

# 4Q-9 グラフデータモデルを用いたDBMSのマルチメディアへの対応例

山本倫子, 宮崎章二, 倉持稔, 佐藤隆子

(株)リコー ソフトウェア研究所

## 1. はじめに

グラフデータモデルは、リレーショナルモデルの表構造にネットワークモデルのグラフ構造を併せもつデータモデルである<sup>1)</sup>。このモデルは構造化されたデータに対する記述力が高い。例えば、I S-AやI S-PART-OFといった関係をグラフ構造によって表現することが可能である。

近年、画像・図形・音声などのマルチメディアデータの管理に対するニーズが高まってきた。これらのデータは、従来の数値データ型や文字データ型では表現できない。そこで、マルチメディアデータに対応するために、グラフデータモデルの基本データ型に新しいデータ型を追加した。

本稿では、まず新しいデータ型の構造について述べ、次いで、その上に構築するアプリケーションの要件を考察する。

## 2. マルチメディア対応のデータ型

新しく追加した基本データ型を、object データ型と呼ぶことにする。object データ型には、あらゆる種類のマルチメディアデータを格納することができる。これにより、数値・文字データに加えて、マルチメディアデータをフィールドに保持することが可能となる。

object データ型の実現は、光ディスク装置の使用を前提とする。データの実体の保持に光ディスクを用いるためである。

光ディスク上での object 型データは、データ本体と属性情報により構成される。属性情報には、マルチメディアデータの種類等が記録される。具体的な構成を図1に示し、以下、図中の情報(1)~(5)について説明する。

### (1) データの種類に依存しない属性情報

以下の3項目から成る。

#### (a) データ表現の定義:

この先の情報の解釈に必要な情報。

- ・バイト中でのビットの順序 (MSB, LSB の位置)
- ・ワード中でのバイトの順序
- ・int, long 他の表現方法、等。

### (b) データの復元方法:

データ本体に適用された、圧縮技法の種類や復元の手順を示す。データ本体については(4)で後述する。

### (c) データの種類:

データが何を表すものかによって、テキスト/画像/音声/実行形式/その他に分類する。複合オブジェクトはこのレベルでは表現しない。

なお、テキストには、roff スクリプト、TeX スクリプト、C プログラムなどの通常の意味のテキストに加えて、PostScript、テレテキストなど画像の記述機能を持つものも含む。

## (2) データの種類に依存する属性情報

データの種類(テキスト、画像、音声等)に応じて最低限必要となる属性情報である。例えば、画像の必須属性として、縦横のサイズ(ピクセル数)・画素の深さ・色空間の次元(白黒なら1、フルカラーなら3)がある。またオプションな属性として、物理的なサイズ(原寸が何cm×何cmか)・中間調データのパターンセルの大きさ(何ピクセル×何ピクセルのパターンで階調を表現するか)がある。データの種類が実行形

データの種類に依存しない属性情報	..... (1)
データの種類に依存する属性情報	..... (2)
ユーザの管理にまかされた属性情報	..... (3)
データ本体	..... (4)
ユーザが自由に書き込む情報	..... (5)

図1 光ディスク上の object 型の表現

An Example of DBMS based on Graph Data Model for Multimedia

Noriko YAMAMOTO, Shoji MIYAZAKI, Minoru KURAMOCHI, Takako SATO

RICOH COMPANY, LTD.

式であるならば、必須属性はプロセッサの種類である。

### (3) ユーザの管理にまかされた属性情報

(1) および (2) 以外で、特に属性として記述する必要が生じた情報のために用意される。例えば、いったんデータベースに登録した object 型のデータの更新を考える。更新後のデータを光ディスクに書き込む際に、更新前のデータへのポインタをここに記録することで、以前のデータとの関連を情報として保持できる。

### (4) データ本体

データが図形である場合、ワードパウンダリで、かつバイト中での LSB は最も左側に位置する、いわゆる X における表現をヘッダ無しで保持することを標準とする。ヘッダ付の、いわゆる X ウィンドウダンパ(XWD)形式や、スキャナで読みとったそのままのデータ等に関しては、(3)に明示して管理する。これらは、アプリケーションプログラムにて適宜、型変換を施す。

### (5) ユーザが自由に書き込む情報

光ディスクへの入出力は、セクター単位で行なわれる。そこで (1) ~ (4) を併せてのサイズがセクターサイズの整数倍に満たない場合に、余りのサイズをユーザに解放する。(必要ならば、さらにこのためのセクターを確保してもよい。) 例えば、この object 型を含むレコードの ID を書いておき、障害復旧対策に役立つ。

## 3. object データ型を用いたアプリケーションの考察

グラフデータモデルは、リレーショナルモデルの表構造を持つため、データの操作が容易である。また、関係付けの自由度が高いというネットワークモデルの利点も兼ね備えている。基本データ型として、いままでは数値型と文字型しか用意されていなかった。そこで object データ型を追加し、フィールドをマルチメディアデータに対応させた。この結果、記述可能な概念モデルの範囲が広まった。

図2に、object 型データを用いて論文の構成をモデル化した例を示す。論文の構成要素である図(イメージデータ)と TeX スクリプト(テキストデータ)は、独立したレコードである。これらのレコードは object データ型のフィールドを持つ。論文の各章もレコードで表現される。そのフィールドに、フォーマット済みの文章を保持する。

ただし、現在の object 型では、入れ子構造は許されない。すなわち、object 型データの部分としての

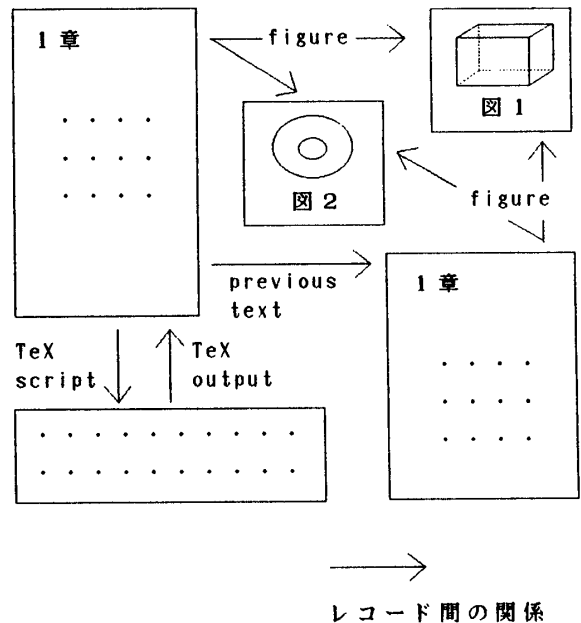


図2 object データ型を用いたデータ表現の例

object 型データは、記述できない。このため、複合オブジェクトの実現には、図2のような方式を用いた上で、アプリケーションプログラムの助けを借りる必要がある。

テキストの文字型への変換とそれに対する検索、また画像データの表示など、マルチメディアデータには各データに固有の操作が多い。これらの表示/データ変換等の操作は、アプリケーションプログラムで実現される。object 型がアプリケーションプログラムを意識した、十分な属性情報を保持しているからである。

## 4. おわりに

object データ型の追加によって記述できる概念モデルの範囲が広まった。object 型データは、属性情報を併せ持つことで、データの自由度が高いものとなった。今後、3章で述べたようなアプリケーションプログラムを作成し、その有用性を確認していく必要がある。

## 参考文献

- 1) 国井秀子：拡張リレーショナル型データモデル，RICOH TECHNICAL REPORT, No.16, pp.90-92(1987)。