

## 4 医療情報処理用語について

上野晴樹 (青学大・理工)

### 1. はじめに

現在わが国では医療のシステム化が進行しつつあり、これにともなって各種の情報処理技術が広くとり入れられつつある。これらのシステムにおいてコンピュータに期待される役割りは極めて大きい。しかしながらこれらの開発に必要なソフトウェアの開発体制は非常に遅れている。この原因としては色々理由が考えられるが、主なものとしては次のようなことがあげられよう。

- 1) 医療は本来患者と医師との1対1の診断・治療行為であり、しかも各個人に最高の医療を行うことを前提とするから、コンピュータの導入が医療水準の向上に結びつきにくい(医療の本質に関する問題)。
- 2) 医療機関にコンピュータを導入し、各種の合理化・効率化を行っても、収支の向上にあまり貢献しない(cost/benefitの問題)。
- 3) わが国の平均的な医療機関の経営規模は非常に弱体であり、汎用中大型コンピュータを導入することは資金的に困難である。一方コンピュータの供給側は中大型機を中心にソフトウェアの開発を行っており、ミニコン用汎用ソフトの開発は置き去りにされている(産業構造の問題)。
- 4) 医療は非常に特殊な分野であり、医師側と情報処理技術者側の双方にお互いを理解できない面が多い。したがって各種の標準化が極めて困難であり、これがソフトウェアの開発の大きな障壁となっている(医療における特殊事情)。
- 5) 一般に医療情報処理においては、入力用端末機器とコンピュータのシステム構成に関心がもたれ、ソフトウェアに関しては(自動診断の論理を除いて)外部仕様のみが論じられており、極めて汎用性・柔軟性にとほしプログラムが個別に開発されている(ソフトウェアに対する認識不足)。

しかし最近になって、ソフトウェアの重要性に対する関心が高まり、医療情報処理に便利な汎用ソフト開発の機運が盛り上って来た。したがってここでは、医療情報処理用語の仕様そのものではなく、仕様決定における諸問題と開発の方向付けなどに関して私見を述べて参考に供したいと思う。

### 2. 医療情報処理用語の必要性

この表題から誤解を受けるかも知れないが、医療情報処理用にオールマイティな言語を一つだけ準備すれば事足れりというわけではなく、また他分野の情報処理用語とは全く異なったものが医療用には必要であるというわけでもない。ここで言いたいことは、他の各種の応用分野における問題と同様に医療の分野にもいくつかの特徴があり、その特徴に適合した言語を準備すれば医療における研究・開発・運用などの面で何かと便利であろうということである。

そこでまず応用の分野として病院を想定して説明する。具体的にコンピュータの適用業務に関しては他にゆずるが、大まかに管理業務、臨床・検査関係および医学研究に分けられる。

管理業務……予約、会計、集計、ベッド管理、薬品管理など

臨床・検査関係……カルテ管理、検査室自動化、情報伝達、自動診断など

医学研究……情報検索、統計分析など

これらの情報処理は平均値としてファイルを中心とするオンライン処理とバッチ処理に集約することができる。ただし各業務でファイルが別々に扱われるのではなく、お互いに関連を持っている。また、各ファイルのデータ構造も処理機構がかなり異なるので多少複雑なファイル体系となる。これらのファイル間の共通なキーはID番号(患者識別番号)である。

入出力の方は、各種自動化機器からのオンライン入力、医用端末装置からのオンライン入力、およびキャラクタディスプレイ等による対話型の入出力と、ラインプリンタからのバッチ出力などである。

したがってこれらの処理システムを開発する為の言語として考えられるのは、データベースにオンライン機能と対話型処理機能を加え合せたようなものとなる。ただし、応答速度に対する要求は多少強くない反面、コストに対する要求は非常に強い。したがって、この言語はミニコンにインストールされる必要がある。ミニコンが適している理由はこの他にもある。医療における情報処理システムは現在研究開発期にあり、部門ごとにコンピュータ化を進めて行く必要がある。この様な状況においては、初期投資額が少いこと、要員の教育および関係者の理解を得つつグレードアップして行けること、開発したアプリケーション・ソフトを他の病院に適用できることなどが重要である。ミニコンはOSが簡単であり、かつグレードアップをミニコン・ネットワークにより比較的容易に達成できる点で病院のコンピュータ化に最適であると考えられる。

現在これらのシステム開発は、ほとんどアセンブリ言語を使用して行われているので、開発・メンテナンスがかなりコスト高になっている。一部で拡張FORTRANや拡張BASICが使用されているが、十分ではない。ミニコン用のデータベース言語としてはMUMPS(医用)やIMAGE(汎用)などがあるが、前者はオンライン処理が困難であり、後者はデータ構造に関して検討する余地がある。

### 3. データベース機能

データベースをデータの大きな集合あるいはファイルの集合とし、汎用の処理システムが付加されたものをデータベース・マネージメント・システム(DBMS)とする。またDBMSはデータベース(DB)の定義機能、DBの処理機能、DBの更新機能をもち、プログラムとDBとの間にある程度の独立性を持っているものとする。DBMSの言語体系は図1のように表わされる。

ここで想定しているコンピュータはミニコンであるが、DBMSを医療用言語の中核にする必要があると考える理由を整理すると、次のようになる。

- 1) 医療におけるデータ処理は、ファイル処理であるとみなしてよい。
- 2) 各ファイルの相互関係が強く、統一的な管理が望ましい。
- 3) ファイル処理に比べて、演算的処理は比較的単純なものが多い。
- 4) システムが開発期であるので、変更や拡張がムンぱんに起る。

#### 3.1 データ構造、ファイル構造

管理業務に対しては一部可変長レコードをもった階層構造もしくは2次元の表が基本となるが、臨床・検査室業務に対しては動的割り付け可能な木構造が適している。たとえ検査成績ファイル为例にとると、検査項目の数は非常に多いが特定の患者にとっては非常に少なく、しかも個人差が多く、かつ同一人でも検査日によって項目が変わるのが普通である。これは $m$ 次元のスパースなマトリックス

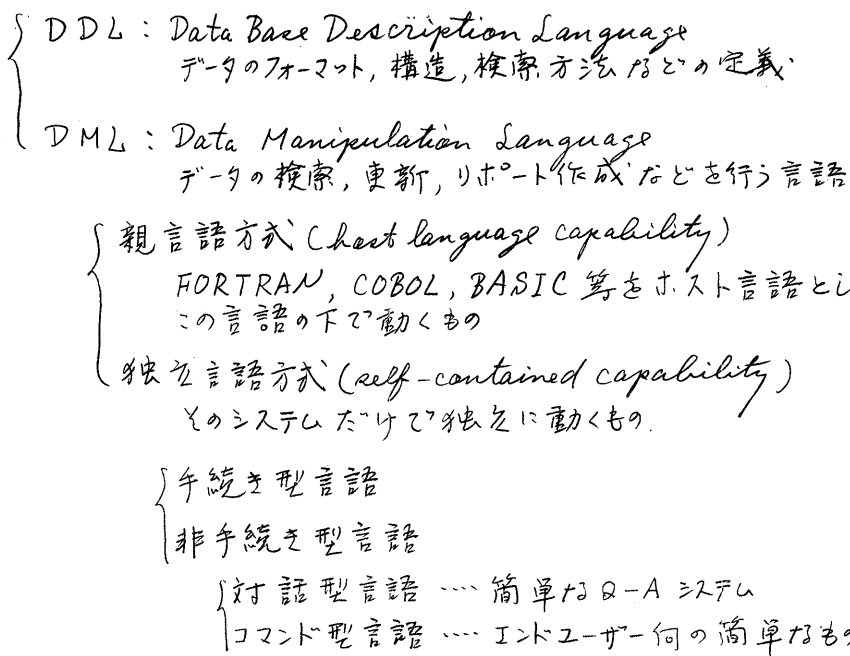


図1 データベース・マネージメント・システムの言語体系

に相当する。各座標抽に階層関係を対応させると、これは木構造で表現される。各座標点の節あるいは葉に対応し、各データがここに記録される。そこで、データの追加・削除を許しかつ各時点で不要な要素を一切もたせないようにすると、動的割り付けを必要とする。MUMPSではこの概念で全てのデータを表現されており、しかも数値データと文字列データをたぐみに融合させている。これがディスク中で表現されているので、融通性は高い反面応答スピードが極めて遅くなっている。例えば、XD(2, 13, 5)は図2のような構造で表現できる。もしここでXD(2, 13)を削除すれば、図2の点線内の全要素が削除される。

このような構造は、データベースに対する対話的処理には向くが、総計などのバッチ処理には不適当である。

オンライン・リアルタイム処理とバッチ処理の両方をカバーするにはデータ構造をある程度単純化し、検索のスピードをあげる必要がある。IMAGIEではレコードの中の検索用のキイ項目を独立なファイル構成にし、キイファイルの検索はハッシュ関数で行い、対応するレコードはキイファイル中のポインタを使って取り出す方式を採っている。キイファイルとレコードファイルの間にはお互いに1対1の関係を許しているが、少し複雑な構造を表現しようとすると相当々っかいになる。

この両者はお互いに対照的であるが、ミニコンを前提とすればいずれにした構造を限定しなければシステムがいたずらに大きくなり、かえって処理効率が落ち

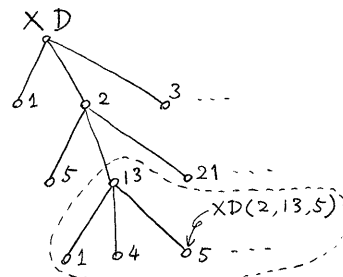


図2 動的な木構造の例

ることにもなりかねない。

### 3.2 データベース言語

データベース定義言語 (DDL) はパラメータ指定型の簡単なもので十分である。データの機密保護機能を重視すべきである。

データベース処理言語 (DML) としては、ホスト言語と対話型もしくはコマンド型の言語を準備する必要はある。ホスト言語にはバッチ処理とオンライン処理を受け持たせる。コマンド型言語の方はマン・マシン・インタラクション機能としこの簡単な Q-A および簡単なデータの更新および作表などを受け持たせる。後者の機能は医療において不可欠である。

ここで特に重要な点は、エンドユーザーが容易に修得できるような理解しやすい言語であることが医療では特に要求されるということである。したがって、暗黙の定義機能をできるだけ取り入れて、初心者でも容易に使用できかつ上級者は高度な処理が可能であるような言語が望ましい。IMAGIE はこの点がよく考慮されているようである。

### 4 データ処理言語

ここでは DBMS 用のホスト言語を中心に考えてみる。ホスト言語としてはコンパイラ言語とインタプリタ言語が考えられる。前者は実行時のスピードが速く、後者は手軽にプログラムを開発・管理できるというメリットがある。

医療においてはプログラムの開発と実際のデータ処理を併行に行う必要性が高く、しかもプログラムの修正が多いと考えられるので、実行スピードよりも融通性の方をとることにしよう。したがってインタプリタの方が望ましい。また、実行時にその度ソースステートメントを解釈する方法をとれば、マルチプログラミングおよび小規模の TSS が比較的容易に実現でき、OS も小さくすむ。

実際には、FORTRAN または BASIC にオンライン機能およびデータベース処理機能を付け加える方法が容易であるが、文字列処理機能およびレポート作成機能を重視すれば、全く新しい言語仕様を作成する方がよい。MUMPS はこの点が非常によく考慮されている。

### 5 開発について

このような言語および OS の仕様を決めるためには、医療の現場における実体を調査・分析するとともに、システム化の動向や標準化の動向と歩調を合せることも大切である。したがって学際的な問題として扱うことが望ましい。

したがって固ましくは国家的機関が開発の主体となることが望ましい。また、このようなシステムの開発を通じて、医療情報処理の体系が明確となり、システム化が促進されることを期待される。