

AV機器を制御するデバイス・コントロール・サーバ

1 B-4

十河 美存 谷口 秀樹 前川 肇
松下電器産業(株) オーディオ・ビデオ研究所

1. はじめに

プレゼンテーションシステムや、デスクトップビデオ編集システムなど、マルチメディア、アプリケーションでは、AV機器を制御して、動画や静止画を取り込んだり、スクリーンに動画を表示したり、編集を行なう必要がある。これらのアプリケーションは、AV機器の制御ドライバを自前で用意したり、AV機器制御ネットワークを用いたりして、必要な機能を個別に実現している。

本来、このようなAV機器を制御するドライバは、プラットフォームが提供するべきであり、実際、Macintosh OSや、VEX(Video Extension to X)などで、これらのデバイスコントロールAPIを定義していくアプローチも存在する。しかし、このようなデバイス制御APIでは、複数のアプリケーションからデバイスを共有したり、リモートのマシンからデバイスを利用したり、デバイスを正確に制御することなど、それらが提供する機能は十分ではない。

筆者らは、今回、複数のプロセスがAV機器を競合して制御するモデルに基づいて、アプリケーションを最上位層とし、AV機器を最下位層とする階層構造のシステムおよびAPIを設計し、試作したので報告する。

2. デバイス・コントロール・サーバ

今回試作、評価中のAV機器制御システムを、デバイス・コントロール・サーバ(DCS)と呼んでおり、その特長は以下の通りである。

2.1 AV機器のコントロール

AV機器を制御する場合、アプリケーションは、デバイスを正確に制御するために下記のふたつのタイミングで動作の確認を行ないたい場合がある。

a) 機器に要求した動作が開始された、あるいは、拒否された

b) 機器に要求した動作が完了した、あるいは、なんらかの理由で完了しなかった

例えば、ビデオデッキを制御して、タイムコード00:10:30:00の地点までテープを進める動作を要求した場合、動作が開始した時点でアプリケーションは、別のビデオデッキを制御するかもしれないし、ユーザのキャンセル要求を受け付けるかもしれない。また、テープが所定の位置まで進んだタイミングで、ユーザに表示を行なうかもしれない。a)のタイミングは関数の戻り値を、b)のタイミングは、非同期のシグナルハンドリングのメカニズムを用いた。

2.2 抽象化

アプリケーションは、AV機器の固有の名前よりも機能に興味がある。ビデオデッキならばメーカーは問わない場合もあるし、動画再生デバイスならば、ビデオデッキでもレーザーディスクでも構わない場合もある。AV機器を直接マシンと接続しているユーザもいれば、AV機器制御ネットワークを介して接続しているユーザもいる。

このように、AV機器の種類や制御の接続状態、台数の違いによる多様な機器構成に対しても、アプリケーションは動作しなければならない。

DCSは、制御下にあるAV機器の種類や制御の接続状態、台数に関する情報をアプリケーションに提供する。アプリケーションは、その情報をもとに、制御するAV機器を選ぶことができる。各AV機器は、ユニークなIDで管理される。

2.3 機器の共有と競合

AV機器は、価格や固定的に配置される使用形態から見て、複数のユーザに共有されることが考えられる。単一のユーザであっても、複数のアプリケーションが同一のAV機器を制御する場合もある。複数のプロセスがひとつのAV機器を制御しようとして競合するとき、この競合を調停してAV機器の共有を実現する仕組みが必要となる。

DCSでは、クライアント・サーバ・モデルを採用し、サーバがプロセス間の競合を調停する。サーバは、ふたつのプロセスが同一のAV機器を利用しようとして競合するとき、

a) 先に利用しているプロセスの要求を続行し、続行中は、他のプロセスに利用できない旨を知らせる。

b) 先に利用しているプロセスの要求を中断し、後のプロセスの要求を実行する。先に利用していたプロセスには、中断された旨を伝える。

のどちらかの方法により、調停を行なう。a), b)どちらの方法を取るかは、各クライアントに与えられたプライオリティを、サーバが比較して決定する。今回のサーバは、各クライアントのプライオリティがいずれも同一レベルであるとして、b)の方法で調停を行なっている。

2.4 階層構造

DCSライブラリをリンクしたアプリケーション・クライアントは、共通のAPIを用いて、AV機器の制御をサーバにリクエストする。サーバは、要求されたAV機器のリクエストを調停して、そのリクエスト

が実行可能ならば、適当なデバイス・ドライバを選択して、AV機器の制御を行なう。リクエストの開始時には、クライアントに受付番号が返され、完了時には、受付番号を伴った完了通知が返される。サーバとクライアント間のプロトコルは、TCP/IP 上のパケットとして規定しており、このパケットを理解するクライアントは、DCS制御下のAV機器を利用することが可能である。

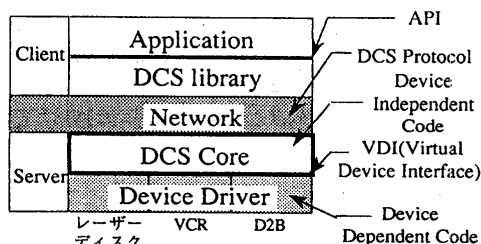


図1 DCSの階層構造

3. 実装および評価

3.1 評価システムの構成

TCP/IPで接続されたEWS(Solbourne S5/600, Sparc station 2, HP9000/835)上で、DCSと、クライアント・ライブラリをリンクしたアプリケーションを走らせて、システム及びAPIを評価中である。なお、DCS及びアプリケーションはC言語で記述した。

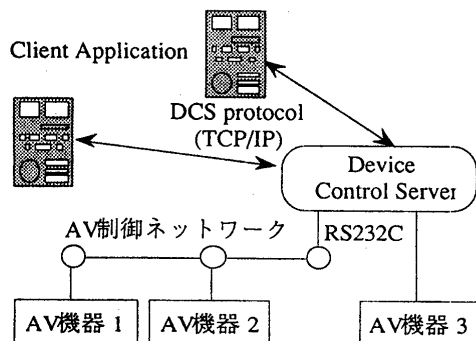


図2 DCSを利用したシステム例

3.2 評価アプリケーション

ビデオのカット編集を行なう、動画編集ツールを試作した。カット切り出し用のツールと、カット編集用のツールが、AV機器を共有して動作する。

カット切り出し用のツールは、1個のツールが1台のAV機器に対応しており、AV機器を操作してカットのイン点とアウト点を決定し、カットを表すアイコンを作る。カット切り出し用のツールから作られたアイコンは、カット編集用のツールのアイコン領域に並べられる。ユーザは、カット編集用のツールの編集領域に、アイコンを並べる。編集領域では、アイコンが並んだ順序にカットがつながるという規則によって、編集順序が決定される。カット編集用のツールでは、AV機器を制御して、カットの内容を

確認するためのカットの再生や、編集順序に従った編集画像の再生を行なうことができる。

カット切り出し用のツールや、カット編集用のツールは、それぞれが独立したアプリケーションである。

3.3 評価

クライアントとして動作している各ツールが、AV機器を競合する状態においては、リクエストを中断されたクライアントに、サーバから非同期の通知が行なわれるので、ツールがAV機器の動作完了を待ち続けてハングアップするような事態は生じない。通知のタイミングは、time criticalなものではなく、TCP/IP等の伝送遅延の影響を受けているが、クライアントの次の動作へのトリガとして使用するのであれば、実用上問題はない。フレーム単位での協調動作をおこないたいのならば、クライアント側でなく、サーバ側で処理を行なう必要がある。

アプリケーション開発においては、機器依存のコーディングをせずにソースを記述することができる。アプリケーション開発者は、クライアント・ライブラリとして提供されたAPIを用いればよく、AV機器がどのようなコマンドを解釈しているかを理解する必要はない。サーバの対応機器が増えたとしても、アプリケーションに手を加える必要はないので、アプリケーション資産は継承される。AV機器共有のメカニズムは、サーバ側で処理されるので、アプリケーション側のソースの負担が軽くなる。

このシステムで新しい制御体系のAV機器に対応するためには、新しいAV機器に対応したデバイス・ドライバをただひとつのサーバに組み込むだけでよく、クライアントの改造は要しない。これは機器依存の部分をサーバが吸収しているからである。

4. おわりに

今回は、AV機器の共有を実現するDCSについて報告した。今後の課題としては、デバイス利用の優先権に関する仕組みの実現、AV機器の接続状態の把握と情報提供の仕組みの実現がある。また、AV機器の共有を実現するAPIの標準化が必要となるだろう。

現在、本システムは、機種依存しない動画編集システムのプラットフォームとして試用中である。今後は、オーディオ機器やカメラ等、各種デバイスに対する制御への拡張を行なっていく予定である。

参考文献

VEX Devices and Controls Version 1.3

Todd Brunhoff

Tektronix, Inc Tektronix Research Laboratories

D2Bシステム HC91-40

岩本幸治他(松下電器)

(社)電子情報通信学会