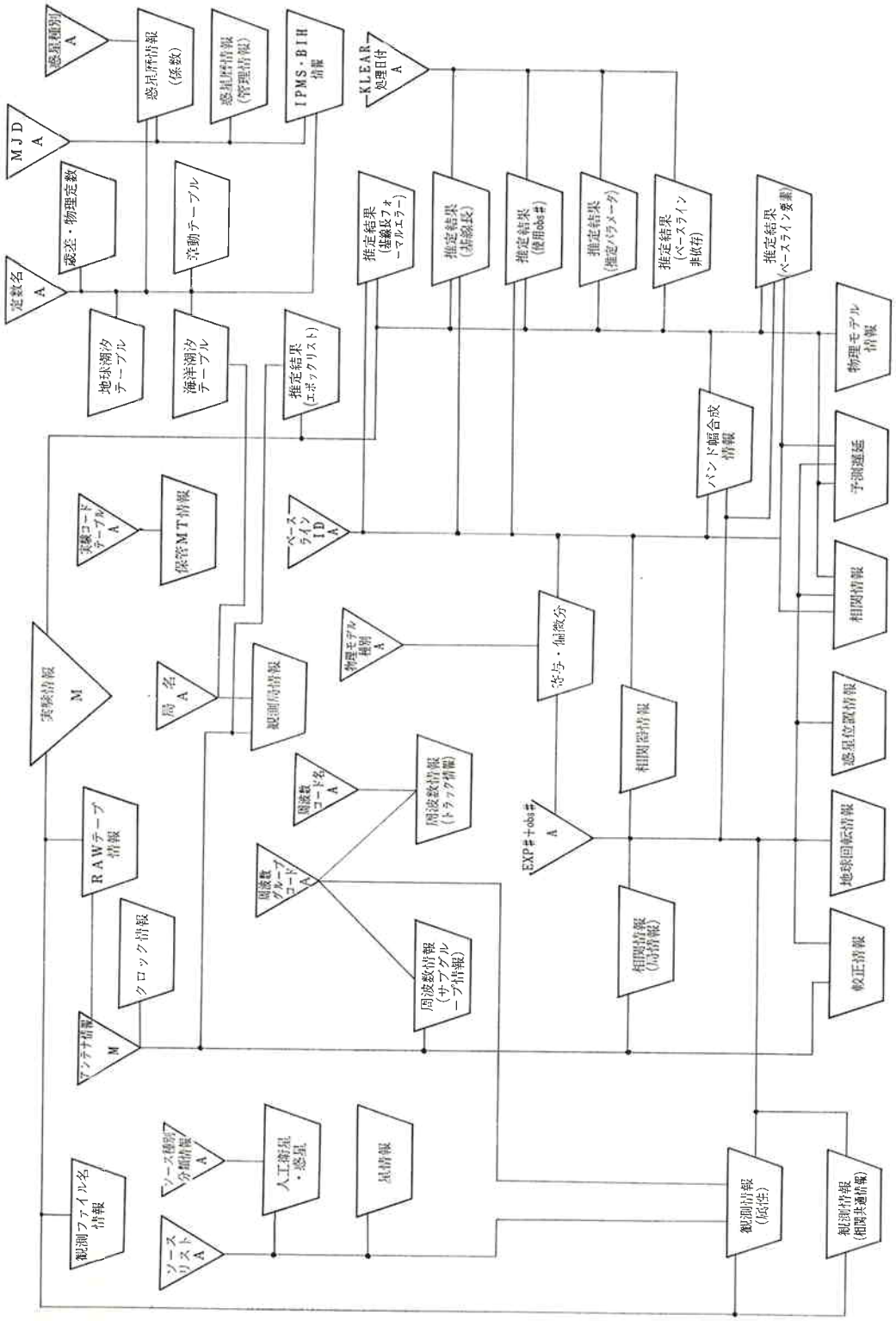
(1) Automatic Master Data Set
(2) Manual Master Data Set
(3) Detail Data Set

であり，適切なデータセット分類と共にデータアクセスの能率を決める要素となる．(1)の Automatic Master はハッシング関数に基き，Detail の対応するエントリのアドレスを高速に見出すため都合がよく，また Detail 側で行ったキーアイテムの登録・抹消に対し Automatic Master 側は自動的にその管理を行う．ただし，この場合アイテムは1つに限られ，それはキーアイテムとなる．

一方，Manual Master はキーアイテム以外に複数のアイテムを持つことができ単独に存在できるデータセットである．またハッシング関数による高速見出しは Automatic Master と同様に可能である．しかし，キーアイテムの管理は Detail 側と独立にユーザが行わねばならない．これらを利用した K-3 の全スキーマ・ダイアグラムを第4図に示す．逆三角形が Master Data Set を意味し，A が Automatic Master，M が Manual Master である．また台形が Detail Data Set を意味する．ここでは Manual Master が2個，Automatic Master が13個，Detail として35個のデータセットを設けた．総計50個のデータセットは IMAGE/1000 の許容する最大のデータセット数に達している．第4図の内容を計算機上にのるスキーマ記述に書き換えたものは約900行のステートメントとなった．ここでは，電波星の赤経を例にとり項目（アイテム）の定義とデータセット中の定義がなされている部分を抜き出し第5図中に下線で示した．また，各データセットの生成単位について第3表に示した．

3.3 データアクセス方法[9]

実際の観測例をもとにして，データアクセスがどのように行われているかを以下に述べる．例として単なるデータ・ファイル・アクセスから考える．第6図にその例を示す．ここで星の名前が 0212+735 が必要であるとして，その位置に達するには先頭から逐一サーチしていく必要があるので方法としては MT の場合と同じくシーケンシャルになる．しかし第7図のように管理用ファイ



第4図　スキーマダイアグラム

```
<<  ***********************************************
<<  * KASHIMA  IMAGE/1000  DATABASE DEFINITION  PROCESSOR *
<<  ***********************************************
<<
$CONTROL:LIST,ERRORS=001,ROOT,NOSET,FIELD,TABLE;
<<$CONTROL:LIST,ERRORS=1,NOROOT,NOSET,FIELD,TABLE;>>
BEGIN DATA BASE:  #KLEVR:K3:30;
LEVELS: 1 LEVEL1;
        2 LEVEL2;
        3 LEVEL3;
        4 LEVEL4;
        5 LEVEL5;
        6 LEVEL6;
        7 LEVEL7;
        8 LEVEL8;
        9 LEVEL9;
       10 LEVELA;
       11 LEVELB;
       12 LEVELC;
       13 LEVELD;
       14 LEVELE;
       15 KASIMA;
<<
ITEMS:  ANTID    , X2(1,15);     << ANTENNA ID
        ANTTMP   , R2(1,15);     << ANTENNA TEMPERATURE
        ANTERR   , R2(1,15);     << ANTENNA ERROR

        PRIODC   , R2(1,15);     << PERIODIC
        QUALTY   , X6(1,15);     << TAPE QUALITY CODE
        RA       , 411(1,15);    << RIGHT-ASCENSION
        RATTUD   , R2(1,15);     << RATTUDE
        RDRAD    , 2X24(1,15);   << ASC. AND DEC. RATE

 **********************
 *   STAR INFORMATION   *
 **********************
 NAME:       #STAR::30,D;
 ENTRY:      STAR(#SLIST),
             NAME,
             RA,
             DEC,
             EPOCH,
             MRADC,
             PRALAX,
             MPRLAX,
             FLUX,
             MFLUX,
             PMOTN,
             MPMOTN,
             FLAG,
             INDEX,
             TYPE,
             MK3,
             PULSER,
             DURATN;
CAPACITY:    251;
```

第5図　スキーマ記述（抜粋）

ルを設けると，アドレスの管理能力が加わる．このケースを IMAGE/1000 にあてはめると管理側のファイルが Master Data Set，管理される側のファイルが Detail Data Set に相当する．こうすれば，0212+735 のアドレス“5”が即座にわかるため Detail 側でのエントリの発見が容易となる．こうしたアドレスの管理を一本だけでなく幾重にも張りめぐらしてユーザの希望を満足させるような機能を作りあげていくことができる．

次に第7図の管理用ファイルにも高速にアクセスするため，さらにユーザのデータ取扱いの便利さのため，キーの持つ値「キー値」を直接にアドレスに使いたい．つまり，「名前を呼べば，すぐにありかがわかる」という虫のいい話はないであろうか．この考え方がハッシングと呼ばれる方法で実現されている．実際，キー値には，文字型，実数型等さまざまな変数が使われるので，そのままアドレスの値としては使用できない．整数型であっても負の数であったり，用意された番地とかけ離れた数値であったりするので，何らかの変換が必要である．これをハッシング・アルゴリズムと呼ぶ．これはキー値を

第3表　データセットの生成単位

データセット	生成単位
実験情報	実験
Raw テープ情報	Raw テープ数／局数／実験
観測ファイル名情報	実験
保管MT情報	解析数／実験
観測属性情報	観測／実験
相関共通情報	観測／実験
校正情報	局数／観測／実験
相関器情報	KROSS ラン／基線/観測/実験
相関情報	KROSS ラン／基線/観測/実験
トラック情報	周波数コード数
サブグループ情報	周波数コード数
局ごとの星位置情報	局数／観測／実験
予測遅延	基線／観測／実験
寄与・偏微分係数	基線／観測／実験
バンド幅合成情報	KOMB ラン／基線/観測/実験
物理モデル情報	実験
推定パラメータ	書込指定／実験
推定用観測データ	書込指定／実験
推定用条件	書込指定／実験
推定基線長	書込指定／基線／実験
基線長フォーマルエラー	書込指定／基線／実験
推定エッポク管理	書込指定／局数／実験
excess path 管理	書込指定／基線／観測／実験
星情報	星
衛星・惑星情報	種別
IPMS・BIH 情報	日数／種別
地球回転情報	観測／実験
惑星暦管理情報	登録数
惑星暦係数	種別／登録数
惑星位置情報	観測／実験
天文・物理定数	1個
章動テーブル	2個（Wahr または Woolard）
地球潮汐テーブル	1個
海洋潮汐テーブル	局数
観測局情報	局数
アンテナ情報	局数
クロック情報	局数

Master 側の記録領域に均等にふり分ける方法である．その結果，見かけ上，キー値を直接アドレス指定に用いたことになり，扱い易く，高速のデータアクセスを保証

レコードNo	実験コード	EXP+OBS	参加局コード	局 数	星 名
1	$ 84FEB25SX	$ 84FEB25SX0001	′>H	3	4C39.25
2	$ 84FEB25SX	$ 84FEB25SX0002	′>	2	0552+398
3	$ 84FEB25SX	$ 84FEB25SX0003	′>H	3	OJ287
4	$ 84FEB25SX	$ 84FEB25SX0004	′>H	3	4C39.25
⑤	$ 84FEB25SX	$ 84FEB25SX0005	′>	2	0212+735
6	$ 84FEB25SX	$ 84FEB25SX0006	′>H	3	4C39.25

第6図 シーケンシャル・データファイル・アクセスによるデータ読出し

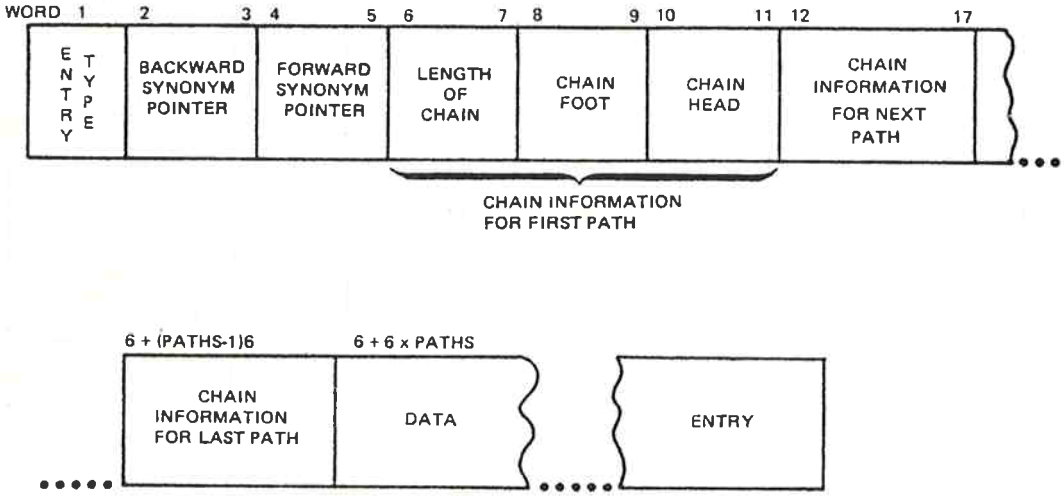
先頭レコード	実験コード
123	$ 82JAN23SX
1	$ 84FEB25SX

先頭レコード	星 名
2	0552+398
5	0212+735
1	4C39.25

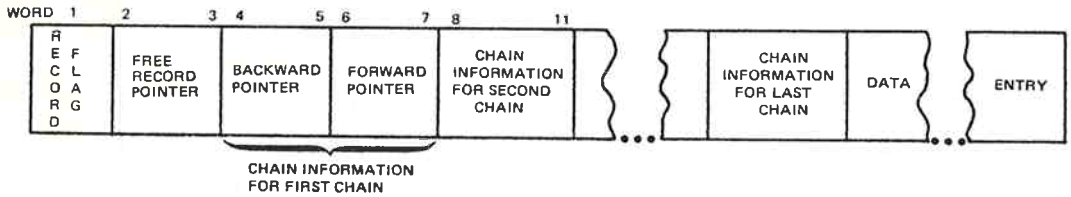
レコードNo	実験コード	EXP+OBS	参加局コード	局 数	星 名
1	$ 84FEB25SX	$ 84FEB25SX0001	′>H	3	4C39.25
2	$ 84FEB25SX	$ 84FEB25SX0002	′>	2	0552+398
3	$ 84FEB25SX	$ 84FEB25SX0003	′>H	3	OJ287
4	$ 84FEB25SX	$ 84FEB25SX0004	′>H	3	4C39.25
⑤	$ 84FEB25SX	$ 84FEB25SX0005	′>	2	0212+735
6	$ 84FEB25SX	$ 84FEB25SX0006	′>H	3	4C39.25

第7図 管理用ファイルを用いたデータ読出し

する．ハッシングにも難点があり，まずいアルゴリズムでは度々同じアドレスを発生してしまい，シノニム（同義語；ぶつかり合い）を生ずる．シノニムが発生した場合，一方のデータは他の格納場所を捜さねばならずアクセス速度は低下する．このためデータができるだけランダムに記憶領域にふり分けられるように，さまざまなアルゴリズムがある．IMAGE/1000の場合，データセットの容量を素数で指定し，キー値をこの値で割った余



(a) Master Data Set レコード



(b) Detail Data Set レコード

第 8 図

り，即ち

IRN＝MOD(KV, IPN)

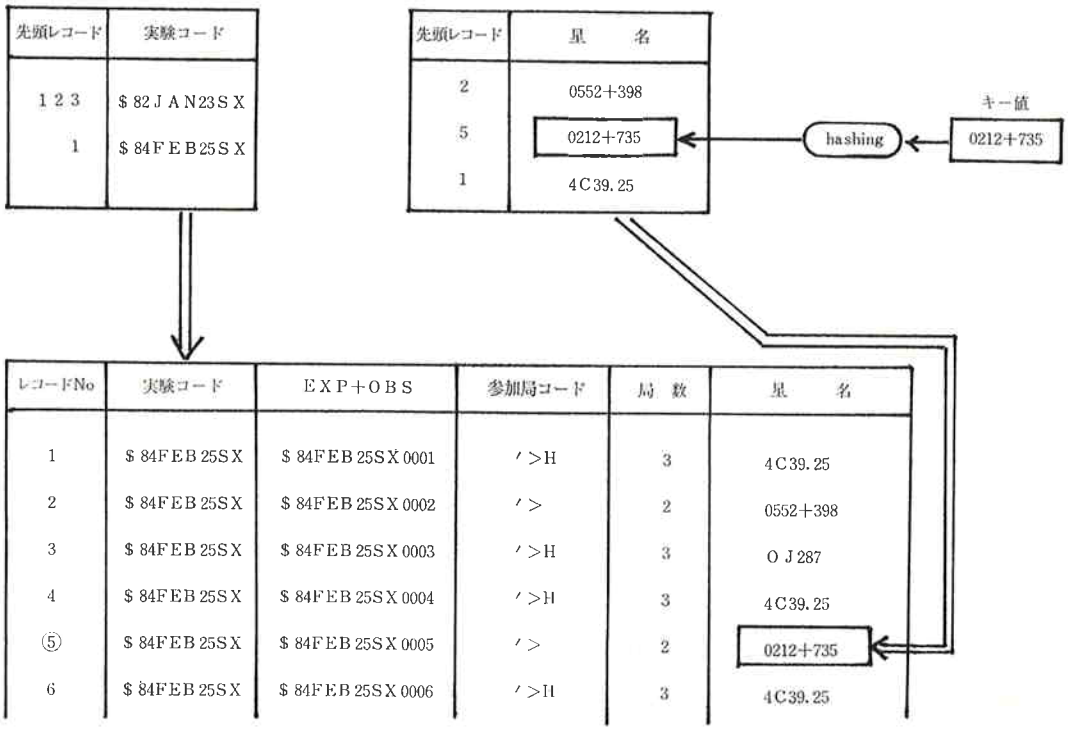
IRN：番地，KV：キー値，IPN：素数

としているようである（具体的には公開されていない）．こうして単なるデータファイルアクセスに比べ，「機械的アドレス指定でなく，アイテムのもつキー値を指定すれば必要なデータを高速で見つけ出す」方法が便利なことがわかる．IMAGE/1000 はこの機能を持っているが，キーアイテムに対してのみ有効であるので，読み出しには工夫が必要である．ここで第8図に Master とDetail の Data Set のレコードを示す．図から理解されるように，データの有無，シノニムの扱い，キーを連結するチェーン情報，データ等が各エントリの中で有機的に結合されている．

次にデータ読出しのいくつかの方法について述べよう．先にあげたのはアイテムの中でもキーアイテムと呼ばれるもので，Master 側と Detal 側のパスを形成する役目を果たしている．そのため，キーアイテムに関してはアドレス管理がしっかりとなされている．しかし，他のアイテムにはそれがない．すべてのアイテムをキーにすることは無駄であり，かつ IMAGE/1000 の制約からも不可能である．従って設計上，キーをどのアイテムにおくかは重要なポイントとなる．IMAGE/1000 には4つのデータ読出し方法がある（第4表）．キーでないアイテムはこの内 serial read または directed read を使えるが通常前者が用いられる．この場合，アクセス速度は chained read より遅くなる．従ってキーを利用して必要データの入っている可能性のあるエントリの集合を絞り込んで，chained read によるサーチを行うのが最も速いと考えられる．ただし複数のキーアイテムをもつ Detail では，キーの指定によって最も小さく集合が

第4表 IMAGE/1000 データ読出し方法

Chained Raed	Detail Data Set 中で，与えられたキー値の全エントリを順に読む
Serial Read	Set 名だけ指定し，空でないレコードを順に読む
Directed Read	相対レコード数を指定して，そのレコードを読む
Keyed Read	キー値を Master Data Set に与え情報を得る



第9図　ハッシングと Master Data Set を用いた読出し

限定されることが望ましい．このため頻繁にアクセスされる「実験コード」(VLBI データの処理・解析は DB 中のある実験データの集合に対して行われるので，実験後 DB をセットアップする際この集合に「実験コード」名が付与される）と「観測数」(VLBI 実験ではいくつものソースを順次切替えては受信データを記録する．各々は「観測」と呼ばれ，VLBI は多くの「観測」からなると言える）に関して次のようにした．前者はある実験に連なる全情報を引出す際に便利である．しかし，集合の要素は数百に及ぶ．当然，他の実験との名称の重複はありえない（例，($ 84 JAN 23 SX)．一方，観測数は特定の観測だけに限定されるので，その数は S/X の区別，処理回数等で増すものの，数個程度であり都合がよい．しかし，他の実験と重複がおこりやすい（例，0314)．そこで両者ともキーアイテムにし，特に後者を実験コードと複合した名称にして管理している（例，$ 84 JAN 23 SX 0314)．

こうして始めにあげた星名 0212+735 を捜すにはハッシングにより Master 側での chain 先頭アドレス発見と Detail 側での chained read を行えば良い（第9図）ことがわかる．

3.4 KASTL

既に述べたように KASTL は，VLBI 用 DB スキーマが完成しデータも格納された IMAGE/1000 DB に対し特別の知識がなくてもデータを利用できるようにするため作られた．KASTL には以下にあげる3つの機能がある．

(1) DB の内容表示（処理状況，保管データ等）

(2) DB の指定範囲削除（実験単位，処理単位等）

(3) アクセス・キーによる DB 作業の請負

(1)(2)については IMAGE/1000 で提供されている会話型照介言語 QUERY である程度カバーできる部分だが定形化した利用については KASTL を利用する方がはるかに楽で，間違いがない．(3)について「アクセスキー設計書」を作成しその定義を明らかにした．K-3ソフトがすべて DB を利用していることから，ソフト開発に際してすべての DB アクセスをアクセスキー方式で行った．これにより，プログラマーは IMAGE/1000 を熟知する義務から解放された．また逆にアクセスキーの作業と各ソフトからの利用のされ方を見れば K-3 におけるデータフローの全貌を見ることができる．キー番号は読込，書込またデータセット等で区分している．各々1つ

```
:KASTL
? SECURITY
Security code ?
? MO=PU
PURGE? SU=BA
PURGE? ID=$83DEC21KV
 Data Base is opened here.< DBOPN >
Data-Base name is -----> #KLEVR:K3:30
 IMAGE/1000 DBGET is called, now.
Data-Set name is----> #EXPER
 #####Now, Data Base was closed. <DBCLS>
PURGE? BA=EF
 Data Base is opened here.< DBOPN >
Data-Base name is -----> #KLEVR:K3:30
 IMAGE/1000 DBFND is called here.
Data-Set name is ----> #STOBS
 IMAGE/1000 DBGET is called, now.
Data-Set name is----> #STOBS
 #####Now, Data Base was closed. <DBCLS>
 Data Base is opened here.< DBOPN >
Data-Base name is -----> #KLEVR:K3:30
 IMAGE/1000 DBFND is called here.
Data-Set name is ----> #STOBS
 IMAGE/1000 DBGET is called, now.
Data-Set name is----> #STOBS
 #####Now, Data Base was closed. <DBCLS>
Baseline  WESTFORD - HRAS_085  (EF)
PURGE? OB=70
PURGE? PURGE
 Data Base is opened here.< DBOPN >
Data-Base name is -----> #KLEVR:K3:30
 IMAGE/1000 DBGET is called, now.
Data-Set name is----> #EXPER
 IMAGE/1000 DBUPD is called here.
Data-Set name is -----> #EXPER
 #####Now, Data Base was closed. <DBCLS>
 Selective condition

  Exp#     = $83DEC21KV
  Obs#     =   70
  Data     = KOMB
  Baseline = WESTFORD - HRAS_085
```

第10図 KASTL のアクセスキーを経由した IMAGE/1000 ライブラリ・コールの例

のキー番号で複数のアイテムの読込または書込を行う．そのためユーザは DB のオープンに始まり複数個のアイテムのアクセスそして DB のクローズに到る作業をアクセスキー指定による KASTL の起動とプログラム中のインタフェースのみに留意すれば良いことになった．第10図にアクセスキーを利用して作られている KASTL の動きを IMAGE/1000 ライブラリのコール部分だけをインバースで表示させたものを示す．DBXXX なるルーチンは応用プログラムが直接呼んでいるのではなく，定義された番号に従い KASTL で呼ばれたものである．このように K-3 DB に対する典型的なデータアクセスに関しては既にプログラミングされているアクセスキーを利用すればよく，IMAGE/1000 の知識を全く必要としない．また，新たなデータ集合の利用要求に対して，ユーザがアクセスキーを定義できるようになっている．

アクセスキー方式の KASTL のもう一つの長所は，DB と応用プログラムの独立性の強化である．第2図に見たように，DB 側の何らかの改造に対し，もし，応用プログラムが直接 IMAGE/1000 を利用しているとすると応用プログラムの関連する箇所すべての改造が必要となる．しかし，ここで行われているように KASTL を用いると DB の改造のために必要となる修正はインタフェース・ルーチンとしての KASTL の部分だけにおさえこむことができる．またその作業量も前者に比べ格段に減少すると予想される．

更に具体的で詳細な点は本特集号 IV-4 を参照されたい．

4. DB の 維 持

4.1 DB の容量

DB は本質的に random access 及び mass storage を要求するため，ディスクの領域を大きく占有する．所要ディスク容量は，1 実験，100 観測，5 局の例で 10 MB を必要とする．現在，我々は 67MB のディスク領域を確保しているので，6 実験を常時保管できる．一方，将来の実験環境をサポートするための所要ディスク容量は第5表に従うと 670 MB となる．これは計算機ハードの現実的な数値ではあるが，先の10倍の領域をとるので，スキーマの見直しによる占有領域の削減が必要となろう．

第5表 将来の実験頻度と規模の仮定

内 容	頻 度	局 数	観測数
POLARIS 相当	1週毎	3	200
国 内 実 験	1カ月毎	5	200
国 際 実 験	3カ月毎	5	200

なお，ディスクの増加による DB 領域の拡大は，スキーマの一部修正及びデータの再ロードで対応できる．しかし，アイテムの修正は過去のデータとのコンパチビリティに問題があるので，容易には行えない．

4.2 データ交換

日米実験の処理結果は Mark Ⅲ型 DB テープで交換することが決まっている．このため K-3 と Mark Ⅲ の DB の変換ユーティリティを作成した．この時，解析に必要十分なアイテムの選択が行われ，双方の対照表が完成した．これに基き，サンプル DB のテープ輸送を行い，双方のデータの確認がなされた．一部のデータを除き，K-3 の DB は Mark-Ⅲ の DB を包含していると言えよう．

4.3 データ保管

IMAGE/1000には，データのバックアップ・ユーティリティが2種類用意されている．一方はDBULD/DBLODであり他方はDBSTR/DBRSTである．一般に後者の方が所要時間が短く扱いが楽であるが，データセットのサイズを変更してDBを再ロードする際は前者によるしかない．DB保管は両者を使い分けている．

データ保守の際，何らかの原因でDBの構造が破壊されていないか，また，DBのディスク領域の利用度をモニタするDBSPAユーティリティを利用する．

5. まとめ

明確なスキーマを持ち，VLBI全ソフトから利用され，会話型ユーティリティをはじめさまざまな機能をもった本格的DBをVLBIソフトに導入したのは世界でも初めてである．使用を続けていく内に，データのアクセス速度，固定的スキーマ，倍精度変数の欠除等に伴う問題点が見えてきたものの，DBは丁度空気のように，なくなって初めてその便利さがわかるという性格を持っており，活発な研究活動に一定の秩序を与えデータが原因の無用の混乱を避けるための重要な機能を果たし始めているものと信ずる．

さて，K-3で完成したDBは変換ユーティリティでMark-Ⅲタイプにおとし交換できる．またその逆も可能である．米国ではアイテムの性質によるタイプ分けを従来の2種類から3種類にふやしている．日本側では大きなK-3 DBから交換に必要なものだけを取り出している．この作業は日米双方にとって，必要にして十分なVLBIアイテムを選択する格好の機会になっている．即ち，VLBIデータ項目が淘汰されているのである．これにより真に必要なVLBI DBの形がアイテムだけのレベルを超え見えてきている．将来にこの点を生かしたい．

K-3 DBの場合，IMAGE/1000の性質からスキーマは固定的であり，他機種とのデータ交換も困難である．しかしパソコンから大型計算機までDBが普及し始め，改良も急ピッチとなることが予想されるので，VLBIの数百実験データを管理する頃には今回のスキーマをもとに更に身軽で充実したDBでデータ管理がなされるであろう．

最後に，K-3 DB設計・製作に協力頂いた日本電子開発㈱の鈴木保立氏に感謝致します．またNASA/GSFCを始め国内外の多くのVLBI関連機関の方からの御教示を頂いた．ここに感謝の意を表します．

用語

DB＝データベース

アイテム＝データ項目．データの最小単位

エントリ＝アイテムの組合せでデータセットの最小単位となる．

データセット＝エントリの集合で，データの性質毎にまとまっている．

IMAGE/1000＝HP社のDB管理システム

スキーマ＝DB構造を定義する時の設計記述

ハッシング＝ちらし（hashing）関数によるデータのアドレスの乱数化

アクセスキー＝K-3 DBアイテムの集合をアクセスする際の定義された番号

参考文献

(1) Chopo Ma, "Very Long Baseline Interferometry Applied to Polar Motion, Relativity and Geodesy", NASA Technical Memorandum 79582, May 1978.

(2) 吉野，近藤，高橋，小池，国森，真鍋，鈴木；"K-3型VLBIデータ処理・解析ソフトウェアにおけるデータベースの位置づけとその機能"，昭和57年度経緯度研究会集録

(3) パネルディスカッション，"我が国のVLBI網におけるデータベース統一に関する問題点"，「測地・地球物理学における宇宙新技術利用」に関するシンポジウム集録，1982年

(4) Hewlett Packard, "IMAGE/1000 92069 A and 92073 A Data Base Management Systems ―― Reference Manual", 1980

(5) "製品調査 データベース・マネジメント・システム/米国のDBMS市場が本格的な成長期に"，日経コンピュータ，1981年 Nov. 2 及び Nov. 16

(6) 穂鷹；"実践講座 データベースⅧ(設計ならびに開発の諸問題その1)"，bit, **Vol. 9**, No. 10

(7) V. ベンカタクリシュナン，"データベース設計を支援するサブジェクト・データ・モデリング"，日経コンピュータ，1983年 July 11

(8) 川喜田；"発想法"，中公新書

(9) J. Martin；"データベース管理"，コンピュータ・サイエンス研究書シリーズ33，日本コンピュータ協会，1982