

ビデオ オン デマンド システムにおける分散動画再生制御

1 S-3

坂上 秀和 羽根 秀宜 久保 信也 川崎 成人 岩崎 未知

NEC C&C 研究所

1 はじめに

計算機の性能の向上に伴ってその応用の可能性が大きく変わり、近年はビデオ / オーディオなどの時間次元を持つマルチメディアデータもが、その対象となっている。筆者らはこのような流れの中で、マルチメディアデータを計算機上で効率的に扱うためのモデル / 実装について研究を行っており、これまでに計算機上でローカルな記憶装置上に持つマルチメディアの再生 / 制御等を行なうためのオブジェクト指向モデルとして「マルチメディアにおける MVC(Media-Viewer-Context) モデル」を提案し、このモデルを C++ のクラスライブラリとして実装した *Xavier* を発表している [1][2]。しかしその後、ビデオオンデマンドシステム (以下 VOD システムと呼ぶ) などのように、サーバに蓄積されたマルチメディアデータをクライアント端末 (以下 STB もしくは Set Top Box と呼ぶ) から通信によって取得 / 再生する形態でのシステムが様々な場面で利用されるようになり、Client/Server 形態に適した新たなマルチメディアハンドリングに関するモデルが必要となってきた。本論文では、従来のモデルを Client/Server に分散された環境に対応すべく拡張した新たなモデル (CS/MVC モデル) について述べる。

2 Media-Viewer-Context モデル

従来の MVC(Media-Viewer-Context) モデルは、計算機上でのマルチメディアの再生を司る機能について、メディアの属性に関する情報 (Media)・再生時の動的な情報 (Context)・表示 (Viewer) に関する情報に分解し、個々の機能を実現するためのモジュールをオブジェクト指向的に分解 / 再構築することにより、様々なメディアに対する操作インタフェースの統一、新たなメディアに対する拡張性、などを実現するものである。図 1 に MVC モデルの構成を示す。

Media メディアデータの属性などの情報を管理する。メディアの種類 (動画 / 音声、圧縮方式) や、サイズ (長さ / 大きさ)、フレームレート (サンプリングレ-

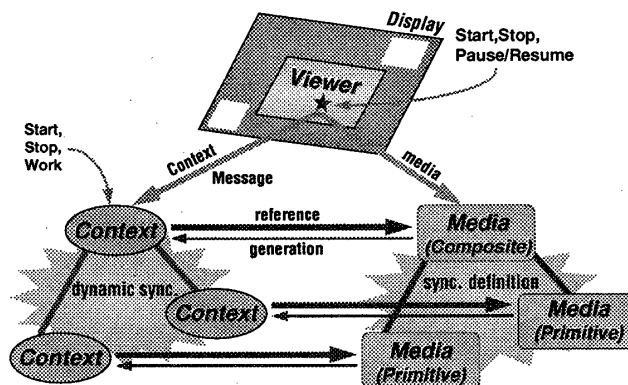


図 1: MVC モデルの構成

ト)などを管理している。

Context Context オブジェクトは、再生時の Status (play/stop)、位置、方向、速度、特殊再生の capability などの情報を管理する。また圧縮データのデコードなどのためのハードウェアデバイスの駆動も行う。

Viewer Context オブジェクトの再生状態で示されるメディアを、画面やスピーカなどに結びつけるためのオブジェクトである。画面上の表示位置と大きさ、可視性 (visibility)、オーバーレイボードを用いる場合のクロマキー合成用の色などを管理する。

MVC モデルでは上記の 3 タイプのオブジェクトに機能を分解することを特徴とし、メディア毎に異なる操作を Media や Context などに抽象化することにより、API の共通化ができ、その結果として再生制御のための Controller 部品の提供なども可能となる。また Media と Context を分離し、かつ Context を隠蔽することにより、(単純な再生表示などを行なう場合には) プログラマは再生時の動的な情報に一切煩わされることなく、プログラミングできる。さらに、デバイス依存部分などの情報が Context に集中しているため、新たなメディアや CODEC に対応するためにはそのデバイスに対応した Context オブジェクトを追加するのみで (他のオブジェクトには影響を与えることなく) 拡張できる。

3 Client/Server 拡張 MVC モデル

MVC モデルを、VOD システムなどの Client/Server 型のメディア供給形態を持つシステムに適応させるために、各オブジェクトのクライアント / サーバ上への配置を検討し、モデルの再構成および拡張を行なっ

Distributed Multimedia Processing for Video On Demand Systems

Hidekazu Sakagami, Hidetaka Hane, Nobuya Kubo, Shigehito Kawasaki, Michi Iwasaki

C&C Research Laboratories, NEC Corporation

た。ここでは、Client/Server 型のシステムに MVC モデルをマッピングするための一例として作成した新たな「CS/MVC モデル」について述べる。図 2 に、CS/MVC モデルにおける MVC 各オブジェクトの配置位置を示す。なお図 2 では、クライアント端末を Set-TopBox と表記している。

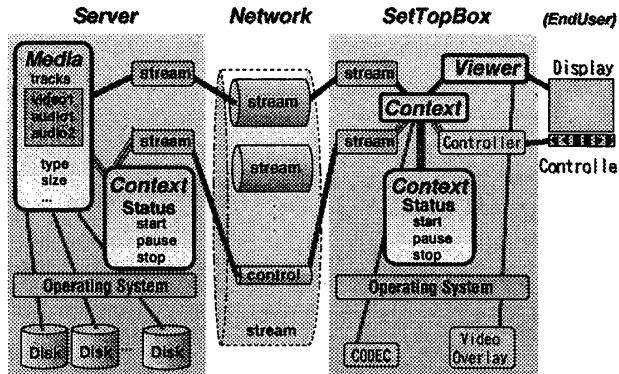


図 2: CS/MVC モデルに於ける MVC の配置

(CS)Media Media オブジェクトはデータの属性情報を扱う部分であり、この情報はサーバの Disk に格納されているメディアデータ情報のみに依存する。したがって、CS/MVC モデルにおいて、Media オブジェクトはサーバ上に配置する。

(CS)Context Context オブジェクトは再生時の動的な情報を扱うオブジェクトであり、情報の種類に応じて Client/Server の両端で多重に保持したり、一方のみで保持する必要がある。

例えば、STB 端末上でユーザからの制御コマンドに即座に対応して「停止」するためには、(Network の delay の影響などを考慮すれば)STB 上に再生状態 (Status) 情報を保持した方が有利である。また同一のメディア形式 (ex. MPEG) のデータに対して STB 上では異なる実装の CODEC (ex. soft or hard) によってデコードを行なうこともあるため、STB のデバイスに依存する情報は STB 上で保持した方が有利である。一方サーバ側でも、一時停止時の「現在位置」情報を持っていないと、再生再開時の位置が STB 側と異なってしまう場合がある。CODEC の種類による特殊再生の capability の制約などは、エンコーダに依存する場合はサーバ側に、デコーダに依存する場合は STB 側にあった方がよい。このように Context オブジェクトに関する情報は、その性質に応じて配置する必要がある。

(CS)Viewer Viewer オブジェクトに関する情報は、オーバレイボードなど STB 上のデバイスに依存する情報や、STB 上で動くアプリケーション内で使われる位置情報などであり、STB 上に配した方が有利である。

これらの 3 種類のオブジェクトの個々の機能は、基

Context の機能 / 情報	オブジェクト配置位置
再生方向、速度、現在位置	Client/Server
再生 status (pause/start,...)	Client/Server
再生 capability	Client/Server
CODEC 情報	Client

本的に (local)MVC モデルにおける各オブジェクトの機能に一致する。ただし本モデルでは Client/Server 間にまたがるオブジェクト間の通信機能が必要となる場合に、その通信量が最小限になるべく各 M・V・C オブジェクトを配置する構成となっている。

CS/MVC モデルを実装することにより、(local な) MVC モデルの特徴を生かしたまま Client/Server 型システムでの動画再生制御を実現することが出来る。これにより VOD システム上の様々なサービスを提供する Information Provider にとっても、STB 上のアプリケーションプログラムの中でサーバからのメディアを利用することが容易となる。

4 実装

本論文で述べた CS/MVC モデルを実装することにより、モデルの有効性を確認した。サーバとして UNIX WorkStation (NEC EWS4800)、STB に NEC PC98 を用いて Windows 3.1 上でアプリケーションを作成した。STB 上では C++ を用いて CS/MVC モデルを実装したライブラリを DLL 形式で作成し、Multimedia ToolBook で作成したメニュープログラムの中でサーバからの動画の再生 / 制御を行なった。また、本モデルの再生制御方式は NEC の VOD システムである HYPERMS に一部適用されている。

5 おわりに

マルチメディアデータを計算機上で再生 / 制御するための MVC モデルを、Client/Server 型に拡張対応した CS/MVC モデルについて述べた。今後は、メディア再生の枠組に関しては複数の分散されたサーバからのメディアをクライアント端末に取得する際の同期関係を実現するための方式等について、さらに検討を進める予定である。

参考文献

- [1] 坂上 秀和, 濱川 礼, 厩本 純一 「マルチメディアオブジェクト構成モデルとその実現方式」 情報処理学会研究報告, 92-OS-57, 1992.
- [2] Rei Hamakawa, Hidekazu Sakagami, Jun Rekimoto. *Audio and Video Extension to Graphical User Interface Toolkits*, The Third International Workshop on Network and Operating System Support for Digital Audio and Video, 1992.