

マルチメディアドキュメントデータベース システムの構成方法

1 R-5

姚 左軍
東京大学佐藤 真一
学術情報センター濱田 喬
学術情報センター

1 概要

本論文は、オブジェクト指向の概念を用いたマルチメディアドキュメントデータベースシステム(MMDDBS)の構成方法についての一つの方策を提案する。本提案はMMDDBの動的な特性という問題の解決法の考察を行ない、広義の抽象オブジェクトモデルを導入し、多種類のドキュメントに適合するDBS構築方法を論じる。

データベースの構成は一度の処理で全体が構築されるのではなく、データベースの使用に伴い徐々に構築されて行くという形態が現実的である。それに対して、従来のデータベースの構築方法は、データベースの将来的な拡張のされ方について展望し、すべての論理構造(例えばRDBの関係表)を定義し、全部の必要な物理領域が一回で割り当てられるという形態をとっている。しかし、データベースの構造の変換、データベース中のデータの量の領域超過には、論理構造の再定義、領域をもう一度割り当てるといったデータベース構造の再構築を行う必要がある。

一方、MMDDBには、1. ドキュメントのデータの量を予測できないから、データベースの領域の大きさを予測できない、2. ドキュメントの種類が多く、データベースの使用中に新たなドキュメントの種類が増える可能性もある、3. 種類が異なるドキュメントに対応した様々な処理方法を提供する必要があるなどの実際の問題が挙げられる。MMDDBにおいては、上述のようによりデータベースの動的な振舞いが求められる一方、種類が不定の多様なドキュメントを扱う必要がある。

2 システムの特徴

現在、プログラムの実行環境の変化に対応するため、様々な対策が出てきた。例えば、ウィンドウシステムは実際のウィンドウのサイズや機能設定などの可変性

に対して、リソースとデフォルトファイルを使用して問題の解決を図っている。しかし、このような方法だけでは、前節で述べた動的性の要求は満たさない。特に、新たなドキュメントの種類を増やす場合、対応のクラスを定義するため、プログラムをリコンパイルする必要がある。そうすると、ドキュメントを処理するモジュールを全てシステムに埋め込む方法には、一つのモジュールの変化と新たなモジュールの追加に伴い、システム全体のリコンパイルが必要となるので、システムの柔軟性は弱くなる。

ところが、オブジェクトの記述によって、同類のオブジェクトは同じクラスに所属するので、二つの同類のオブジェクトは、ただ属性の値が違うのみである。従って、データベースはオブジェクトの属性の値とメソッドの情報(メソッドの使用情報)のみを保存し、その処理モジュールをDBSから抽出し、独立に扱うことができる。この場合、処理モジュールの変換及び追加に対しても、システムの全体に影響を与えないことになる。図1ではこのような処理方法を表現する広義の抽象ドキュメントオブジェクトのモデルを示している。この広義の抽象オブジェクトモデルは同類のドキュメントオブジェクトと処理モジュールの中の抽象のドキュメントオブジェクトと一対多の関係になるように構成する。

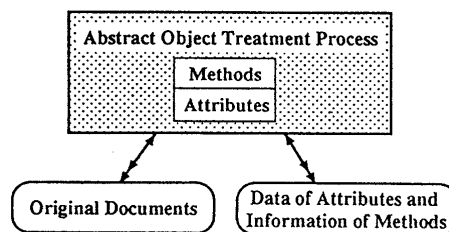


図1: 広義の抽象オブジェクトモデル

3 システムの仕組み

MMDDBS(図2) はシステムカーネル、検索 GUI からなる。そのまわりに各種類のドキュメントを物理的に処理する独立オブジェクトプロセスがある。オブジェクトとの接続インタフェースはシステムから提供される。

3.1 システムカーネル

システムカーネルは四つの独立モジュール(管理マネージャー) からなり、それぞれDBSの四つの部分の管理を担当する。

3.1.1 GUI マネージャー (GUIM)

GUIMは主に、1. コマンドを受理するGUIのコントロール、2. 受理したコマンドの分析、3. 分析したコマンドを一定のフォーマットで他のマネージャーへの転送、4. 他のマネージャーからの表示情報をGUIに出力、を担当する。

3.1.2 オブジェクトプロセスマネージャー (OPM)

ドキュメントオブジェクトが一種ずつ広義の抽象オブジェクト、いわゆるオブジェクトプロセスを作る。具体的なドキュメントを処理する時、オブジェクトプロセスマネージャーは処理するドキュメントによって適切なプロセスの実行ファイルを選び起動し、データと処理の命令を起動されたオブジェクトプロセスに転送する。初期値設定によって、同時に n 個までのオブジェクトプロセスを起動できる。

3.1.3 インデックスマネージャー (IM)

IMはインデックス情報の登録とキーを用いたドキュメントの検索を管理する。登録時、OPMから登録情報を受け、既存のインデックスに入れる(その時関連のインデックスがなければ、まず新たなインデックスを作る)。検索時、GUIMから分析した検索コマンドを受理し、検索条件と論理関係によって、インデックス構造からOID(ObjectID)を探し、条件を満たすOIDsを抽出し、処理コマンドと一緒にOPMに送る。

3.1.4 ストレージマネージャー (SM)

ドキュメントを表すデータ情報は二つある。一つはドキュメント「原稿」である。一つはドキュメントの処理のため、必要な情報を含むオブジェクトである。それらはそれぞれ二つのデータファイルの中で保存され、その内、オブジェクトデータファイルはSMに管理される。データファイルへのアクセスの要求に応じて、SMはオブジェクトのデータを主記憶と二次記憶間で交換する。複数のユーザーが同時に同じオブジェクトをアクセスする時には、排他メカニズムを用いて競合解消を図る。

3.2 システムとオブジェクトプロセス

通常、オブジェクトプロセスは実行ファイルの形式で存在する。オブジェクトプロセスの実行ファイルを作るのはアプリケーションの仕事であるが、オブジェクトプロセスとシステムカーネル間の通信、オブジェクトプロセスの制御関数などのインタフェースはMMDDBSに提供される。即ち、アプリケーションは処理するドキュメントの特徴によってクラスを定義し、処理関数をシステムが提供したインタフェースと結合し、オブジェクトプロセスの実行ファイルを作る。

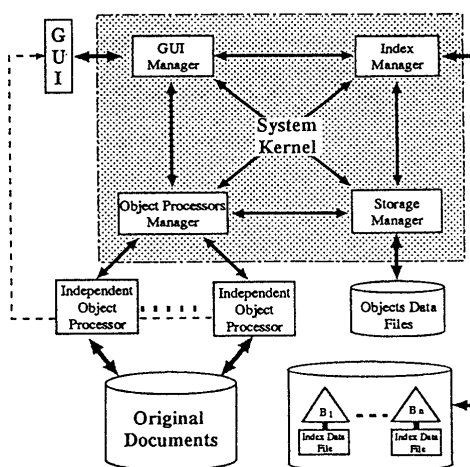


図2: システムの仕組み

4 まとめ

本論文、MMDDBにとって、データベースの構築の動的な過程に対して、従来のDBSのような管理方式を利用できないことを明確に挙げ、システムの「動的な特性」を解決するため、広義の抽象オブジェクトの概念を導入した。そして、種類多様なドキュメントを管理するDBSの構成方法を提出し、ドキュメントオブジェクトを処理実体から分離し、各種類のドキュメントに対応する独立のオブジェクトプロセスを構成することを論じた。最後に、DBSとアプリケーションの関係説明した。

参考文献

- [1] M.H.O'Docherty and C.H.Daskalakis, "Multimedia Informatin Systems : The Management and Semantic Retrieval of all Electronic Data Type", *ACM The Computer Journal*, Vol.34, No.3, January 1991 pp225-238