

# OODB における拡張可能マルチメディアクラスライブラリ

3 X - 5

佐伯 剛幸 波内みさ

NEC C&amp;C メディア研究所

## 1 はじめに

近年、マルチメディアの普及とソフトウェア部品化の流れの中で、ライブラリのような部品を追加することにより機能拡張ができる拡張可能マルチメディアデータベースシステムが登場している。しかし、既存のシステムでは、ライブラリ部品の構成要素を入れ換えることにより容易に機能をカスタマイズするためのライブラリ部品自体の拡張性に欠けるという問題がある。

一方で、オブジェクト指向設計方法論の分野では、設計パターンやフレームワークなどの、クラス構造のレベルでソフトウェアの部品化を進め拡張性を高めるようなオブジェクト指向設計技術が提案されている。

我々は、オブジェクト指向設計技術を利用した拡張性の高いマルチメディア部品を実現することを目指し、オブジェクト指向データベース PERCIO[1] 上にマルチメディア文書クラスライブラリ (MMDCL) を構築している。本稿では特に、マルチメディア管理の基盤となるメディア格納機構と検索機構の拡張性の実現について述べる。

## 2 マルチメディア文書クラスライブラリ

現在のところ、MMDCL はイメージとテキストに対応しており、イメージはイメージ処理や類似画像検索、テキストは構造化文書管理 [2] や全文検索などの機能を実現している。

### 2.1 メディア格納機構

本節では、メディアの格納が必要とされる拡張性を課題としてあげ、それを解決するためのアプローチを示す。

#### 2.1.1 課題

メディアの格納に関しては、以下のような課題がある

1. メディアのデータベースへのさまざまな格納方法へ対応する必要がある  
メディアファイル名を格納する場合、メディアコンテンツを格納する場合などの他、多様な格納方法が要求される可能性がある。
2. メディアコンテンツの最適な管理方法が、メディアの種別やサイズにより異なる  
サイズの小さいコンテンツを処理する場合はすべてをメモリ上にロードする方が高速に処理できるが、サイズの大きなコンテンツの場合はメモリの圧迫を避けるため必要な部分のみをロードすることが必要である。また、コンテンツがバイナリ形式かテキスト形式かにより管理方法が異なる。

#### 2.1.2 アプローチ

課題で示した拡張性を実現するオブジェクトモデルとその動作について説明する。

An Extensible Multi-media Class Library for an OODB

Takayuki Saeki, Misa Namiuchi

C&C Media Research Laboratories, NEC Corporation

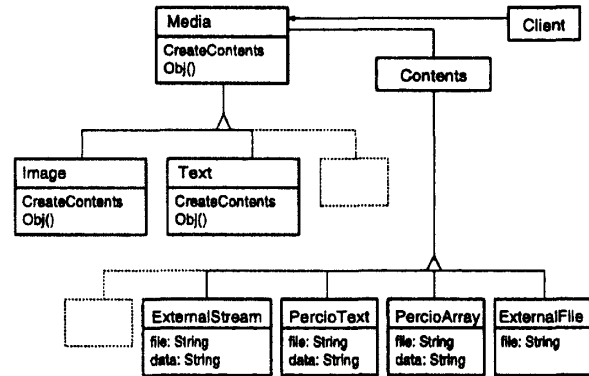


図 1: 格納に関するオブジェクトモデル

図 1 に、格納に関するオブジェクトモデルを示す。メディアを抽象化するクラスとしてメディア (*Media*) クラスを導入し、それとは独立にメディア格納機構を実現するクラスとしてコンテンツ (*Contents*) クラスを導入する。

コンテンツクラスのサブクラスである外部ファイル (*ExternalFile*) クラスは、ファイル名のみを保持する。一方、それ以外のクラスはファイル名とコンテンツを保持できる。メディアクラスとコンテンツクラスを分離したため、メディアオブジェクト生成時に設定するコンテンツオブジェクトを上記のサブクラスから選択することにより、利用者が要求する格納方法に対応することができる。また、別の格納方法が必要な場合は、コンテンツクラスのサブクラスを新しく定義し、そのオブジェクトをメディアに設定する。これにより、課題 1 を解決する。

PERCIO テキスト (*PercioText*) クラスと PERCIO 配列 (*PercioArray*) クラスは、PERCIO の機能を使用することにより必要な部分のコンテンツのみをメモリにロードすることが可能である。これに対して、外部文字列クラス (*ExternalStream*) ではすべてをメモリにロードする。メディアコンテンツを格納する場合にどちらのクラスのオブジェクトを生成するかは、メディアクラスの間数 *CreateContentsObj* で制御しており、利用者に意識させずにサイズに基づいて生成するコンテンツオブジェクトを切り替える。さらに、各メディアクラスでこのメソッドを再定義することにより、そのクラスごとに適したコンテンツオブジェクトを生成するように変更できる。これにより、課題 2 を解決する。

### 2.2 検索機構

マルチメディアデータに対する検索は、全文検索や類似画像検索などメディア固有の検索手法が存在し、かつ、さまざまな実現方式が検索エンジンとして実装されている。このため、検索機構をデータベースの機能として提供するのではなく、検索エンジンを取り込んでデータベースから使用できるようにすることが望ましい。本節では、検索機構が必要とされる拡張性を課題としてあ

げ、それを解決するためのアプローチを示す。

### 2.2.1 課題

検索機構に関しては、以下のような課題がある。

1. 同じメディアに対する複数の検索エンジンに対応する必要がある  
性能やコストなどにより最適な検索エンジンを選択できることが要求される。また、新しい検索エンジンの追加を容易に行なえる必要がある。
2. 異なるメディアに対する検索エンジンのAPIを共通化する必要がある  
APIが統一されていないと非常に使いづらい。また、新しいメディアが追加された場合に、それに対応する検索エンジンを容易に組み込めない。

### 2.2.2 アプローチ

課題で示した拡張性を実現するオブジェクトモデルとその動作について説明する。

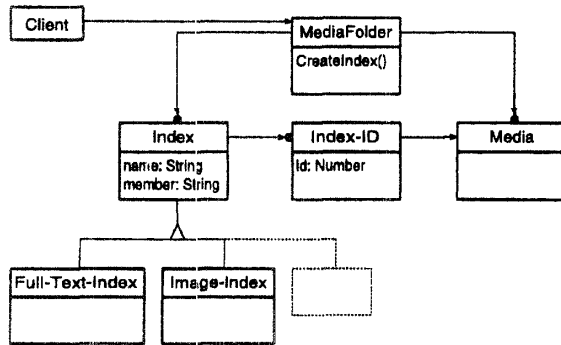


図 2: 検索に関するオブジェクトモデル

図2に、検索に関するオブジェクトモデルを示す。検索エンジンを抽象化するクラスとしてインデックス (*Index*) クラス、メディアオブジェクトの集合を管理するクラスとしてメディアフォルダ (*MediaFolder*) クラスを導入しており、メディアオブジェクト集合の検索エンジンへの登録と集合からの検索ができる。

図3に動作モデルを示す。登録では、まず、メディアオブジェクトのメンバ変数名と使用するインデックスクラスの名前を指定して、メディアフォルダオブジェクトに対して関数 *CreateIndex* を呼び出しインデックスオブジェクトの生成を行う (図3の1)。インデックスオブジェクトが生成されたメディアフォルダにメディアを追加すると、メディアオブジェクトのメンバ変数値がインデックスオブジェクトを通して検索エンジンに渡され登録が行なわれる。この時にふられた識別子 (ID) とメディアオブジェクトとの関係を、インデックス識別子 (*Index-ID*) クラスで管理する (図3の2)。

検索では、検索対象とするメディアオブジェクトの属性を指定する。指定された属性に対してインデックスオブジェクトが付加されていれば、検索エンジンに対して検索処理を呼び出し結果のID集合を取得する。そして、インデックス識別子オブジェクトを通して、IDに対応するメディアオブジェクトを取得する (図3の3)。

インデックスオブジェクトを、インデックスクラスのサブクラスである全文検索インデックス (*Full-Text-Index*) クラスや、類似画像検索インデックス (*Image-Index*) クラスから選択することにより、検索エンジン

を選択することが可能であり、また、インデックスクラスのサブクラスを新しく定義することにより他の検索エンジンを取り込むことができる。これにより課題1を解決する。また、メディアフォルダで異なるメディアを管理しており、それらに対する検索エンジンのインタフェースの共通化しているため、課題2を解決する。

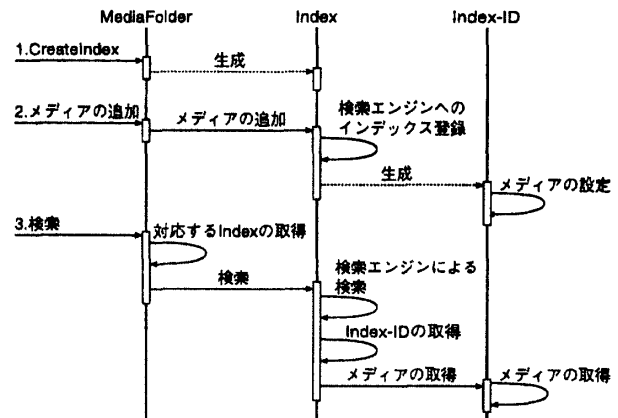


図 3: 検索に関する動作モデル

## 3 関連研究・製品

商用のものとしては、Object Design社のオブジェクト指向データベース ObjectStore用のクラスライブラリである Extended Object Managerがある。しかし、これは拡張性を考慮した構造を持っていないと思われる。

研究レベルとしては、情報検索用フレームワーク (IR-FWK)[3]がある。相違点は、MMDCLでは実用性を考慮してオブジェクトモデルをより簡単にしていること、異なるメディアの検索APIの共通化を実現していることなどである。

## 4 まとめ

メディア格納機構と検索機構の拡張性を考慮した、マルチメディア文書クラスライブラリのオブジェクトモデルと動作について説明した。

このような拡張性の実現による利点をまとめると、利用者に影響を与えずにライブラリの拡張が行なえること、変更部分が局所化されるためクラスライブラリを基にデータベースを構築した後でもライブラリの変更が行なえることなどをあげることができる。今後は、MMDCLのコンポーネント化などを予定している。

## 参考文献

- [1] 鶴岡, 他, オブジェクト指向データベース管理システム PERCIO の開発と今後の課題, 電子情報通信学会論文誌, D-I, Vol, J79-D-1, No.10, pp.587-596, Oct. 1996.
- [2] 波内, 鶴岡, SGML/HTML 文書 DB におけるテキスト格納方式の提案, 情報処理学会第 54 回全国大会, 3-205, 1997.
- [3] G. Sonnenberger and H. P. Frei, Design of a Reusable IR Framework, ACM SIGIR'95, 1995.