

## 4H-4

マルチメディアデータベースシステムの  
ユーザインタフェース

池田 秀人

(広島大学総合情報処理センター)

## 1. はじめに

マルチメディアデータベースシステムをユーザから見たときどのように見えるのがもっとも自然で、かつ使いやすいであろうか。関係データベースの定義域 (Domain) のいくつかがグラフや画像になっていればそれで充分という考え方も成り立ちうるがこれでは情報をデータベースに入力する前のイメージと検索した時に出力されるイメージが掛け離れすぎてかえって判りにくくなってしまふ。従って入力されたイメージがそのまま自然に検索されることが必要となろう (原画再現性)。しかし入力されたものがそのまま出てくるだけではデータベースとは言いがたい。やはり幾つかの画像をもとに新しい画像や表を簡単に作り出せなければいけない (合成画面導出性)。また文字・数値・グラフ・画像などのいろいろな表現を持つデータを一元化された方法で操作する必要がある。(データ操作の一元性)。このような要求を満たすマルチメディアデータベースのユーザインタフェースはどのようなものであろうか。本報告ではその一つの方法として認識画面蓄積法を提案するものである。

## 2. 認識画面蓄積法によるデータベースのユーザイメージ

データベースの内部的な表現は別として、データベースをエンドユーザの側からみたときそれがどのように見えるのが最も自然でかつ使いやすいであろうか。CADで作った図面、地図、説明文を書き込んだ写真、図入りのワープロ文書などの蓄積としてデータベースが見られることが支配的になるとこれらをそのままの形で取り出すことがデータベースの最も重要な機能となることは間違いない。この場合数値、文字、グラフおよび画像がすべて画像レベルで合成され視覚化されたもの (ここではこれを「編集画像」とよぶことにする。) を一つの単位として蓄積されたものがデータベースということになる。この方法を「編集画像蓄積法」ということにする。光ディスクを使った電子ファイリングシステムの多くはこの方法によっている。蓄積される画面を作成する過程ではイメージスキャナなどによる画像データとワープロによる文字や数字とプログラムやマウスで描いたグラフを合成して一枚の画面が作られるがそれが蓄積される段階では一枚の画像データとして蓄積される。しかしこの方法をとると、画面上に書かれた数字は画面が検索され呼び出された段階ではもはや数値としての

性質は有しておらず、従って計算の対象にはならない。文章も元はワープロで作ったにも拘らずそれが画面上に画像として表現され拡大・縮小されてしまうとワープロによる挿入・削除などの編集ができなくなってしまう。この場合検索は別に用意される索引によって行なわれる。呼出しのキーワードは既に保存される画面上に視覚的には見えているにもかかわらずキーボードなどからこれを入力しなおしてやらなくてはならない。

CADによる作画システムの画面保存法はこれをもっとうまく解決している。この場合人間による作画の一つ一つのステップを再現できるようなパラメータの集まりをベクトルにして保存している。この方法を「動作ベクトル蓄積法」と呼ぶことにする。この方法なら画面上に現れた数値を計算の対象にすることができる (可能性がある) し、文章をワープロで再び編集することもできる (可能性がある)。しかし、このシステムで蓄積されたベクトル群 (すなわち画面) をそのまま検索の対象として使おうとすると問題が大きい。画面の中に書かれた文字列がその画面に描かれている人の名前を意味しているなどの画面の構成要素とその関係に関する情報が無いと意味的な検索の対象にならないからである。また保存画面をすべて (多くの場合、あまり長くない固定長の) ベクトル群で表現しようとするとう写真などの画像データを組み込むことが難しくなる。

「認識画面蓄積法」はこのような問題を解決しようというものである。ユーザによる画面作成は「編集画像蓄積法」のようにイメージスキャナやワープロやマウスあるいはディジタイザを用いて作成するが作成された画面は一定の法則でフォーマット化されたベクトルにより保存される。「動作ベクトル蓄積法」と異なるのは

- (1) 画面の構成要素の意味のあるまとまり (部分画面) に関する情報もベクトルとして含まれていること、
- (2) 画面には視覚的には出てこない事実もベクトルとして含みうること、
- (3) 画像データもその実保存場所を示すアドレスベクトルとして含みうること、
- (4) 画面の構成要素の作成された過程 (歴史) が (これもベクトルとして) 記録されていること、
- (5) 同種の画面を一つのまとまりとして管理するための「ゼネラルフォーマット」 (通常のデータベース管

理システムのデータ辞書に相当する。)を持っていること、

- (6) 部分画面、画面、ゼネラルフォーマットのそれぞれの間のあるいは相互間の関係が記述され管理されていること。

などである。

### 3. 認識画面蓄積法によるデータ定義

具体的なユーザインタフェースを示すためにまずデータ定義の方法を示す。この方法によるデータベースは画面ファイルの集まりである。一つの画面ファイルは同一フォーマットを持った画面の集まりである。このフォーマットは一つの基本画面といくつかの領域構造データよりなっている。領域構造データは基本画面内の領域とその属性群からなる。複数の領域が重なり部分を持っていてもよい。また別の領域から画像操作(回転, 白黒反転, 重ね合せなど)や計算(スプレッドシートが採用している方法)により作られるものでもよい。これを示したのが図1である。

実際のデータ定義はこのような文字列を使うのは不便である。特にアドレスの指定は数字で直接与えるのは不可能に近い。基本画面の上でマウス, デジタイザ, タッチスクリーンあるいはライトペンなどのポインティングデバイスを用いて行う。図2はマウスを用いたシステムの例である。

### 4. 認識画面蓄積法によるデータ操作

データ操作もデータ定義の方法とよく似たインタフェースを提供出来る。

データ入力にはまず基本画面を呼出し入力したい項目の入るべき領域をマウスで指示すればその属性に応じた入力モードになる。画像領域ならばイメージスキャナが使える状態になるし、文字領域ならばキー入力可能になる。他の領域から属性操作関数により作られる領域なら自動的にそれが実行されその領域の画像が表示される。こうして画面が作成されたらそれを保存すればよい。

データ検索は基本画面の中に条件を書きこめばよい。ある領域がある値を持っている画面がほしならその領域にその値を書きこめばよいし、もっと複雑な条件なら変数を含む式を書きこめばよい。このような方法でかなりの検索能力を持ちうること

はQBE/OBEなどで実証済みである。

新しい画面の合成法も同様である。まず基本画面を定義し、その上に領域とその属性を定義する。この場合属性は操作関数により他の画面からのデータ変換されて生成されるものとして与えられることになる。

このようにデータ定義およびほとんどの操作が極めて均質な方法で行える。

### 5. おわりに

本報告ではマルチメディアデータベース管理システムのユーザインタフェースとして認識画面蓄積法を提案した。この方法によるとデータベースは、文字、グラフ、画像を混在させたマルチメディア文書の集まりである。しかし内部的には単なる画像データではなくどの部分が文字でありグラフであるかシステムが知っている。また同種の文書がまとめられ構造を持ったファイルとして管理される。この方法採ることによりユーザインタフェースは大きく機能改善され、かつ使いやすいものになる。

```

<データベース> := { <画面ファイル> }
<画面ファイル> := { <画面> : FORMAT ( <画面> ) = 一定 }
FORMAT ( <画面> ) := ( <基本画面> , { <領域構造> } )
<領域構造> := ( <領域> , { <属性式> } )
               <構造操作関数> ( { <領域構造> } )
<属性式> := <属性値> | <属性操作関数> ( { <属性式> } )
<属性値> := N [数値] | C [文字] | G [グラフ] | I [画像] . . .
<領域> := ( 開始アドレス , 終了アドレス )
<構造操作関数> := AND | OR | EXAND | EXOR | . . .
<属性操作関数> := NTOC [数値を文字型に変換する] |
                  CTOI [文字をフォントに変換する] | . . .
    
```

図1. 認識画面蓄積法によるデータベースの見え方

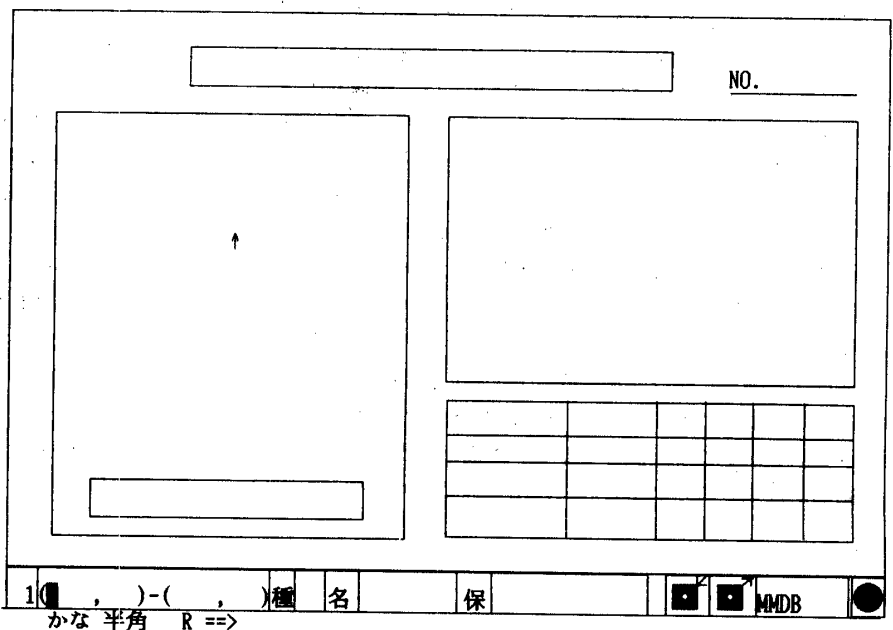


図2. 認識画面蓄積法によるデータ定義