

# マルチメディアデータベースサーバ INADA と その上のマルチメディアデータベースモデル

4D-7

金子邦彦, 進英二, 牧之内顕文  
(九州大学工学部)

## はじめに

近年、テキスト、静止画、蓄積ビデオ、ライブビデオ、音声、音楽、立体画像などのマルチメディアが重要となっている。マルチメディアでは、遠隔地の情報源からの効果的な検索・伝送が必要なので、今後、データベース技術の適用が進んでいくものと思われる。

我々は、分散マルチメディアデータベースサーバ『INADA』の研究・開発を開始した。

### 1. 出世魚

今まで、我々は、オブジェクト指向データベースの基盤としての『出世魚』を開発し、エンジニアリングベンチマーク OO7 による評価を行ってきたところである。

出世魚は、図1のように、2層に分かれている。第1層は、分散環境で利用可能なデータベースファイル（分散共有ヒープ）を提供するストレージサーバ（WAKASHI）である。WAKASHI の機能は、並行処理制御（two-phase lock, two-phase commitment）とリカバリ（分散ロギング方式, shadow page 方式）である。

WAKASHI は、オペレーティングシステム（OS）のメモリマップドファイルの機能を利用し、データベースファイルを、アプリケーションプログラムの仮想空間へ写像する。従って、データは、アプリケーションの存在するサイト（しばしば遠隔）の実メモリとディスクに常在する。そのことで、メモリ資源を限界まで利用することを可能としている。WAKASHI は、データをページ単位で転送する。

第2層は、WAKASHI の分散共有ヒープ上に容易に C++ オブジェクトを生成するための言語と環境（INADA）である。

Multimedia Database Server INADA and its Multimedia Database Model,  
Kunihiko Kaneko, Eiji Shin, and Akifumi Maki-nouchi  
Kyushu University  
6-10-1 Hakozaki, Higashi-Ku Fukuoka, 812, Japan

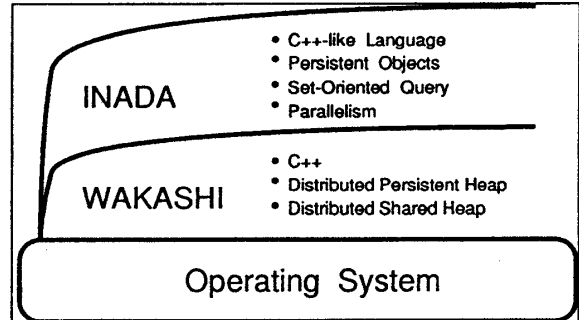


図 1: INADA の構造

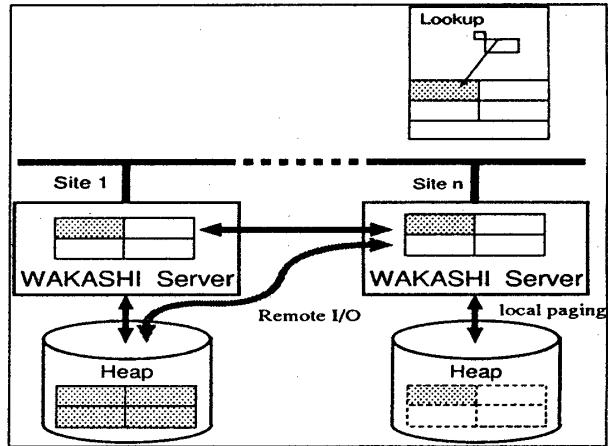


図 2: WAKASHI のメカニズム

### 2. マルチメディアデータベースサーバ

現在、WWW, Netscape, Mosaic, WebSpace, hotjava, mbone, nv, vat などを用いることで、ネットワーク上でマルチメディアを簡単に扱うことができるようになった。これら現状のシステムでは、マルチメディアをファイルとして扱っているために、多量のマルチメディア情報源の検索、編集、提示、表示の同期を行うようなマルチメディアアプリケーション（図3）を必ずしも簡単には実装できない。

我々は、出世魚をベースとして、図4のような分散マルチメディアデータベースサーバの設計・開発を開始した。図4のように、マルチメディア

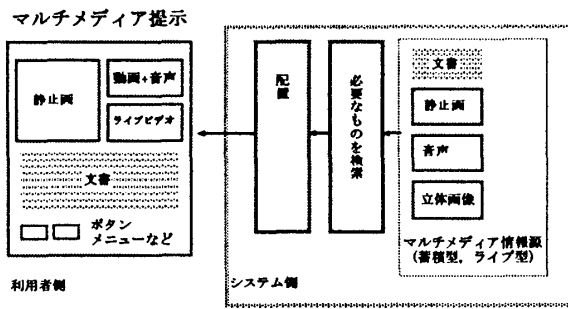


図 3: マルチメディアシステム

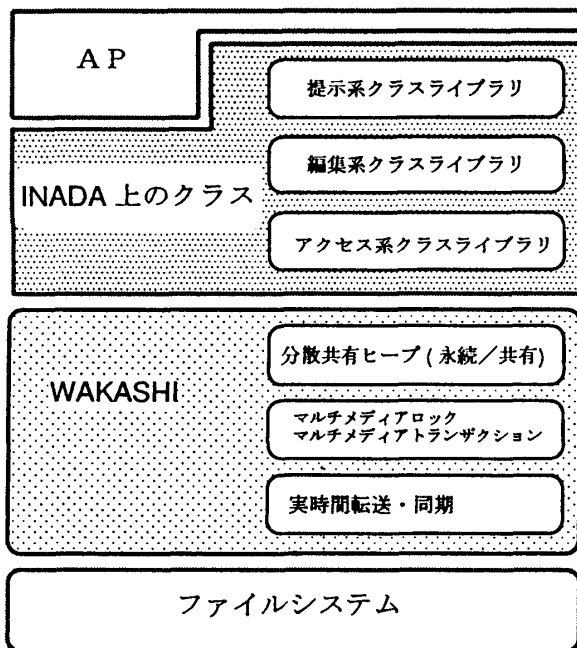


図 4: マルチメディアデータベースシステムの構想

データはファイルとして格納されるが、アプリケーション (AP) とファイルの間には、マルチメディア用に改良された出世魚がある。現在、出世魚は C++ のクラスとして実装されているので、利用者が、メディアの型に応じたクラスを実装し、出世魚と統合することは容易である。

以下に、マルチメディアデータベースサーバ構築上の課題を報告する。

### 2.1 マルチメディア・ビデオモデル

- ビデオデータベースの構造  
ビデオを”カット”、”セグメント”のような要素に分割し、格納する形式を考えている。そのことで、出世魚の問題でもあったクライアント側の仮想空間の枯渇の解決を目指す。
- マルチメディア・ビデオモデル  
単なるビデオではなく、モデルを持ったビデ

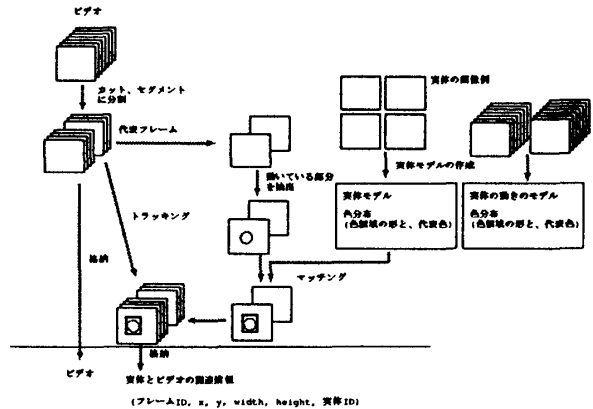


図 5: マルチメディア・ビデオシステムの構想

オを考え、より意味論的検索・編集が可能とする。それには、従来の画像認識アルゴリズムを統合 (図 5) し、認識結果とビデオとの関連もデータベース化せねばならない。

### 2.2 マルチメディア用ロック・トランザクション

現在の出世魚に採用している厳格な並行処理制御では、書き込みロックのかかったページを、他の利用者は読み出すことができない。従って、ライブビデオ、黒板など同じデータを複数のユーザが共有する局面 (共有メディア) では、他の人が書いている途中のデータを適宜読み出せるようにする必要がある。そこで、排他制御の意味を若干変更する必要がある。

マルチメディアデータには、(A) データサイズが非常に大きい、(B) 操作の大部分が読み出し (read only) である、(C) しかも、読み出しにおいて、ある程度情報の品質を落しても構わない場合が多い、という 3 つの特徴がある。そこで、我々は、共有メディアを可能とするために、WAKASHI の入れ子トランザクションを拡張した、マルチメディアトランザクションの導入を考えている。

従来のトランザクションの 4 条件 Atomicity, Consistency, Independency, Durability を A, C, D の 3 条件 (作業の undo, ログの作成というデータベースらしい機能は残す) に緩和する。さらに、他のクライアントが作業中のデータ (汚れたデータ) を読むことを許す。

それには、書き込み中のクライアントで、適宜、トランザクション中でロックモードの変更が必要である。読み出し中のクライアントは、汚れたデータを読んでいることを常に意識する必要がある。