解　説

医療におけるデータベース*

開原成允** 上野晴樹*** 若井一朗****

1. 医療の中のデータベース

一般に「医療」は，一つの領域をなすと考えられがちであるが，医療は，その中に著しく多くの分野を含んでいる．

従って，その中に現れるデータベースの問題も決して一つではなく，対象とするデータベースによって全く異なった問題点を有している．ここでは，こうした医療に現れるデータベースを仮に「複数の利用者が利用する全てのデータを冗長性を少なく共同利用可能な形に統合したものであって，その目的が医療用であるもの」と定義し，その諸問題を三つに分けてのべることにする．すなわち，1. 病院業務の基礎となるデータベース，2. 診療記録のデータベース及び 3. 医療情報サービスのためのデータベース，の3つである．

第一の病院業務のデータベースは，病院が日常の診療を行う上で必要とする事務会計・検査・投薬・入退院等の患者の情報を中心としたデータベースである．

第二の診療記録のデータベースとは，主として，臨床研究のために，病歴の内容を貯蔵したデータベースである．

第三の情報サービス用のデータベースとは，医師が診療を行う上で必要な各種の情報を提供するデータベースである．例えば，薬剤情報・文献情報のデータベースである．この三つのデータベースは，それぞれ目的が全く異なっている上に，データ量も処理形態も異なっているため，システムとしては，異なった配慮をする必要がある．以下にそれらの各々について，その問題点を考えつつ解説を加えたい．

* Data Base in Health Care by Shigekoto KAIHARA (Hospital Computer Center, University of Tokyo Hospital), Haruki UENO (College of Science nnd Engineering, Aoyamagakuin University) and Ichiro WAKAI (Chukyo Hospital).
** 東京大学医学部
*** 青山学院大学理工学部経営工学科
**** 社会保険中京病院

2. 病院業務の基礎となるデータベース[1)]

病院では，日夜診療が行われ，そのためのデータは，現在，主として伝票の形で行き交っている．ここでは，患者に関する情報を事務・検査・診療等で相互に利用しているわけであるから，これらをデータベースの観点から整理し，一つのシステムにまとめることは，当然考えられる所である．

このとき，病院の各種のデータは，どのような特徴をもち，これに適したデータベース管理システムとは，どのようなものかが，ここで考慮しなければならない問題である．病院のデータベースの中の患者データの部分は，**図-1** にみられるように，患者の ID 情報を中心とした Tree 状の構造をもったデータと考えることができる．すなわち，それぞれの患者が診療をうけるたびに，その日付の下にその診療内容が付加されていく構造である．

また，処理の形態としては，ほとんどのものが，real time 処理でなければならないこと，処理の内容としては，複数の key からの検索，作表が主たる処理であると考えられることができよう．この意味から，処理自体は，それ程，複雑なものが現われるわけではないが，病院データの特徴は，少量多種であることであり，例えば，検査まで含んだシステムを考えるとすると，検査項目は，200 項目以上に及んでいる．

また，病院のデータは医療という性格上，データの保全と秘密保持には厳重な注意が必要である．特に保

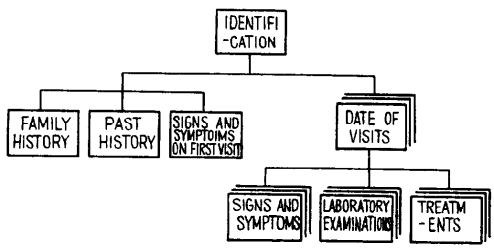


図-1 医療データの構造

全の問題は重要である.

しかし，この保全と秘密保持を厳重に行うと，データベース本来の責務である1つのデータを複数の使用者が real time で使用するという利点が失われかねない．これは，データベースの内容が著しく速く入れ替っていく病院業務にあっては，特に深刻な問題をなげかけることになる．このように，病院のデータ・システムにおいては，患者の入れ替わりと共にデータベースのメインテナンスを如何に real time でかつ誤りなく行うかが最も重要な機能ということができるであろう.

もう一つ，病院のシステムの中では，単にデータを一つの場所から別の場所へ伝送するだけのことが，重要な機能の1つであるという点にも注意を要する．これは，データベース本来の機能からすれば，あまり重視されないものであるが，病院では，こうした伝送だけで 50% 以上の目的が達せられる場合も多い．従って，計算機能や，データベース管理機能よりも，計算機の伝達機能を主に使っていることにもなるわけで，この点も汎用のデータベース管理システムを用いる場合の1つの問題点である.

こうした病院における様々なデータベースを扱うシステムを作っていくには，これまで次の2つのアプローチがあった.

第1は，すでに開発されている汎用のデータベース管理システムを用いてシステムを作っていこうとするアプローチであり，第2は，データベースとしての汎用性を考えずに，医療用として使うことを主たる目的として専用のデータベース管理システムや言語を開発しようとするアプローチである.

第1のアプローチをとると，新しくシステムの開発をする必要はないが，汎用のものを用いるため，システムは一般に大きなものとなる．それにも拘らず，現在の汎用データベース管理システムは，医療の End User 向きにできていないため，その周辺に多くのサポート・システムを付加する必要が生じ，システムとしては，いよいよ大きなものとなる傾向にある．従って，ある程度の規模をもった医療機関においては，かかるシステムを維持することは可能であるが，小規模の病院等では，到底使用できない.

このような汎用データベース管理システムを基礎とした例としては，ウィーン大学病院の WAMIS (The vienna general medical information system), 英国の united cambridge 病院を中心とした4病院の協同利用のシステム，米国 Rockland State Hospital を中心とした6州に亘る MSIS (Multi-state information system)，スウェーデンのウプサラの臨床検査データを中心としたシステム等がある.

これらは，いずれも Codasyl 型のデータベース管理システムかまたは IMS を基礎にして開発されているものである.

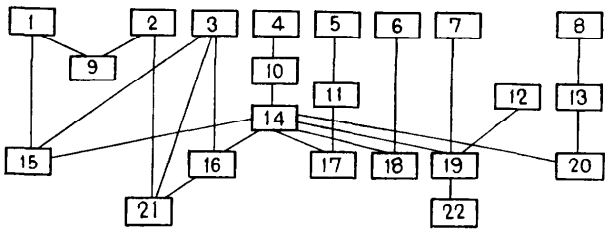
データ構造の1例を図-2 に示す.

第2の医療用に開発されたデータベース管理システムは，医療にあらわれる処理を充分考慮しつつ開発されたシステムである．汎用性はある程度犠牲にしても，医療上よくあらわれる処理をできるだけ簡単に行おうとしている．また，規模も，小病院でも用いられるように，いわゆるミニコンピュータに implement されるものが多い．米国 Duke 大学の GEMISH (GEneralized Medical Information System for Community Health)，ボストンで開発された MUMPS (Massachussetts General Hospital Utility Multi-Programing System) 等がある．MUMPS は一つの言語であり，厳密にはデータベース管理システムとはいえないが，ファイルの構造に特徴があり，米国を中心として，広く普及しつつある．また，最近では医療以外の分野でも使われるようになっている．MUMPS については，医療関係者以外では比較的知られていないため，以下にややくわしくのべることにする.

3. MUMPS におけるデータベースの構造と設計

3.1 MUMPS のデータベースの構造（論理的）

MUMPS のデータベースはグローバル (Global) と



1. 病　棟
2. 診療日
3. コンサルタント
4. 総索引
5. 診療所
6. 薬　局
7. 検査室
8. 手術室
9. ベッド利用状況
10. 登　録
11. 医　師
12. 検査項目
13. 外科医
14. 診療録
15. 入院患者
16. 待患者
17. 予　約
18. 薬　剤
19. 検　査
20. 手　術
21. 入院待患者
22. 検査結果

図-2 汎用データベースの構造

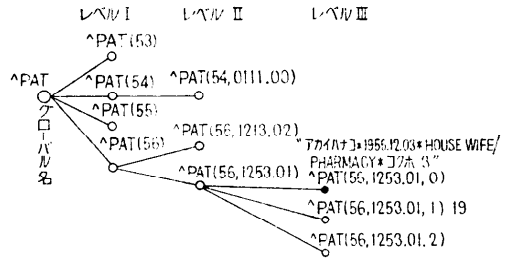


図-3 グローバルの論理的構造

呼ばれ，ディスク中に Tree 構造の節 (Node) を形成する「グローバル変数」の集合である．グローバル変数は，エントリーの頭であるグローバル名をのぞいて，すべて添字つき変数で表現され，変数名に“へ”(上向き矢印)をつけて表現される．グローバル変数の中味は，全く空であっても，数値変数または文字列を入れてもよい．

添字つき変数は，“宣言なし”のスパースアレーであり，第1添字の等しいものはすべて，レベルⅠの同一添字の Node のもとに包括される．第2添字以上も同じである．例えば**図-3** の如くユーザ領域のコアの中で扱う「ローカル変数」に PAT (56, 1253.01, 0) があり，これに文字列データとして“アカイ ハナコ＊1956.12.03＊HOUSE WIFE/PHARMACY＊コクホ 3”を入れたとすると，^PAT(56, 1253.01, 0) というプログラムコードはディスクのグローバル・ファイルのレベルⅢの●印 Node を作り，そこにこの文字列データを入れる．

添字に意味をもたせることは，Tree 構造上の派生的データが論理的に一定の連結をする場合に便利である．例えば，^PAT(56) の Node がその下に 1956 年生れの人をすべて包括するようにすることもできるし，また ^PAT(56, 1253.01) が 1956 年でれで 12 月 3 日生れの女性で同日生れの女性のうち，登録を第一番におこなったもの，という意味をもってもよい．

3.2 グローバル・ファイルの設計

グローバル変数の添字はコンマで区切って無制限に，すなわちレベルの数を無制限に深くすることができるが，しかし深い Node は検索に時間をとるので，せいぜいレベルⅤまでが一般的である．また同一レベルで同一の親を Node にもつ兄弟Nodeの数は，添字のとりうる数値の数だけ作りうることになり，MUMPS-11 Version 2 では約 209 万個である（さらに新しい MUMPS 言語では添字の数値限度は拡大されている）．兄弟 Node に入るグローバル変数のデータ間に

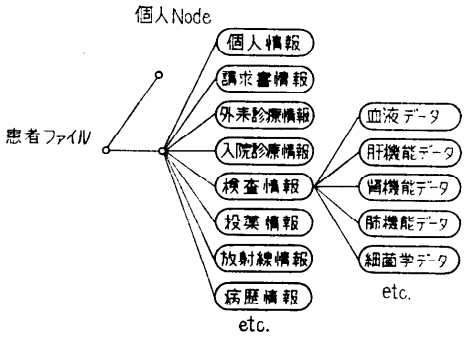


図-4 患者ファイルの設計様式

は論理的従属関係は必要でない．このことは1個の Node の下に多くのデータを並列に並べられるということを意味しており，**図-4** の患者ファイルの模式構造のように，個人を示す Node の下に互いに独立した情報群をおくことができる．

各情報群の多数の Node は特定の情報を表わすコード的変数を添字にもってもよく，また，情報コードに日付を付加したものを添字としてもよい (data containing subscript)．検査データの場合時間を含めることもできる．

3.3 グローバル変数の設計

グローバル変数のデータ型としては，可変長文字列型である場合と，数値型である場合の二つがある．例えば図-3 の ^PAT (56, 1253.01, 0) は文字列型のデータ型をとっている．文字列の文字の制限は 255 字であり，その中に多くのデータ要素を境界子(delimiter)で結合させることができる．境界子にはいかなる文字を用いてもよいが，通常“^”,“;”.“＊”,“／”,“＼”などが好んで用いられる．境界子と境界子の間に別の境界子を subdelimiter として用いることも，sub-subdelimiter として第3の境界子を用いることもできる．

これらはオプション・データをはさみ込むときに好んで用いる．数値型データは例えば年令だけを単独に入れておく場合，^PAT(56, 1253.01, 1) 即ちレベルⅢの添字(1)は年令値のグローバル変数として，19 を入れるのみである．ここに年令と性を F 19 というデータとした場合には，文字列型データとなり変数値は“F 19”として入れられることになる．

3.4 グローバルの物理的構造

ディスクの基本的な物理的記録単位はブロックである．1つのブロックの大きさは 762 文字で一部分が他のブロックとの関係およびブロック内の使用領域の記

録に使用される.

同一レベルにある Node 群のデータは常に1個のブロックまたは，チェーンで連なった複数個のブロックに記録される．ブロック間のチェーンをおこなうポインタは，グローバルの親子関係 Node 間のポインタとは異なり同位配列子のデータ・ブロックを連鎖するのであるが下位レベルの Node へのポインタは Node 添字値，データ値と共に一つのブロック内にある．新しい Node の情報が添字値，データ値，ポインタ値ともに，1つのブロックの未使用領域に収まる場合は，その場所に記録されるが，完全に収まらない場合は，これをやめて新しいブロックへポインタを出して移る．

以上でみたように，論理的なグローバル・ファイルの設計は容易であったが，その物理的なディスク上の記録様式とは異なるために，ブロックアクセス数が少なくなるようなファイル設計が必要となる．プログラマが最も困難を感ずるのがこの点であり，ファイル設計が不適であると，応答時間が適正な場合に比して10倍～100倍も遅くなる．それ故にプログラムの開発に先だって，グローバル・ファイルの設計が必要となる．まず，チェーンされている N 個のブロックがあり，それぞれのブロックに目的とする Node が記録されている確率を同じとすると，ある Node を探し出すまでにアクセスするブロックの数の平均は $N/2+0.5$ となる．

即ち，チェーンされたブロック数が4個ある場合，アクセスされる数の平均は2.5である．応答時間は，ディスク・アクセス回数と密接な関係があり，ファイルサーチはレベルⅠに節の数を少なくし，レベルⅡ，レベルⅢと下るにつれて節を多くする富士山型の構造のものがブロック・アクセス，したがってディスク・アクセス回数を少なくすることになる．

3.5 プログラムとユーザ領域での変数の扱われ方

ユーザ領域での変数はすべてローカル変数であり，変数名には“^”がついていない．^PAT(56, 1253.01, 0) は，ユーザ領域では PAT(56, 1253.01, 0)＝“イトウ　ハナコ*…”として文字列型変数値をもつスペースアレイとなっているはずである．プログラムによって種々のデータ要素を結合し，文字列を形成して，それを1変数値とし，変数名に“^”を付したときに，ディスク中のグローバルに Node が生ずると表現した方が良い．

MUMPS 言語には，文字列操作のための演算子[[(含む)，]](文字セット上，順位があとにくる)，－(結合する)，?(パターン照合)]があり，データの性格，特徴を診断して入力・出力時のエラーをふせぐことを高度に可能にし，データを多数結合して，一つの文字列としたりする．FUNCTION 文字列をいくつかに分解する $EXTRACT，データ型を識別する $DATA，数字を文字に変換する $CHAR，文字列の中の特定の文字を数字に変換する $ASCII，文字列の長さをみる $LENGTH，特定境界子間の文字列を抜きとる $PIECE などを持つ．

3.6 MUMPS プログラム言語の特徴

プログラム言語としての特徴は，interpreter であって，コードの左から右に向けて1字1字実行に移す．これはコンパイラ言語に比して実行速度を遅くするといわれるが，バッチ処理をとらない会話型ではそれほど苦にならない．MUMPS の特徴は会話型であり，ソースプログラムを対話式にデバッグしながら書いてゆくことができ，この点高度にユーザ指向型の言語であるといえる．データベース指向型であるため，組込み関数が少なかったが，標準 MUMPS では無指定の組込み関数が追加できるようになっている．1975年ごろから，標準 MUMPS を中心に他の言語(FORTRAN など)とのリンク，または，コンパイラ化，図形表示・解析への努力がなされつつある．

MUMPS 言語は以上みた如く，汎用のデータベース管理システム (DBMS) としては必ずしも完全ではない．スキーマの考え方もなく，また，データ独立でもない．データの integrity に関しても不完全という他はない．しかしそれにも拘わらず汎用のデータベース管理システムにないいくつかの機能をもち，しかもミニコン・レベルで一つのデータベース管理言語を提供した点で，大きな意味をもっている．

MUMPS プログラミング言語の欠点としては，user-dependent な変数名，迂廻話法などによって，プログラムの読み易さが犠牲になるという点がある．

また，データ・ベースの設計が終ると，すでにユーザはデータの入力・出力プログラムにうつり，仕様書の作成さえおろそかにするので，一人よがりなプログラムとなり，秘語化するおそれがある．ユーザがたとえプログラムする場合にも，プログラムの普遍性を念頭におかないと，開発時間の短かさだけを追求することになり，他人が読めず，従って移送しにくく，多くの人々によって次第に完成されるという可能性を狭くすることになる．こうした MUMPS の利点・欠点は A. Wasserman の A Balanced View of MUMPS[2]

に詳しいので参照されたい.

4. 診療記録のデータベース

診療記録の内容をコンピュータに入れて処理したいという要望は，かなり古くから議論された問題である．しかし，この問題は，その目的があまりにも多様であるために決して容易な問題ではない．診療記録を形作るデータは，確かにデータベースの観点から扱えるものである．しかし，この診療内容をいかに処理するかについては一般には定め難い．その理由は，この処理が主として，臨床研究・疫学研究にあるため，何を key にして処理するかが，事前には定め難いからである．この意味では，データベースの最適設計ということは，診療記録のデータベースに関しては考え難い．この意味から，診療内容のデータベースを現在あるデータベース管理システム的な観点から眺めてもあまり意味がないようにも思われる．強いて考えるならば，現在ある relational model のように，end user の立場からデータベースを如何に approach するかの手順を考えていくことに意味があるように思われるがこれは技術的にも容易なことではない[3]．現在あるデータベース管理システムの中では ADABAS が flexibility にとみ，この目的には最も使いよさそうであるが実際に適応された例はない．

こうしたデータを扱うときの一つの救いは，処理の時間を考慮する必要があまりない点で，ある程度時間がかかっても望むデータが容易に得られるシステムの方が望まれているのである．

5. 医療情報サービス用データベース

これは通常の Information Retrieval のデータベースで，医学上でこれまでも既に使われているのは，医学文献情報及び，薬剤中毒情報のデータベースである．特に，医学文献情報に関しては，MEDLARS とよばれるシステムが稼動している[4]．こうした情報は，データはほぼ固定している上に，処理も検索に限られるから，検索の能率のみを考えて，inverted file を主とした最も効率のよいシステムを組みあげることが必要である．MEDLARS のシステムは，検索のみを考えた特殊な専用システムである．

最近では更に，こうしたシステムをコンピュータ・ネットワークにのせてオンラインでサービスを提供することも考えられるようになっている．MEDLARS も MEDLINE とよばれるオンライン・システムが開発され米国では実用化されている．しかし，かかるシステムで最も重要なのはデータ自身であり，いかにして正確なデータベースを作っていくかが最も大きな問題となっている[4]．

6. 結 語

以上，医療に現われるデータベースの問題を3つに分類して，その概略と問題点についてのべた．こうした問題は，まだ今後研究しなければならない多くの点を有しており，特にわが国では実用化への一層の努力が必要である．医療の中には，複雑なデータベースが実例として多く存在し，これを研究することは，データベースの技術的問題としても実に興味あるものである．米国では技術的に最も高度なデータベースは医療データベースであるといわれているが，わが国では，残念ながら，本格的なデータベースは医療の世界にはまだ少ない．医療に携るものとしては，わが国のデータベースの専門家が，医療のような実用的分野にも眼を向け，本格的な医療データベースの開発に参加されることを切望する次第である．

参 考 文 献

1) J. Anderson and J.M. Forsythe: Methodology of Health Data Base Development MEDINFO 74 pp. 309～448, North-Holland (Amsterdam) 1975.
2) A. I. Wasserman: A Balanced View of MUMPS 医療情報処理研究会資料 10 (1976年4月).
3) A. I. Wasserman: Clinical Information and Relational Data Base Organization, Technical Report 17 Medical Information Science, University of California San Francisco 1975.
4) 開原成允: 医療における情報処理，情報処理，Vol. 16, No. 11, pp. 1001～1010, 1975.

(昭和51年7月7日受付)
(昭和51年7月28日再受付)