

家庭内 AV ネットワーク技術「HAVi」の概要

Outline of HAVi

樋口 正生, 森岡 隆一郎, 稲垣 勝利, 戸崎 明宏

Masao Higuchi, Ryuichiro Morioka, Katsutoshi Inagaki, Akihiro Tozaki

要旨 HAViは家庭内においてネットワーク接続されたデジタルAV機器をメーカーや機種にとらわれることなく相互に運用するための仕様である。HAVi アーキテクチャはホームネットワーク上の分散型アプリケーションを開発するためのAPIのセットとサービスを提供する。本稿ではHAVi仕様の概要として機器制御モデルとソフトウェアアーキテクチャについての紹介を行う。

Summary HAVi(Home Audio Video Interoperability) facilitates multivendor interoperability between consumer electronics devices on a home network. The HAVi architecture consists of a set of application programming interfaces and services for the development of distributed applications on a home network. The authors will outline the HAVi specification, and introduce about its device control model and specific software elements of the HAVi architecture.

キーワード : HAVi , AV 機器 , ホームネットワーク API , 分散型アプリケーション

1. まえがき

近年,Set Top BoxやD-VHSなどAV機器のデジタル化が進み,そのデジタルインターフェースとしてIEEE1394⁽³⁾をサポートする機器も普及しつつある。またインターネットをとりまく環境の急激な進歩やBSデジタル放送など情報インフラが整備されてきたことにより,ネットワーク経由で多大なデジタルコンテンツが家庭の中に入ってくるようになってきた。これらのコンテンツを家庭内の様々なシチュエーションで効果的に扱えるようにするためには,外部ネットワークとデジタルストレージデバイス,あるいはディスプレイデバイスとのデジタルネットワーク接続が不可欠である。このような家庭内のデジタルAV機器を賢く,柔軟に接続するためにHAVi(Home

Audio/Video Interoperability)⁽¹⁾が開発された。HAViは家庭内におけるデジタルAV機器をメーカーや機種にとらわれることなく相互に接続するための基本仕様である。HAViの検討は1996年にソニーとフィリップスによって開始され,その後,家電メーカーである東芝,松下,日立,シャープ,トムソン,グルンディッヒが加わり,合計8社で仕様の作成が進められ,2000年1月にバージョン1.0の仕様書が発行された。またHAVi仕様の普及と発展のためにHAVi推進協会⁽²⁾が設立され,当社もメンバー会社としてテクニカルワーキンググループの活動に参加している。

本稿ではHAViで提供される機能を中心に概要を述べ,その後AV機器制御モデルとHAViソフトウェア構造,さらに提供されるサービスについて解説する。

2. HAVi の特徴

HAVi の最大の目的はホームネットワークで接続されたデジタルAV 機器を分散コンピューティングプラットフォームとみなして、この上で動作する分散アプリケーションがそれぞれ協調してそのタスクを実行できるような仕組みを提供すると共に、異なったメーカーの製品の相互接続・相互運用を確保することにある。HAVi アーキテクチャはこのためのサービスを提供するソフトウェアを定義したミドルウェア仕様であると共に、分散型アプリケーションを開発するためのAPIのセットを提供している。HAVi の特徴について以下に述べる。

2.1 プラグアンドプレイのサポート

機器がネットワークに接続された際にコントローラが機器を認識し、必要なソフトウェアを自動インストールしてGUIなどをユーザーに提供する。HAVi 準拠機器であれば異なるメーカーの機器でも制御できる。

2.2 従来製品のサポート

家庭内のAV 機器がすぐにHAVi 対応のネットワーク機器になるわけではないので、HAVi に対応していない従来製品も制限付きではあるがHAVi コントローラから制御できるような仕組みを提供している。

2.3 将来発売される製品にも対応

HAVi コントローラは新たな機器がネットワークに接続された際に、機器を制御するための制御情報を機器から取り出して機器を認識し、さらに制御プログラムをインストールして実行するという仕様になっているので、この制御情報により未知の製品にも対応できる。

2.4 リソースマネジメント

AV 機器をネットワークで接続すると複数のユーザーまたはアプリケーションが同時に機器にアクセスしてお互いの動作を妨げるということが起こり得る。これを解決するために排他制御を行い、さらにアプリケーション間で調停を行う仕組みを提供する。またホームネットワークに接続された複数の機器を組合わせた予約動作を可能にする。

。HAVi のコントローラは各機器の予約状態など必要なリソースの状態を管理するため、予約時に予定時刻における動作が可能かどうかを確認することができる。また予約している機器がネットワークから外れたなど、予約動作が実行できない状態になった場合にはアプリケーションに対し通知を行う。

2.5 Java のサポート

HAVi はJava(JDK 1.1 準拠)をサポートしているため、メーカーやプラットフォームに依存しないアプリケーションを開発することが可能である。HAVi Java API というパッケージが提供されている。

3. HAVi の AV 機器制御モデル

3.1 Device Class

HAVi ではホームネットワークに接続するAV機器を図1に示す4つのカテゴリーに分類する。FAV, IAV, BAVがHAVi 準拠機器となりそれぞれのレベルに応じて要求事項が決められている。またHAVi のコントローラになれる機器はFAVとIAV、コントロールされるのみの機器はBAVとLAVとして明確に区別されている。

HAVi 準拠機器にはSDD(Self Describing Device)データと呼ばれるHAVi 機器としてのプロフィールや制御情報を含んだデータがConfiguration ROM^{(3),(4)}中に置かれる。SDDデータに含まれる情報は下記の通りである。

Full AV Device (FAV) HAViの全てのサービスを提供できる。またJavaの実行環境を持ち、HAViアプリケーションやDCMをアップロードすることができる。
Intermediate AV Device (IAV) HAViのメッセージ通信を行うことができ、基本的なHAViのサービスを提供できる。またBAV機器の一部とLAV機器をホストすることができる。これらを利用したHAViアプリケーションを実行することができる。
Base AV Device (BAV) IEEE1394とIEC61883をサポートしており、Configuration ROM中にHAViで定義されるHAVi SDDデータを含む機器。FAVの全てとIAVの一部にホストされる。IAVにホストされる場合はLAVモードと呼び、メーカー独自の機能は実現できない。
Legacy AV Device (LAV) HAVi SDDデータを持たない機器。上記の3つの条件に当てはまらない機器は全てLAVである。必ずしもIEEE1394をサポートしていなければならないというわけではないが、サポートしない場合は独自の方法でコントローラにホストされることとなるためコントロールできる機器は限られる。

図1 HAVi のデバイスクラス

(1)HAVi Device Profile

デバイスクラスやベンダー情報 ,HAVi機器としての能力を記述

(2)HAVi User Preferred Name

HAVi 準拠機器ではユーザーが自由に名前を付けることができる。これを書きこむためのエリア

(3)HAVi DCM ,HAVi DCM Profile ,HAVi DCM Reference

DCMと呼ばれる制御プログラムとその属性情報 , URL 情報などを記述

(4)HAVi Device Icon Bitmap

HAVi 機器を表すアイコンのデータ

3.2 DeviceとFunctional Component

HAVi では機器全体(Device)と , 機器に含まれる機能要素(Functional Component)を区別して扱う。例えば D-VHS という機器には Tuner と Tape Recorder という機能要素が含まれる。電源の ON/OFF は機器に対する制御であるが , Tuner の選局やテープの再生はそれぞれの機能要素に対する制御となる。HAVi における制御とは , 機器単位というよりは主に機能要素単位の制御が中心である。コントローラはネットワーク接続された AV 機器の中から必要な機能要素だけを組み合わせることができる。機器の中に複数の機能要素が存在する場合 , 一つの機能要素をあるアプリケーションが使い , 別の機能要素を別のアプリケーションが使うといったように一つの機器を複数のアプリケーションで共有することもできる。

HAVi では機器固有の制御方法をカプセル化し , 標準化されたAV機器のAPIを利用してアプリケーションを開発することができるように上述した機器と機能要素をモデル化したソフトウェアモジュールを定義している。機器全体をモデル化したものを DCM (Device Control Module) 機能要素をモデル化したものをFCM(Functional Component Module)と呼ぶ。機器に対して1つのDCMが存在し , DCMにはその機器の機能要素の数に応じて複数個のFCMが含まれる。

DCM は機器をホストする HAVi コントローラ上にインストールされて動作するソフトウェアオブ

ジェクトである。HAViのアプリケーションはDCMあるいはこれに含まれるFCMのAPIを呼び出すことにより機器およびこれに含まれる機能要素の制御を行うことができる。DCMをホストするコントローラとアプリケーションが動作するコントローラは同じ機器である必要はなく , DCM(FCM)はホームネットワーク上のリソースとして全てのアプリケーションから利用できる。

3.3 DCM

DCM には機器電源の制御や機器内部接続の設定など機器全体に関連した基本的な制御を行うためのAPIと機器の状態取得のためのAPIが定義されている。DCMは以下の役割を担う。

(1)機器内部接続の管理

機器内部のプラグ間の接続の確立 , 切断と接続に関する情報提供を行う。

(2)機器の予約状況の管理

機器とこれに含まれる機能要素の予約状態を管理し , 予約の命令を受けた際にはオーバーラップがないかどうかのチェックを行う。

(3)ユーザーインターフェースのサポート

HAVi コントローラに提供するためのユーザーインターフェースデータを提供する。またコントローラ上で実行できる機器制御のためのアプリケーション自体の提供も行う。

3.4 FCM

FCMの種類としてはTuner ,VCR ,Clock ,Camera , AV Disc , Amplifier , Display , AV Display , Modem ,Web Proxyが定義されているが , メーカー独自のFCMを定義することも可能である。これらのFCMにはその機能に応じて標準的な制御を行うためのAPIと状態取得のためのAPIが定義されている。また各種の状態変化が発生した際にはアプリケーションなどに通知(Notification)を送る機能も備えている。FCMは以下の役割を担う。

(1)プラグの管理

ストリームを流すためのプラグを管理し , ストリーム種別に関する情報を提供する。

(2)機器の利用状況の管理

FCMは複数のアプリケーション(ユーザー)

からのアクセスに対して排他制御を行う仕組みを提供する。また利用状況に関する情報提供を行う。

3.5 制御モデル

AV 機器を操作する際にはコントローラ上のアプリケーションがターゲットの機器に対して何らかの方法で制御命令(コマンド)を送信することになるが、この場合双方でコマンドの内容や手順を取り決めておく必要がある。1394TA⁽⁵⁾ではAV/Cコマンド⁽⁶⁾と呼ばれるIEEE1394対応AV機器を制御するためのコマンドセットを規定しており、さまざまな種類のAV機器に対して標準的なコマンドが用意されている。異なるメーカーの機器を制御するにはこのような標準的なコマンドを使うことが一つの方法である。しかしこのコマンドに対応していない機器や、メーカー独自の機能を機器に組み入れたい場合には標準コマンドを使うことができないため、異なるメーカーのコントローラでは制御できない。この問題を解決するのがHAViのDCMとFCMである。HAViのアプリケーションは機器を制御するためのコマンドを直接機器に対して送信するのではなく、DCM/FCMに用意され

たAPIを呼び出す。アプリケーションが機器を制御するための唯一のアクセスポイントがDCM/FCMとなる。DCM/FCMは呼び出されたAPIを解釈し、機器に対応した固有のコマンドに変換して独自の方法で機器にコマンドを送信するという仕組みになっている(図2)。

DCM/FCMにはそれぞれ機器を制御するための標準的なAPIが用意されているので、アプリケーション開発者は機器固有のコマンドを知らなくてもHAViで定義されたAPIを使ってアプリケーションを開発することができる。これによりメーカー間のインターオペラビリティが実現できる。

またDCM/FCMはターゲット機器のメーカーにより作成されるものであるためメーカー独自の機能を実現するAPIをDCM/FCMに追加することもできる。独自の機能は通常他のメーカーのアプリケーションでは使えないが、HAViではDCM/FCMに加えてユーザーインターフェースやアプリケーション自身もターゲット機器側から提供することができる仕組みになっているため、これらをコントローラ側で利用することによりメーカー独自機能を他社のコントローラからも利用することができる。

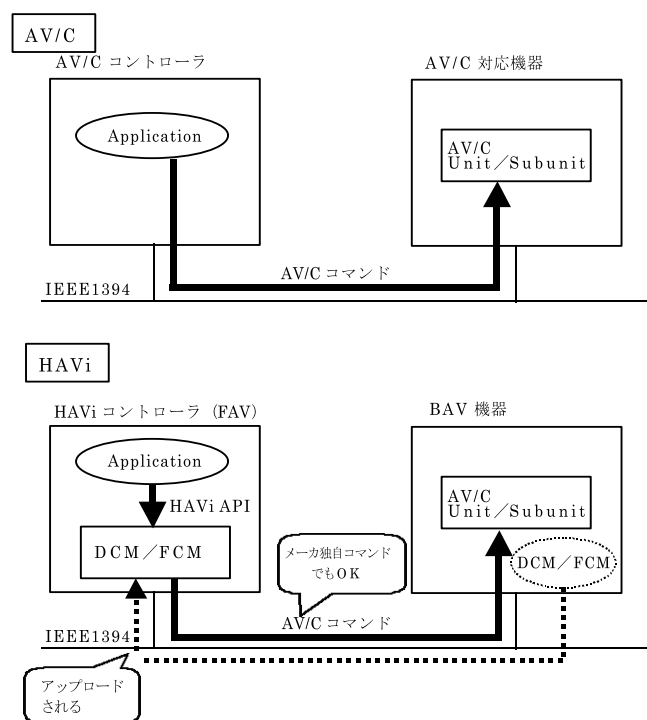


図2 AV/CとHAViの違い

3.6 DCMのローディング

BAVおよびLAV機器のDCMはDCMコードユニットと呼ばれる実行コード形式のファイルとして提供される。DCMコードユニットは後述するDCM ManagerによってHAViコントローラ上にインストールされ、DCMとFCMがソフトウェアエレメントとして起動される。DCMコードユニットがどこに存在するかによってDCMは以下の2種類に分けられる。

(1)uploadable DCM

BAV機器のSDDデータの中に格納されているDCM。あるいはメーカーのURLから取得するDCM。HAViではアップロード可能なJavaのDCMCodeUnit(jar形式)に対してのみアップロードとインストールの方式が規定されている。その他のものについてはメーカー依存である。

(2)embedded DCM

FAV・IAV機器にあらかじめ実装されているBAV・LAV機器のためのDCM。インストールの方式、実装方法はメーカー依存である。

また、FAV・IAV機器自身のためのDCMはあらかじめ機器に組み込まれており、メーカー独自の方式で起動される。

3.7 ユーザーインターフェース

HAViはアプリケーションに対してユーザーと対話するための仕組みを提供する。HAViではプラットフォームに依存しない2つのレベルのユーザーインターフェースが用意されている。

3.7.1 レベル1ユーザーインターフェース

レベル1ユーザーインターフェースはDDI(Data Driven Interaction)と呼ばれ、ディスプ

レイやリモコンなどのユーザーインターフェース(以下UIと記述)を扱うDDIコントローラとUIデータを提供するDDIターゲットが定義されている。DDIコントローラとDDIターゲット間の情報のやり取りはDDIプロトコルとしてHAViで規定されている。DDIコントローラはディスプレイ機能を持つIAV機器には必須であるがディスプレイ機能を持つFAV機器でも実装することが推奨される。DDIモデルを図3に示す。

DDIコントローラはまずDDIターゲットと通信してUIデータ(パネル、ボタン、アイコン、テキストなどのDDIエレメント)を取得し、これらをディスプレイ上に表示する。ここで表示される画面はディスプレイの能力によってはDDIターゲット側で意図した画面とは異なる場合もありうるが、最低限の要求事項は規定されている。この画面に対し、リモコン入力などユーザーからの入力があった場合には、どのDDIエレメントに対するどのようなアクションであるかという通知(例えば1番目のボタンが押されたなど)がDDIコントローラからDDIターゲット側に送られる。DDIターゲットはこれを受けて対応するDCM(FCM)のAPIを呼び出し、機器に対して操作命令を送る。機器を操作したことにより機器の状態が変化して画面の表示を変えたい場合にはこの旨の通知がDDIターゲットからDDIコントローラに送られ、対応するDDIエレメントの表示を変化させる(例えばボタンの色を変えるなど)。このような一連の動作によりユーザーが機器を操作することができる。DDIエレメント自体は(画面のデザインも含めて)DCMと共に機器側から

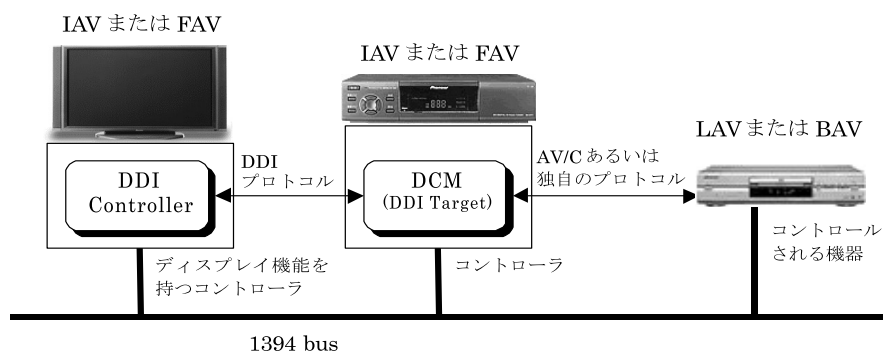


図3 DDIモデル

提供されるので、コントローラのアプリケーションはDDIプロトコルに対応することによりDCMやFCMにメーカー独自のAPIが定義されていたとしてもこれを意識することなく機器を操作できる。

3.7.2 レベル2ユーザーインターフェース

レベル2ユーザーインターフェースはJavaで記述されたアプリケーションソフトウェアで、havletと呼ばれる。HAViではJava AWTをTVフレンドリーな形(リモコンにも対応)で拡張したUIパッケージを定義しており、これを利用してhavletを開発することができる。havletはDCMと共にBAV機器に搭載されており、ディスプレイ機能を持ったFAV機器上のアプリケーションからDCMのGet Havlet Code Unit APIを呼び出すことによりインストールされて動作する。havletは機器メーカーからDCMやFCMと共に提供されるので、メーカー独自の機能を用いたアプリケーションソフトウェアを実現することができ、これを異なるメーカーのFAV機器上で動作させることができる。

4. ソフトウェアアーキテクチャと提供されるサービス

HAViのソフトウェア構成を図4に示す。この図はFAVの場合の構成例を示している。HAViアーキテクチャはIEEE1394をサポートするベンダ独自のプラットフォーム上に実装することができる。

HAViアーキテクチャを構成する各オブジェクトはソフトウェアエレメントと呼ばれ、HAViの高機能な各種サービスのためのAPIを提供する。このAPIはインターオペラビリティAPIとして機能するので、アプリケーション開発者はこれらのAPIを使ってプラットフォームや機種に依存しないアプリケーションを作成することができる。

ここではHAViのシステムサービス機能を実現するソフトウェアエレメントについて解説する。

4.1 CMM1394

CMMはCommunication Media Managerの略で、他の機器との通信プロトコルを抽象化するためのものである。現時点ではIEEE1394のみをサポートしているためCMM1394しか存在しないが、将来他のプロトコルをサポートする際にはCMMを拡張することにより対応が可能となる。

CMM1394は全てのFAVとIAV機器に実装され、他のソフトウェアエレメントに以下のIEEE1394関連のサービスを提供する。

- ・Write,Read,Lockトランザクションを実行する。
- ・ネットワーク上の機器のGUIDのリストを提示する。
- ・バスリセットを検出しイベントを発行する。
- ・他の機器からのトランザクションが発生した際に、処理が必要なソフトウェアエレメントに対して通知を行う。

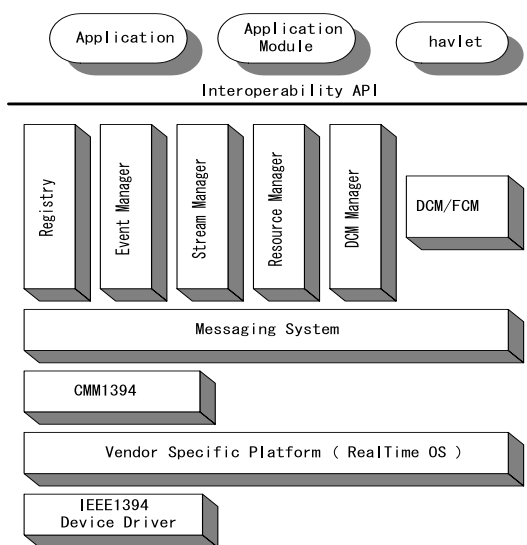


図4 HAVi ソフトウェアアーキテクチャ

4.2 Messaging System

Messaging Systemは全てのFAVとIAV機器に実装され、HAViメッセージ伝送サービス機能を提供する。HAViのソフトウェアエレメントは他のソフトウェアエレメントと通信を行う際に(APIの呼び出し、API呼び出しに対する返答、イベントやNotificationの配布など)、HAViのメッセージ伝送サービスを利用する。ソフトウェアエレメント間のメッセージ通信は同じ機器上だけにとどまらず、ネットワーク接続された他の機器との間でも行われる。

ソフトウェアエレメントは通信を行うために、まずMessaging Systemのopen関数を呼び出してコールバック関数を登録する必要がある(図5)。この際SEID(Software Element Identifier)と呼ばれるホームネットワーク上でユニークな10バイトの識別子が付与され、HAViメッセージを送信する際に送信元、送信先を示す値として使用される。SEID

によりアプリケーションはソフトウェアエレメントがどの機器上のオブジェクトであるかということを意識することなくメッセージ伝送サービスを利用することができる。

HAViメッセージの伝送モードには相手がメッセージを受け取ったかどうかを確認しないSimpleモードと確認を行うReliableモードの2種類がある(図6)。またリモートメソッド呼び出し(RMI)をサポートし、リクエスト送信のためのAPIとレスポンス送信のためのAPIを提供する(図7)。この仕組みを利用してソフトウェアエレメントは他のソフトウェアエレメントのAPIを呼び出すことができる。

またMessaging Systemはネットワーク上のソフトウェアエレメントを監視するサービスを提供しており、監視したいソフトウェアエレメントを登録しておくこと、そのソフトウェアエレメントが消滅した際に通知を受け取ることができる。

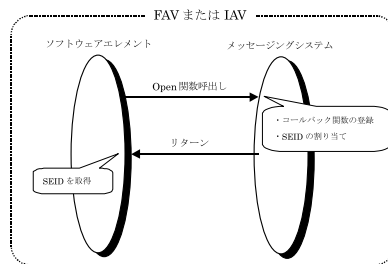


図5 Open関数の呼び出し

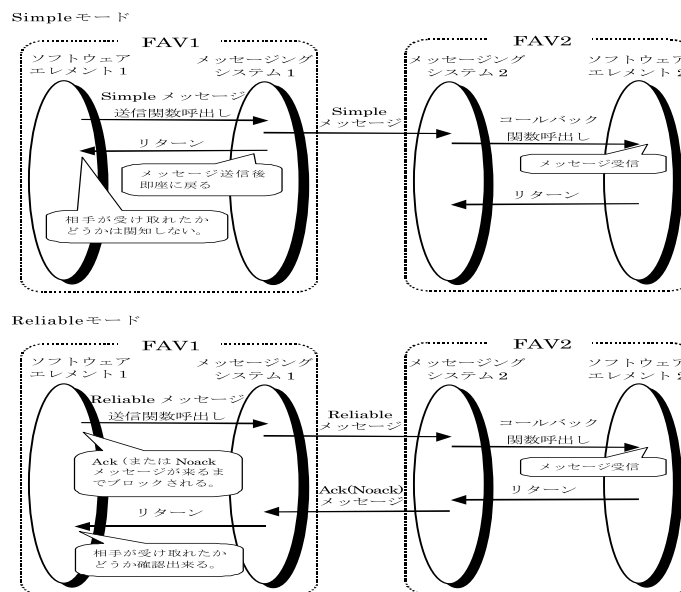


図6 メッセージ伝送モード

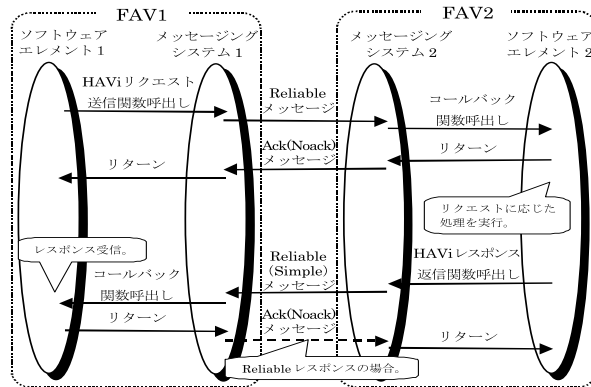


図7 リクエスト・レスポンス プロトコル

4.3 Event Manager

HAViのソフトウェアエレメントは機器やネットワークあるいはソフトウェアエレメント自身の状態が変化した時にイベントを発行し、他のソフトウェアエレメントにその状態変化を通知することができる。Event Managerは全てのFAVとIAV機器に実装され、イベントの発行とその登録などのイベント管理サービス機能を提供する。イベントの種類はSystem EventとしてHAViで定義されているものの他にメーカー独自のイベントを定義することも可能である。イベントはソフトウェアエレメントが存在する機器内(ローカル)だけに送る場合とホームネットワーク上の他のFAV・IAV機器全て(グローバル)に送る場合がある。System Eventは種類によってローカルかグローバルかが規定されているが、メーカー独自のイベントは発行する側で決めることができる。グローバルのイベントの場合はイベントを受けたEvent Managerが他のFAV・IAV機器上にある全てのEvent Managerにイベントの転送を依頼する。イベントを発行したいソフトウェアエレメントはイベントの種類と付加情報を用意してローカルのEvent ManagerのPostEvent APIを呼び出す。またソフトウェアエレメントがイベントを受けたい場合にはEvent Managerに興味のあるイベントの種類を登録しておくことにより、そのイベントが発行された際にEvent Managerから通知を受けることができる。

4.4 Registry

Registryは全てのFAVとIAV機器に実装され、ホームネットワーク上のソフトウェアエレメントを検索するためのディレクトリサービスを提供する。Registryは機器内(ローカル)にあるソフトウェアエレメントのSEIDとその属性情報が登録されるデータベースを管理する。HAViのメッセージ通信を行いたいソフトウェアエレメントはRegistryのRegisterElement APIを用いて、自分のSEIDとその属性情報(ソフトウェアエレメントの種類やベンダIDなど)をRegistryに登録する。登録されたソフトウェアエレメントはホームネットワーク上の他のソフトウェアエレメントから見えるようになり、通信(APIの呼び出しなど)が可能になる。特定のソフトウェアエレメントのSEIDを検索したい場合は、属性情報をパラメータとしてRegistryのGetElement APIを呼び出すことによりSEIDを得ることができる。またSEIDをパラメータとしてRetrieveElement APIを呼び出すことにより属性情報を取り出すこともできる。これらのAPIを受けたRegistryは自ら持つデータベース内を検索すると共に、他のFAV・IAV機器上にある全てのRegistryに対し検索依頼を出すので、ホームネットワーク全体としての検索を行うことができる。

4.5 DCM Manager

DCM ManagerはBAVまたはLAV機器のためのDCM

コードユニットのインストールとアンインストールを行うサービスを提供する。DCM ManagerはFAV機器には必須であるが、IAV機器はBAVまたはLAV機器をホストしない場合には必要ない。ホームネットワークに機器が新たに接続された時にDCMコードユニットがインストールされる手順を以下に示す。

1. DCM ManagerはまずSDDデータを解析してその機器のDevice Classを調べ、BAVまたはLAVであった場合に次のステップに移る。
2. ネットワーク上に複数のFAVまたはIAV機器がある場合HAViで規定されたDCMマネジメントプロトコルにしたがって各DCM Managerが調停を行い、この中の1つがリーダーとして選ばれる。
3. リーダーは他のDCM ManagerからDCMのインストール情報を集め、どのDCM Managerがホストととして最適なかを決定し、インストールの指令を出す。
4. 指名されたDCM Managerは指定の場所(BAV機器のSDDデータ、指定のURL、指定のembeddedDCM)からDCMコードユニットをインストールする。
5. DCM ManagerはインストールされたDCMコードユニットに対してinstall APIを呼び出す。これによりDCMとFCMがインストールされてソフトウェアエレメントとして起動する。

4.6 Stream Manager

Stream ManagerはIEC61883⁽⁷⁾プロトコルに基づいた接続管理サービスを提供する。Stream ManagerはFAV機器には必須であるが、IAV機器はこのサービスを必要とする場合にオプションとして実装される。ここではHAVi接続の特徴とStream Managerのサービスについて解説する。

4.6.1 HAVi接続の特徴

HAVi接続の特徴を以下に示す。

(1)FCM間のEnd to End接続

HAViの接続はFCMプラグに始まり、FCMプラグで終端する。また1つのソース(送出側)FCMプラグに複数のシンク(受信側)

FCMプラグが接続できる。

(2)3種類のトランスポートタイプ

Internal(機器内部の接続)、IEC61883(IEEE1394バスで接続された機器間のIEC61883に基づく接続)、Cable(IEC61883以外の機器間の接続)の3種類をサポートする。Cable接続の具体的な方法についてはメーカー依存となる。

(3)ストリームタイプを定義

コネクション毎にストリームタイプが設定される。FCMプラグは複数のストリームタイプをサポートできる。

4.6.2 Stream Managerが提供するサービス

Stream ManagerはHAVi接続の確立、破棄、状態取得など以下のサービスを提供する。

(1)コネクションの確立と破棄

FCMプラグとFCMプラグの両端を指定した接続に加え、Broadcast IN/OUT⁽⁷⁾もサポートする。また不正にコネクションが破棄されないように管理を行う。

(2)リソースの確保

アイソクロナスチャンネル⁽³⁾、帯域幅⁽³⁾、PCR⁽⁷⁾などコネクションに必要なリソースを確保する。

(3)コネクション情報の管理

自らが管理するコネクション情報に加え、他のFAV・IAV機器上のStream Managerからも情報を取得し、ホームネットワーク全体のコネクション情報を提供する。

(4)プラグの互換性チェック

コネクション確立要求を受けた際に両端のプラグのストリームタイプをチェックし、互換性のないプラグ同士の誤接続を防ぐ。

(5)ホームネットワークの状態変化への対応

バスリセットが発生した際にコネクションの復旧を行う。また機器がネットワークから外れた場合やリソースが不足してコネクションが維持できなくなった場合などにはイベントを発行し、ソフトウェアエレメントに通知を行う。

4.7 Resource Manager

Resource ManagerはResource ReservationとScheduled Action Managementと呼ばれるリソースマネジメントサービスを提供する。Resource ManagerはFAV 機器には必須であるが、IAV 機器に対しては少なくとも1つのDCMのホストになることができる場合に実装される。ここではリソースマネジメントの各機能について解説する。

4.7.1 Resource Reservation

HAViにおけるリソースとはTunerやVCRなどのFCMを示す。アプリケーションはこれらを排他的に使用する際にリソースの制御権を得る(リザーブする)必要がある。このためにはFCMを指定してResource ManagerのReserve APIを呼び出す(図8)。リザーブにはPrimaryとSecondaryの2種類がある。Primaryのリザーブはリソースに対する全ての制御権が得られるが、これは1つのソフトウェア

エレメントだけに限られる。Secondaryのリザーブは他のソフトウェアエレメントが使用していない機能のみに限定の制御権ではあるが、リソースによって決められる数までのソフトウェアエレメントに対して認められる。このように1つのリソースを複数で共有して使用することができる。

また既にリザーブされているリソースを横取りすることもできる。この場合には通常リザーブしているソフトウェアエレメントに対しResource Managerを通じて権利の放棄を要求し、承諾が得られた場合に権利を獲得することができる(図9)。

4.7.2 Scheduled Action Management

Scheduled Actionはホームネットワーク上のリソースを1つ以上使った連携動作を所定の時刻に開始・終了するといういわゆる予約動作のことである。アプリケーションは予約を行うために、開始・終了時刻とそのときに発行するコマンド、使用する

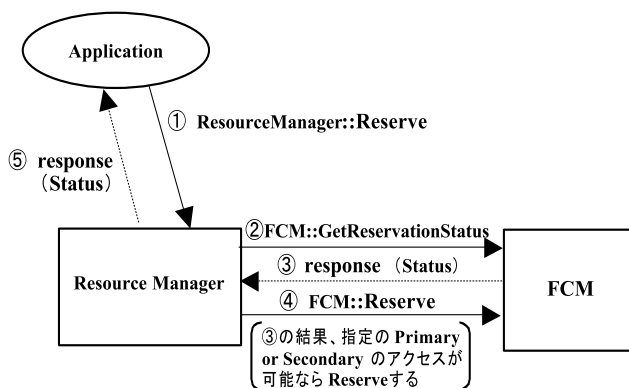


図8 リザーブの手順

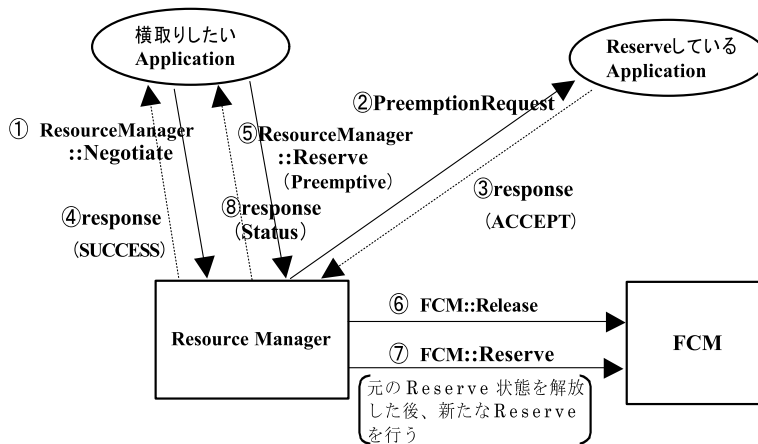


図9 ネゴシエーションの手順

FCM プラグなどをパラメータとして Resource Manager の ScheduleAction API を呼び出す。Resource Manager は指定の時刻において全ての必要なリソース(DCM, FCM, コネクション関連のリソース)がホームネットワーク上に存在し, なおかつ確保(リザーブ)できることをチェックした上でこの Scheduled Actionの動作を保証する役目を果たす。この際関連する DCM に対して予約状況のチェックを依頼する。開始時刻に実際に各種コマンドを実行するのは Resource Manager 自身でも良いが, コントロールアプリケーションと呼ばれるソフトウェアエレメントで実行させることもできる。コントロールアプリケーションを使用する場合, Schedule Action API を呼び出す際にコントロールアプリケーションの SEID を指定しておくこと開始時刻に Resource Manager がコントロールアプリケーションに対して通知を発行することができ, これをトリガとして実際の動作が開始される。またネットワークから機器が取り外されるなど予約を受け付けた Scheduled Action が実行できない状態になることも考えられるが, Resource Manager はこのような場合にイベントを発行し, ソフトウェアエレメントに通知を行う。

5. まとめ

HAVi の機器制御モデルとソフトウェアアーキテクチャおよび提供されるサービスについての概略を説明した。AV 機器を HAVi に対応させることによりマルチブランドによるホームネットワークを柔軟に運用し, 魅力的なアプリケーションを開発することができる。

HAVi 仕様は 2001 年 5 月にバージョン 1.0 の改訂版であるバージョン 1.1 が公開され, HAVi 仕様に基づいた製品が普及するための環境が整いつつある。今後は Jini や UPnP など他のネットワークとのブリッジの開発や他のカテゴリーの機器を HAVi に適用するための活動が行われていく予定である。

参考文献

(1)HAVi SPECIFICATION 1.1

- (2)<http://www.havi.org/>
- (3)IEEE Std 1394-1995, Standard for a High Performance Serial Bus.
- (4)ISO/IEC 13213:1994 Control and Status Register (CSR) Architecture for Microcomputer Buses (IEEE Std 1212-1994).
- (5)<http://www.1394ta.org/>
- (6)AV/C Digital Interface Command Set General Specification Version 3.0 April 15, 1998
- (7)IEC 61883 Parts 1 - 5, Standard for a Consumer-Use Digital Interface.

筆者

樋口 正生(ひぐち まさお)

- a. 研究開発本部 AV 開発センターホームネットワークシステム開発部
- b. 1990 年 4 月
- c. 光学式ピックアップの自動調整検査装置開発, 液晶プロジェクターの自動調整装置開発, HD-DVD デコードシステムの開発, MPEG ストリーム変換技術の開発, ホームネットワークソフトウェア技術の開発
- d. MPEG, IEEE1394 関連ソフトウェア開発

森岡 隆一郎(もりおか りゅういちろう)

- a. 研究開発本部 AV 開発センターホームネットワークシステム開発部
- b. 1992 年 4 月
- c. Write Once デジタルビデオディスクのスピンデルサーボ回路開発, 固体メモリ映像システムの開発, HD-DVD 用 UV-LBR のサーボ系回路開発, 高密度 ROM ディスク再生信号処理の研究など
- d. Java ソフトウェア開発

稲垣 勝利(いながき かつとし)

- a. 研究開発本部 AV 開発センターホームネットワークシステム開発部
- b. 1991 年 4 月
- c. CATV スクランブル開発, デジタル画像圧縮の研究開発, CATV ヘッドエンド機器の開発, ホームネットワークソフトウェア技術の開発
- d. MPEG, Network 関連ソフトウェア開発

戸崎 明宏(とざき あきひろ)

- a. 研究開発本部 AV 開発センターホームネットワークシステム開発部
- b. 1982 年 4 月
- c. 3 ディスクハイビジョンプレーヤの開発, DVD 規格の開発, MPEG ストリーム変換技術の開発, ネットワークシステムの開発
- d. ソフトウェアシステム開発