

*分散オブジェクト環境を用いたVODシステム用ミドルウェアの実現

4 R - 7

山川 直巳 小野 泰志 村松 孝治

(株) 東芝

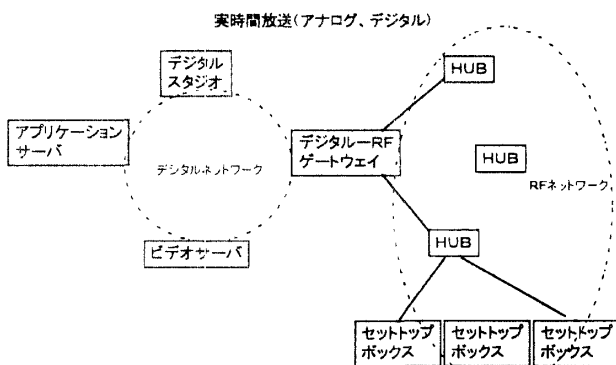
1.はじめに

デジタルCATVという大規模なネットワーク上でビデオ・オン・デマンド（VOD）システムなどを構築する試みがすでに米国などで行われており、近い将来実用化されることが予想される。本報告では、CATV上にVODを典型とするアプリケーションを構築するために必要なミドルウェアの試作について報告する。

2. CATV上のVODシステム概要

CATV上のVODシステムの構成は、次のようなものである。

図1 VODシステムの構成図



各家庭に置かれるセット・トップ・ボックス（STB）は、Radio Frequency (RF) ネットワーク、ゲートウェイ、デジタルネットワークを介してサービスを提供するサーバ群と接続されている。

STBは、安価で操作性の容易な家庭用端末である。ディスクをもたないので、OSや常駐アプリケーションはROMに置かれ、アプリケーションを必要時にダウンロードする。

RFネットワークは、スター型 Hybrid Fiber Coax(HFC)でアナログ信号による通信が行われる。VODシステムでは、従来のCATVで未使用であった450～750 Mhzの周波数帯を使用し、64 QAMというデジタル変復調技術により約1.5 Gbpsを確保している。

（従来のアナログで1チャンネルとして使用していた6MHzあたり、約30Mbps確保できる。）

ゲートウェイは、デジタル・アナログ変復調の他、IPのMPEG2TSへのカプセル化、MPEG2TSの多重化などの機能を備える。さらに、STBのIP割付、STB-サーバ間のセッション確立なども行う。サーバは、STBとセッションを確立しユーザーの要求を受け付けるアプリケーションサーバと、ビデオストリームを配信するビデオサーバで構成される。アプリケーションサーバは、STBとIPプロトコルを用いて、アプリケーションのダウンロードやイメージファイルの送信などを行っている。ビデオサーバは、ビデオストリームをMPEG/ATMプロトコルで出力する。ビデオサーバの制御はアプリケーションサーバが行う。

3 ミドルウェアの実装

3.1. ミドルウェアの位置づけ

VODなどのアプリケーションは、STB上のアプリケーションとバックエンドのアプリケーションサービスが必要である。バックエンドのサービスはアプリケーションサーバ上で動作する。アプリケーションは、適当なタイミングでSTBにダウンロードされ、サービスと協調してビデオ再生などを実行する。

ミドルウェアとして必要なことは、こういったクライアントサーバシステムが容易に構築できる部品（オブジェクト）を提供することである。機能的には、STB上でアプリケーションが起動でき、それが通常のコンピュータに準じた形でファイルアクセスでき、さらにビデオ再生できることが要求される。

実装したミドルウェアは、STBからの要求を処理するサーバとなっており、アプリケーション開発者がサービスを構築する必要がない。またSTB上に提供したサービス要求のためのライブラリのインターフェースは、分散オブジェクト環境をあまり意識することなく使用できるようになっている。実運用システムでの使用を考えた場合、アプリケーション開発者への負荷が少ないことが必要と考えるからである。

* Implementation of VOD Middleware on Distributed Object- Oriented Environment
Naomi YAMAKAWA, Yasushi ONO, Koji MURAMATSU (TOSHIBA corporation.)

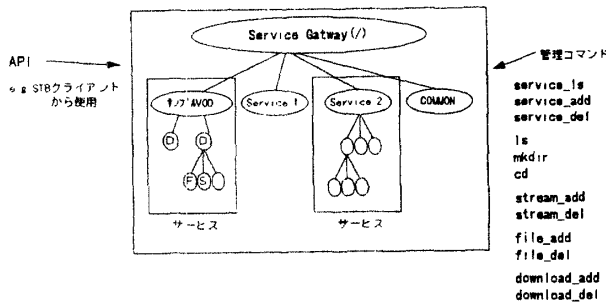
3.2.オブジェクト構成

アプリケーションサービス(バックエンド)の機能単位であるオブジェクトとして、CORBA 準拠の分散オブジェクト環境上に DSM-CC U-U の core オブジェクトに近い形で実装した。実装した疑似 DSM-CC U-U オブジェクトとその機能は次のようなものである。

- ・ Directory: オブジェクトの名前空間を作成する
 - ・ ServiceGateway: セッション接続・切断する
 - ・ Stream: MPEGストリームの制御に使用する
 - ・ File: バイト列データアクセスに使用する
 - ・ Download: プログラムをダウンロードする
- それ以外のものとしては、
- ・ Content: ビデオデータ,ファイルなどコンテンツ自身の管理を実装した

一般には、アプリケーションごとにこれらのオブジェクトを使用しサービスを構築するのであるが、コマンド操作だけで必要なディレクトリ構造を持つサービスを構築できるようにした。アプリケーション開発者がすることは、サービス側のディレクトリを作成すること、STB 上のアプリケーションを作成すること、そのアプリケーションを決められたディレクトリ登録することである。ただし、コマンドでサービスを作成するようにするため、サービスのディレクトリ構造を単純なものに制限した。

図2 サービスの階層構造

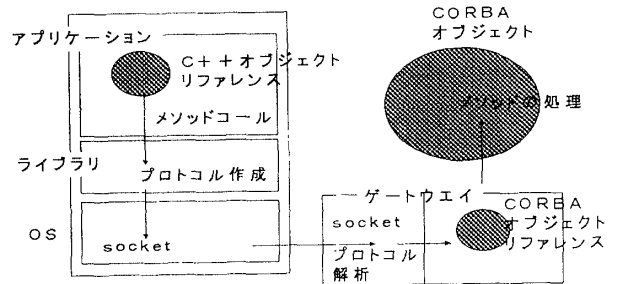


各アプリケーションサービスのディレクトリ構造は、具体的には次のようなものである。まず ServiceGateway (/) の直下に、アプリケーションサービスごとのルートディレクトリがすべて配置される。このルートディレクトリからサービス毎に自由にディレクトリを作成でき、Stream,Fileなどのオブジェクトを配置できる。ディレクトリ構造は、Directoryオブジェクトを用いて構成されている。アプリケーションサービスごとのルートの選択(open())時に、あらかじめ登録されているSTB上のアプリケーションがダウンロードされるようになっている。なお、ビデオデータなどのコンテンツの登録はこれらの構造とは無関係に登録でき、Contentオブジェクトで管理している。コマンドによりこのオブジェクトとディレクトリリーフ(Stream,File,Download)を関係づけるようになっている。

3.3.STBインターフェース

前節で述べたとおり、サーバ側は分散オブジェクト環境上に実装した。しかし、STB上には分散オブジェクトへのインターフェースを実現できなかったため、STBからの要求を分散オブジェクトの要求に変換するゲートウェイを実装した。

図3アプリケーションとCORBAオブジェクトの関係



アプリケーションは、STB上のライブラリを使用して、サーバ側にあるオブジェクトと接続する。サーバの分散オブジェクトと対応するC++オブジェクトを生成し、そのC++オブジェクト呼び出せば、ゲートウェイへの要求が生成され、CORBAの要求に変換される。

STB側に実装したインターフェースは、オブジェクト生成などCORBA独自のメソッドをできるだけ含まないようにした。たとえばStream,Fileなどのオブジェクト選択時に、サーバ側に各STB毎に別々のオブジェクトが生成され、それと対応したオブジェクトへののが返される。伝統的なソフト開発者にCORBAのインターフェースを使ってもらうことは現段階では無理であろう。

3.4.VODアプリケーション

上で説明したミドルウェアを用い、階層メニューをもつVODアプリケーションを開発した。アプリケーション開発は、サーバ側の機能拡張はできないが、CORBAなしの従来と同等の工数で開発できた。

4.おわりに

実運用システムへの使用を見通して、疑似 DSM-CC U-Uオブジェクトを中心としたミドルウェアを実装した。実装方式にはゲートウェイの採用など妥協もある。またインターネットの勢いで、DSM-CC U-Uが実運用で採用される可能性が少なくなっている。しかし、より簡単にアプリケーションを開発できるこのようなミドルウェアが必要とされている。

註)

DSM-CC:ISO/IEC 13818-6 Digital Media Storage Media Command & Control
CORBA: OMG Common Object Request Broker Architecture