

頭蓋骨図形データベースシステムのインプリメンテーション

Implementation of a Relational Database System for Skull Line Drawings

金森吉成

Yoshinari Kanamori

東北大学歯学部

(School of Dentistry, Tohoku Univ., * Univ. of Library and Information Science,

** Research Institute of Electrical Communication, Tohoku Univ.)

増永良文

Yoshifumi Masunaga

図書館情報大

野口正一

Schoiti Noguchi

東北大学通研

Abstract This paper describes design and implementation of a relational database system for the purpose of retrieving and processing of skull line drawings in orthodontics. QUEL like language of INGRES was adopted as end-user language. The function, also, was provided for user that the system links automatically the retrieved results to an application program of line drawing processing. The system was implemented on the top of mini-computer system using PASCAL. The B-tree file structure with 14 key indices was used for relational file. An optimization of simple method was executed for file access.

1. まえがき

図形、画像データベースの構築やこれらのデータベースシステムの研究・開発が盛んになっている[1, 2]。

特に、データモデルとして関係モデルを採用したシステムの開発が多い。関係モデルはデータの独立性や表形式による平坦性の故に、データベースシステムの専門家でない画像処理のユーザにとって、モデルが理解し易く、従ってデータベースの構築やシステムの利用が容易であるなどの特長がある。これらの特長を利用することにより、画像処理と関係データベースシステムを統合した柔軟性のある高度な画像情報処理システムの構成が現実的なものになってきた。

我々は、歯科矯正学で使用している頭蓋骨図形の処理を効率的に研究するために、図形処理とデータベースシステムを結合させた研究向きの図形データベースシステムを開発してきた[3]。

本システムの特徴の1つは、関係モデルに基づき頭蓋骨図形をハターンの特徴量、解剖学的な構造の名前といった記号レベルで抽象化してデータベ-

ス化していることである。信号レベルの図形データへのアクセスは、これらの記号レベルの論理的な図形情報から質問言語を用いて可能である。それ故ユーザは図形データの格納されている物理的ファイルの構造を知らなくても図形の本体へアクセスできる。

予想されるユーザは、データベースシステムや図形処理に熟知していない矯正医である。従って、検索された図形に対して各種処理をやる応用プログラムをユーザ自身が作成することがあまり期待できない。そこで、他の特徴として応用プログラムが作成したプログラムを検索結果と自動的にリンクする機能をシステムに持っていることが挙げられる。このようにして、親言語埋め込み方式で応用プログラムを書かなくても、ユーザは応用プログラムを利用することができる。

本稿では上記の特徴を持つ関係データベースシステムをミニコンピュータ上にインプリメントする方法について、以下に説明する。

2. システムの概要

本図形データベースシステムの構成は、図1に示すように図形処理システム[4,5,6]とデータベースシステムの部分から成っている。図形処理システムについては、データベースシステムに関連する部分について簡単に説明する。

2.1 図形処理システム

図形処理システムでは頭蓋骨図形をファクシミリを用いて入力した後各種処理をして、図形の特徴量を抽出する。

歯科矯正学の研究目的のために、図形のデータベース化は3段階のレベルで行なう。図2に示す如く、図形の抽象化の程度により、(1) 入力の量子化(1792×2320点)による点集合のレベル、(2) 解剖学的な構造成分名によるレベル、(3) 図形の特徴量、特徴点によるレベルがある。

図1のLDB(Line Drawing Database)には(1)のレベルのデータが入る。一方、RDB(Relational Database)には、(2)、(3)のレベルのデータが関係表現で格納される。

線図形関係(図形番号、成分名、格

納場所)、特徴点関係(図形番号、特徴点名、X座標値、Y座標値)、特徴量関係(図形番号、特徴名、特徴量)の3関係が導出された。

線図形関係の属性、格納場所はLDB上のどこに図形データ(1)があるかを関連付ける。すなわち、LDBとRDBとの間をリンクする役割を持っている。

図形処理システムから、(1)、(2)、(3)をデータベースに格納するエーディティが用意されている。

2.2 図形データベース(LDB)のファイル構造

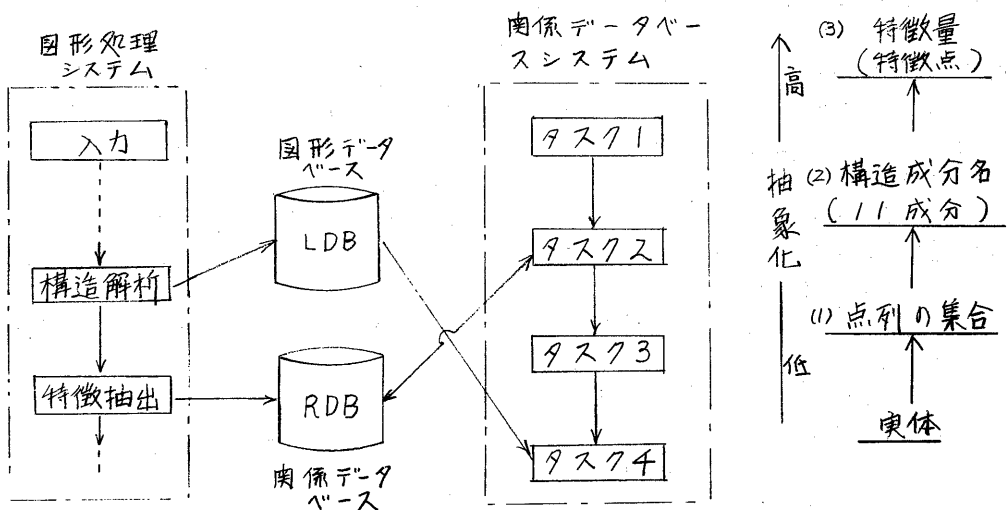
LDBのファイル構造は図3の如くである。LDBはランダムファイル(512バイト/レコード)である。

図形を構造解析した後の図形記述は図形を構成する線分や線分間の接続関係を表現する4種類の表、構成成分名リスト、線分を構成している点列のX Y座標値から与えられる[6]。

LDB上では、図形1枚当り220レコードを占める。点列のX、Y座標値、図形記述の表はそれぞれ1レコー

図2 図形のデータベース化

図1 システム概要



ドと176レコードから格納される。

ランダムファイルのレコードポインタは、ミニコンピュータの16ビット語長のため最大32767である。それ故、ランダムファイル1個につき、 $32767/220 \div 148$ 枚の図形が入る。LDBは複数のランダムファイルから構成されるから、図形などのランダムファイル上にあるものの管理が必要である。この目的のために図形管理テーブルがLDB上に用意されている。

2.3 関係データベースシステム

このシステムは、Coddによる分類に従うと[7]、最小限の関係システム(minimally relational)になる。すなわち、構造は関係表現であり、最小限の関係処理能力を持つがIntegrityは持っていないというシステムである。

本システムでは、質問言語としてINGRES[8]で用いられているQUELと準じた言語を用意した。QUELのサブセットになっている。QUELの構文は、

RANGE OF variable list IS RelationName
Command [result name] (Target List)
where Qualification

で与えられる。

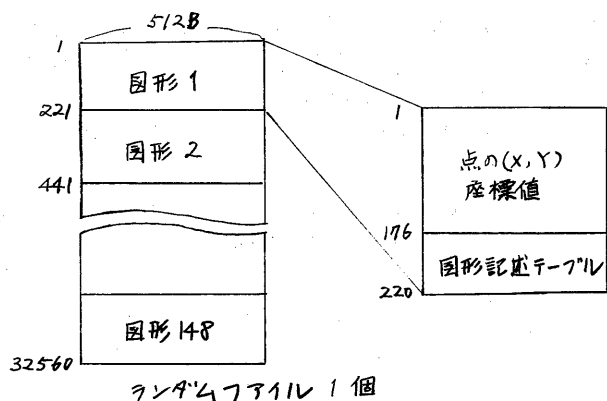
コマンドとして、Retrieve, Insert,

Rewrite, Delete がある。QUELとの違いは、Insert, Rewrite, Deleteの場合に關係演算に制限を加えていることである。すなわち、これらのコマンドでは1つの關係上でのみ關係操作ができて、結合処理ができない。このデータベースシステムが検索主体であり、ユーザによる更新がほとんどなりこと、またユーザのレベルを考慮すると更新する操作に対して制限を加える必要があることなどの理由による。

その他の違いは、現在のバージョンではAggregate functionの機能を持っていないこと、さらにQualificationではユーザがConjunctive Normal Form(CNF)の型で書くという制約を持っている。CNFの型で書くことに対する制約はユーザにとってほとんど問題にならないものと思われる。

一般に、質問言語を親言語に埋め込むことにより検索結果に対して、応用プログラムを適用するという方式がデータベースシステムでは採用されている。この方式は汎用的であるが、図形処理の応用プログラムを書けないユーザにとってはデータベースを利用することができない。そこで、本システムで採用した方法は、応用プログラムとユーザが別であるという前提に基づきこの両者を結合するためのリンク機能

図3 図形データベース(LDB)のファイル構造



SLD#	REC#	File#
1000	1	1
1001	221	1
1002	441	1
...	...	...
1148	32341	1
...	...	...

図形管理テーブル

をシステムに用意し、応用プログラムをコマンド化することである。

データベースから検索された図形データをリンカーファイルに入れると応用プログラムが起動されてこのファイルの図形データを処理する。応用プログラムによってリンカーファイルの構造も違うから、応用プログラムとそれのリンカーファイルの対応を用意し、システムが管理する。ユーザは次のようにしてリンク機能を使用する。

```
{ QUELによる文 }  
// LINK(A)    (//DUMMY)  
// MDOVER
```

この例では、応用プログラム MDOVER がリンカーファイル A を使用することを意味している。一方、応用プログラムとリンクしないときは、DUMMY を入力する。

3. インプリメンテーション

3.1 システム環境

インプリメントするために使用したミニコンポエータシステム (TI990) の環境について説明する。

オペレーティングシステム DX10 は Unix に似ている。Tree 構造のファイル管理システムを持ち、順ファイル、ランダムファイル、B-tree ファイルが利用できる。ファイルは 86 オバイトを単位として管理している。B-tree ファイルには 14 個のマルチキーインデックスがあり、1 個の主キー、13 個の副キーから成る。

エディターでは 1 レコード最大 240 バイトの長さの順ファイル扱う。このエディターで作成した順ファイルを B-tree ファイルに、逆に B-tree ファイルを順ファイルに変換するコマンドが準備されている。

ユーザ ID を入力したもとの 1 タスク (Unix のプロセス) のアドレス空

間は 64 KB である。64 KB を越える場合はオーバーレイか Inter Task Communication という機能を利用する。

システムとは BCI (Unix の Shell) コマンドを介して会話する。

本データベースシステムのインプリメンテーションの基本方針として、上記 OS がサポートしている機能を利用してユーザプログラムとして作成することにした。インプリメントに使用した言語は PASCAL である。この方針によるインプリメンテーションの特徴を挙げると、以下のようになる。

(1) PASCAL で B-tree ファイルを操作するための Procedure (key & File) が準備してある。key & File では、ファイルへの入出力アクセスが文字 (packed Array) である。

関係の属性によって、文字、整数、実数が混合している場合が一般的であるが、本データベースシステムではすべて文字で表現する。従って、演算の場合には文字 ↔ 数値変換を行なう。

(2) 文字で表現する利点はエディタを使ってデータを作成し、B-tree ファイルへ変換することかひきることである。それ故、データベースの作成や編集、修正といった作業がスクリーンエディタで簡単に行なえる。多量のデータをデータベースに挿入するときや書き換えるときなどは、質問言語で操作する場合に比べて、エディタを利用する方が便利である。一方、1 タップルの長さが 240 バイトに制限されるといふ欠点もある。

(3) 同時にオープンできる B-tree ファイル数は 16 個である。

key & File にはファイルをアクセスするコマンドが 31 種類あり、キーによる read, write 等のすべてのファイル操作がコマンド名の指定だけで可能になっている。

以下に図 1 のタスクを説明する。

3.2 タスク1

タスク1は質問文を入力して単語解析、構文解析をするコンパイラである。図4にタスク1のブロック図を示す。

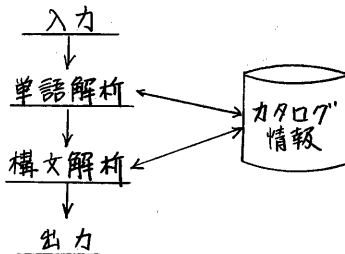


図4 タスク1

入力では質問文を1文字ずつ読み込んで記号/(最後を除く)かゝるまでArrayに入れる。

単語解析では入力文を単語に分解して、Terminal Word (RANGE, OF, IS ...) Relation Name, Variable (X1, X2 ...), Literalの4つのクラスに分ける。単語とクラスから成るテーブルとして出力する。

構文解析では、2.3節の質問言語の構文に照らして見て正しいかどうかの判定をする。解析では、3つの部分、すなわち、RANGE文、COMMAND文、QUALIFICATION文に分けて構文の正当性を検査する。

出力はテーブル形式にしてファイルに格納する。

タスク1での制限として、入力文字数が700字、Range文は10個、Qualificationの項数(CNF表現)は70個などがある。これらの制限は変更可能であるが、タスクのアドレス空間64KBによる制約で上限が決まる。

3.3 タスク2

タスク2のブロック図を図5に示す。このタスク2は、データベースシステムの効率を決める重要な部分である。

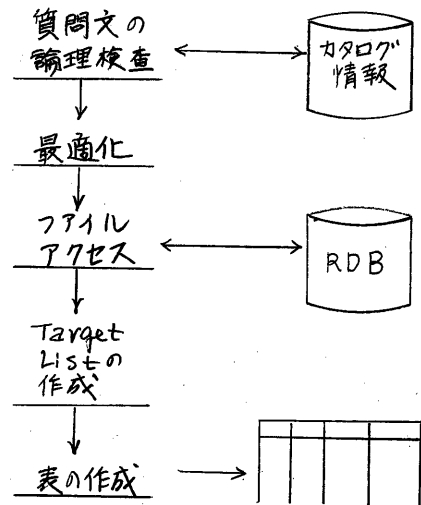


図5 タスク2

論理検査では、関係と属性の対応関係を調べることで、Target ListとQualificationの部分との結合性(Connectivity)を調べることの2つの部分から構成される。結合性の検査はINGRES [8]と同様にincidence matrixによる方法を採用した。

前者の検査は、例えば、

RANGE OF X1 IS RD

...

where X1.SEX='F'

...

のような場合に、RDの関係にSEXの属性があるかどうかのチェックをする。これらの検査には、関係名、属性名の入ったカタログ情報を参照する。

QUELのような高水準な非半統一言語では、入力した質問文をシステムが解釈して、データベースの中での条件を満足するデータへ高速にアクセスできなければならない。これは最適化問題として知られており、INGRES, SYSTEM-R [9]のアルゴリズムが有名であるが、他にも多数の研究が行なわれている [10]。

最適化は関係データベースの効率を決定するという意味でインプリメントにおいては最も重要な課題である。

本システムで採用した最適化法は、3.1節で述べた B-tree ファイルの特長を充分に利用するという方針に依存しているのび、関係のファイル上での実現法を以下に説明する。

(1) 1つの関係は1つの B-tree ファイルで実現する。

(2) 関係の属性は、B-tree ファイルのキーによって表わす。すなわち関係のキー候補はファイルの主キーを、その他の属性にはファイルの副キーを対応づける。ファイルのキーが14個あるから、属性が13個(後で説明するが1つは別の目的に使用)を越えれば、残りはキーにならない。

このように、属性をファイルのキーインデックスで表現するから、属性の実現値は大小、又はアルファベット順に常にソートされてファイル上に格納されている。従って、キーインデックスで実現されている属性の検索は高速にできる。一方、キーインデックスで実現されていない属性の検索は全数のサーチになる。

現在のバージョンで許している Qualification の構文は4種類のカテゴリーに分けられる。例で説明すると、

- (1) $X1.AGE \geq '14'$ のような右辺が定数である場合
- (2) $'14' \leq X1.AGE \leq '20'$ のような両側が定数になる場合
- (3) $X1.At1 \leq X1.At2$ のように同一関係内の属性値の比較
- (4) $X1.SLD\# = X2.SLD\#$ のような関係間での比較

Qualification では、これらの条件項が AND でつながっている。

(1) と (2) は同じわり、1つの関係上の制限が与えられる。(2) は、(1) の適用を2回する。

(3) は、同一関係内での属性間の比較であり、全数のサーチが必要となる。

(4) は、関係間での結合である。

以上の事から、本システムでの最適化法は、以下に述べる簡便な方法を採用した。

[I] Qualification の条件式の各項を上記 (1) ~ (4) のカテゴリーに分ける。

[II] (1) と (2) のカテゴリーは属性がファイルのキーになっているかによって、処理の優先順位を下図の如くに決める。

	key	non key
カテゴリー-(1)	1	3
カテゴリー-(2)	2	4

[III] カテゴリー(3)を5番目の処理順位にする。

[IV] カテゴリー(4)を最後に処理する。複数個あるときは、書いた順序にする。

ファイルへのアクセスは上記の優先順位に従って処理する。

次に、ファイルへのアクセスの方法を説明する。

図6に示すように、1つの関係に対して同一構造を持つ2つの作業ファイルを用意する。

ステップ1

1番目に条件項を処理したとき、条件を満たすタプルを原関係から取り出し、作業ファイル1に入れる。

ステップ2

2番目以後では、条件を満たさないタプルに対して識別の目印 "D" を付けて rewrite する。これは作業ファイル1の上で実行する。カテゴリー

(1) ~ (3) の条件項がなくなるまでこのステップをくり返す。

ステップ3

識別の目印“D”が付かないタプルのみを作業ファイル2へ入れる。

ステップ1 ~ 3 までは、すべての関係について実行する。

ステップ4

作業ファイル2上でカテゴリー(4)の結合操作を実行する。等結合操作である。2つの関係の属性の値を調べて同値のときタプルの識別子を付ける。すべての結合について同様に調べる。

ステップ5

識別子を見て、関係間で同値であるタプルのみを取り出し、作業ファイル1に入れる。すべての関係間で同値を調べる。

ステップ2で“D”の目印を付けてタプルを除去しない理由は、タプルの除去に伴うB-treeファイルの再構成の時間が目印を付けてrewriteする時間比べて大きいためである。

ステップ4ではキーインデックスによりソートされているから、merge join 法による結合と等価である。

ステップ4の識別子には、キーインデックスの14番目を使用している。

Target Listの作成では、ステップ5で求めたタプル集合に対して、射影を実行して結果の関係を作る。

最後に、結果の関係を見易くするために表形式にする。

以上がタスク2の処理である。

3.4 タスク3

タスク3では、結果の関係と図形処理のアプリケーションとをリンクする。

結果の関係には、論理的な図形データすなわち、図形番号、構成成分名が与えられる。そこで、これらの属性値に基づきLDB内の図形データを取り出すという機能が必要となる。タスク

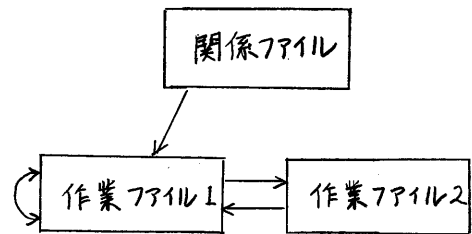


図6 作業ファイル

3でこれ进行处理する。LDBから検索した図形データをリンカーファイルに格納する。リンクコマンドLINK(A)の変数Aは、リンカーファイルの構造を意味していて、応用プログラムと対になっている。図7のように与えられている。

変数名	応用プログラム
A	MDOVER
A	NFOVER
⋮	⋮
B	MONDB

図7 リンカーファイルと応用プログラム

変数名と応用プログラム名との対を調べて、正しければリンカープログラムを起動する。このプログラムの実行終了後に応用プログラムを起動する。

リンクコマンドの代りにDUMMYがあるときは、タスク2の表を出力して終了する。

リンカーファイルの構造は、LDBの構造と同一であり、検索された図形のみをこのファイルに格納する。作業ファイルのような働きをする。現在あるリンカーファイルは10枚の図形データが入るもの(A)と1枚のもの(B)である。図形処理のアプリケーションと合わせてユーザがリンカーファイルの

構造を定義することもできる。

3.5 タスク4

この部局は、図形処理の応用プログラムであり、リンカーファイル作成後にシステムによって起動がかけられる。

応用プログラムに開放されているから、目的によって種々のものが考えられる。

応用プログラムをインプリメントしたときに、SCEIのコマンドにして05に登録すると同時に、図のテーブルにも記入する。

以上で、関係データベースシステムのインプリメンテーションの概要を述べた。

4. むちび

頭蓋骨図形の検索と処理を目的とした関係データベースシステムのインプリメンテーションの方法について述べた。本システムはミニコンピュータ上にPASCALを用いてインプリメントした。

現在のバージョンでは、タスク1が約900ステップ、タスク2が約2500ステップ、タスク3が約300ステップ（コメントを含む）である。

タスク4の応用プログラムは、すべてFortranでインプリメントしている。現在、5種類ある。今後、歯科矯正学の研究目的によっていろいろな応用プログラムが開発されていくものと思われる。

本システムのような質問言語で検索することは、一般ユーザにとって使い易くはない、より高水準なエンドユーザ言語の開発が必須である。

我々は、この方向に沿って既に研究し、基本的設計は完了している[11]。

さらに、本データベースシステムを利用した歯科矯正学症例教育システムの研究[12]も行なっており、知識システ

ムの構築を目標にしている。

謝辞

本システムのインプリメントに協力して戴いた関専司君（現IBM）、原島秀次君（豊橋技科大）、山本良樹君、加藤靖先生（仙台電波工専）に感謝する。

文献

- [1] 横矢直和、田村秀行：画像データベース研究の動向、電統研彙報 45, 9, 10 57-71, 1981
- [2] IEEE Computer Vol 14, No. 11, Nov. 1981
- [3] Kanamori, Y. et al: Design of a database system for skull line drawings processing in Orthodontics based on the relational model MEDINFO 80, North-Holland 1980
- [4] 藤岡芳夫他：ミニコンによる線図形処理ソフトウェアの開発、信学会 PAL79-44, 1979
- [5] 瀬々哲夫他：歯科矯正学における顎骨の形に関する特徴抽出、信学会、PRL79-45 1979
- [6] 金森吉成：頭蓋骨図形の構造解析 信学会、PRL83, 1983
- [7] Codd, E.F.: Relational database: A practical foundation for productivity Comm. ACM, Vol 25, No 2, 109-117, 1982
- [8] Stonebraker, M. et al: The design and implementation of INGRES ACM TODS, Vol 1, No 3, 189-222, 1976
- [9] Astrahan, M.M. et al: System R-Relational approach to database management, ACM TODS, Vol 1, No 2, 487-491, 1976
- [10] 北上始, 他: 関係データベースシステム RDB/V1 の最適化技法, 情報処理論文誌 Vol 24, No 3, 302-310, 1983
- [11] 金森吉成, 増永良文: 歯科矯正学診断サポートシステムの設計, アドバンスド・データベース・システム シンポジウム, 昭和56年12月
- [12] Kanamori, Y. and Masunaga, Y.: An Orthodontic case study instruction system based on a relational database system IEEE Quarterly Bulletin on Database Eng. Vol. 6 No. 4 Dec. 1983