

AV 機器相互運用のための通信ミドルウェア技術 (HAVi)

HAVi Home Audio/Video Interoperability

小 田 正*
Masashi Oda

要 旨

家庭内の機器をネットワークに接続することで、機器単体では提供できない、新たなサービスを提供することが可能となる。しかし、そのためには、複数メーカーの異なった機能を持つ機器が、相互運用されて、一つのホームネットワークシステムとして動作する必要がある。HAViは家庭内ネットワーク上の機器（特にAV機器）が連携して動作するために必要となる、ネットワーク経由での制御や管理の相互運用を行う通信ミドルウェアプロトコル仕様である。メーカーが異なっても、HAVi仕様に対応した機器であれば相互運用ができ、ホームネットワークシステムとしての快適で便利な新たなサービスの提供が可能となる。本稿ではHAViの特徴や技術の概要を述べる。

Home appliances connected by home network will be given new functionalities that aren't possible when they are isolated. However, this will entail cooperation between products from different manufacturers and with different functions.

HAVi is a communication middleware protocol specification pertaining to via-network control and management interoperability, both of which are necessary for such cooperation among appliances - especially, audio-visual appliances.

The protocol specification enables interoperation between appliances from different manufacturers and contributes to new services that are more convenient and comfortable to use.

The purpose of this paper is to briefly describe its outline.

まえがき

AV機器をデジタルな通信路で接続し、音楽や映

像のコンテンツそのもののデジタル情報以外に、機器制御のための情報も通信可能なネットワーク環境において、ユーザにそのネットワーク環境での利便性をより効果的に提供するための通信ミドルウェアプロトコル技術としてHAViがある。ここでは、バージョン1.0betaとして公表されているHAVi仕様案のドキュメント¹⁾を元に、HAViの目指している目標や、その目標を実現するために策定中の規定の概要を説明する。

1. HAViとは

HAVi (Home Audio/Video interoperability) は、当社を含む日欧8社（グランディヒA.G.、株式会社日立製作所、松下電器産業株式会社、フィリップスエレクトロニクスN.V.、シャープ株式会社、ソニー株式会社、トムソンマルチメディアS.A.、株式会社東芝）が共同で仕様策定／普及活動を進めている、家庭内ネットワーク上の機器（特にAV機器）が連携して動作するために各機器に必要な、ネットワーク経由での制御や管理を行うための相互運用アーキテクチャの通信ミドルウェアプロトコル仕様である。

本原稿執筆時点においてバージョン1.0betaの仕様が公開されており、バージョン1.0としての正式リリースへ向けて、仕様内容の検証テスト等が行われている。

HAViアーキテクチャは、コンシューマエレクトロニクス機器とコンピュータ機器とに適用することができ、家庭内ネットワークで提供されるアプリケーションの相互運用と開発を容易にするためのサービスを提供する。

HAViのバージョン1.0では、複数の音楽や映像のデジタルなストリームデータを同一の通信路上で送受信することが可能な通信媒体である、IEEE1394を利用することを意図して規格化を行っているが、HAViのアーキテクチャ自身は通信媒体を限定するものではなく、将来的に、他の通信媒体での利用も可能である。

* 情報家電開発本部 マルチメディア開発研究所

2. 目的

DVD プレーヤや DV カムコーダ等の現在のコンシューマエレクトロニクス機器は精巧化したデジタル処理とデジタル記録のシステムである。これらの機器をネットワークで接続することで、処理と記録の機能を複数の機器間で共有することが可能となり、

(1) 複数のコンシューマエレクトロニクス機器のコントロールを同時に行う。

(2) ユーザによる機器の操作を簡素化する。

のような新しいアプリケーションが実現可能となる。

ホームネットワークは、このように、個々の機器が家庭内で独立した存在である場合はできなかつたり、できたとしても複雑な操作が必要であったサービスを、機器をネットワークで接続することで、簡単な操作で実現できる環境をユーザに提供するための仕組みである。またホームネットワークは、複数のメーカーのコンシューマエレクトロニクス機器が接続される、分散コンピュータ環境とみなすことができ、それらの異なったメーカーの機器が相互運用されて、一つのシステムとしてサービスを提供する必要がある。特に、AV機器を含むコンシューマエレクトロニクス機器を接続するホームネットワークは、接続された機器に制御情報のやり取り(一つの機器が別の機器にコマンドを送信する)以外に、AV コンテンツのやり取り(一つの機器が別の機器にオーディオやビデオのストリームを送信する)を行うことを可能にしなければならない。

また、このようなホームネットワークが成功するためには、

- (1) 高いデータレートの AV ストリームの転送
- (2) 自動設定と自動管理
- (3) ホットプラグアンドプレイ
- (4) 低価格のケーブルとインタフェース

等の条件を満たす必要がある。

IEEE1394 は、ホームネットワークの条件の多くを満たしている強力な技術であるが、ユーザに操作性が良く快適で便利な環境を提供するためには、それだけでは十分でなく、ホームネットワークに接続されている機器の機能や状態の管理を行い、ホームネットワークのシステムとしてのサービスをユーザに提供する仕組みが必要である。

HAVi アーキテクチャは、コンシューマエレクトロニクスメーカーやサードパーティに、ホームネットワークのシステムとしてのサービスを提供するための機器やアプリケーション開発を可能にする、通信ミドルウェアプロトコルの仕組みであり、アプリケーション開発のためのアプリケーションプログラミングイン

ターフェイス (API) を規定している。

3. HAVi の特徴

HAVi アーキテクチャの主要な特徴として

- (1) プラグアンドプレイ
- (2) 機器の相互操作性
- (3) ネットワークの拡張性

等のユーザに対する利便性向上がある。

3.1 プラグアンドプレイ

ホームネットワークに接続されるコンシューマエレクトロニクス機器は、利用するために複雑な初期設定を必要とする従来のパソコンのような機器とは異なり、ケーブルを物理的に接続すること以外、どのような作業もなしで簡単にインストールできて、ユーザにそれらの機器やホームネットワークの機能やサービスを提供できることが重要である。

プラグアンドプレイとは、ユーザが様々な機器を接続するだけで家庭内ネットワークを構築することができることである。このネットワークに、例えば新しく機器を接続したり、また取り外した場合でも、機器同士が通信し合ってネットワークが更新されたことを認識できるため、ネットワークはその機能を停止することなく、新しい機器配置に自動的に対応する。

例えば、時計機能を持つ機器を使用する場合、従来は初期操作として「時刻設定」が必要であったが、HAVi アーキテクチャを使用すれば、その機器をホームネットワークに接続した時に、ホームネットワークに接続されている別の機器の持っている時計の値にセットすることが可能となる。

3.2 機器の相互操作性

一般的に、ホームネットワークには、複数のメーカーのコンシューマエレクトロニクス機器が接続されて利用される。HAVi アーキテクチャでは、それらの異なったメーカーの機器であっても相互接続ができ、相互運用や相互操作を可能にしている。さらに、ネットワーク上の機器間で機器の持つ機能を共有することが可能であり、例えばある機器を操作することで、その機器が持たない他機器の機能を利用することも可能となる。

3.3 ネットワークの拡張性

ホームネットワークでは、新しい製品や機能が既存の製品や機能と共に動作することが重要である。ネットワークの拡張性とは、将来の新しいホームネットワークアプリケーションで使われる新しい機能を、

ユーザが既に利用しているホームネットワーク上で利用するための拡張性で、HAVi アーキテクチャでは、ソフトウェアベースのメカニズムによって、その機能が用意されている。従って、新たな機器や機能が開発された場合でも、すでに構築されたネットワークに機器を接続するだけで動作させることが可能である。

4. HAVi 技術

4.1 機器のクラス分類

HAVi アーキテクチャでは、コンシューマエレクトロニクス機器を、

- (1) Full AV devices (FAV)
- (2) Intermediate AV devices (IAV)
- (3) Base AV devices (BAV)
- (4) Legacy AV devices (LAV)

の 4 つのカテゴリに分類している。

HAVi アーキテクチャ対応の機器は FAV, IAV, BAV の 3 カテゴリの機器であり、他の HAVi に対応していない既存のコンシューマエレクトロニクス機器は LAV のカテゴリの機器になる。コントローラ機器とコントロールされる機器との区別を見ると、FAV と IAV がコントローラ機器で、BAV と LAV がコントロールされる機器である。

4.1.1 FAV(Full AV Devices)

FAV 機器は、HAVi アーキテクチャを実現するのに必要なすべてのソフトウェア要素の集合で構成される。この機器クラスは一般的に豊富な機能を持ち、また複雑なソフトウェア環境をサポートする能力がある。FAV 機器の主要な特徴は Java[®] バイトコードのランタイム環境の存在である。これは、FAV 機器に、他の機器からバイトコードをアップロードし、それらを制御するための能力の拡張を行うことを可能にしている。

4.1.2 IAV(Intermediate AV Devices)

IAV 機器は FAV 機器より機能において限定されているコントローラ機器であり、一般的に、FAV 機器と比べて低コストで実現可能である。IAV 機器は Java バイトコードを実行するランタイム環境を有しておらず、従って、ホームネットワーク内の任意の機器に対するコントローラとしては動作せず、ホームネットワーク内の特定機器の制御に対するサポートを固定的に提供する。

4.1.3 BAV(Base AV Devices)

BAV 機器は、アップロード可能な Java バイトコードを保有することで、将来の動作保証を実現することを選択した被制御機器であり、HAVi アーキテクチャのどのソフトウェア要素のホストでもない。BAV 機

器は、アップロード可能なバイトコードによる FAV 機器での制御、または固定的コードによる IAV 機器での制御が可能である。

4.1.4 LAV(Legacy AV Devices)

LAV 機器は、HAVi アーキテクチャが現れる前に作られた、HAVi 機器と通信路で接続することは可能であるが、HAVi アーキテクチャを有さない既存の機器である。これらの機器はコントロールのために、一般的に簡単で一方的なコントロールを受けるのみの、独自のプロトコルを利用している。そのような機器をホームネットワークで動作させるためには、FAV 機器または IAV 機器がゲートウェイとして動作することを必要とする。

4.2 制御モデル

ホームネットワークは、AV 機器の 1 つの集合から成っていると考えられる。個々の機器は、最低限、システムの他の機器と通信することを許すために十分な機能を持っている。これへの唯一の例外はレガシー機器である。

HAVi では、対話の過程で、機器は一对一の制御とデータの交換を行うことができる。これは、通信レベルにおいて、どの機器も、システムのためのマスターやコントローラとして動作することを要求されないことを保証している。但しこれは通信媒体そのものの通信制御構造までは含まない。

HAVi 制御モデルは、コントローラとコントロールされる機器の区別をする。コントローラは、コントロールされる機器のためのホストとして動作する機器である。コントロールされる機器及びそのコントローラは、同じ物理的な機器上に、または別々な機器上に存在することができる。HAVi コントロールモデルにおいて、コントローラは、コントロールされる機器に対する DCM ホストと呼ばれる。コントロールインタフェースはこの DCM の API を経て開示され、この API が、アプリケーションが機器をコントロールするための唯一のアクセスポイントである。

例えば、居間のインテリジェントテレビは、多くの相互運用された機器のためのコントローラになりうる。コントロールされる機器は、機器のためのユーザインタフェースを構成し、機器の外部コントロールを許す Java バイトコードを含んでいる。最初これらの機器が接続される時に、コントローラはユーザインタフェースと制御コードを得る。機器を表しているアイコンは、テレビスクリーンに表示され、アイコンを操作することは、表示されている機器や機器グループを、規定された方法で操作するための、制御プログラムの要素を動作させることになる。

4・3 機器モデル

機器間及び機能要素間で区別が行われる。例えばテレビは一般に1つの物理的なボックスであるけれども、それはいくつかの明瞭なエンティティ、例えばチューナー、音声出力などを含んでいる。機器内の制御可能なエンティティは、機能要素と呼ばれ区別して認識される。

4・4 HAViのソフトウェアアーキテクチャ

4・4・1 オブジェクトベース

HAViアーキテクチャのサービスはオブジェクトとしてモデル化される。個々のオブジェクトは、ソフトウェア要素と呼ばれる、規定されたインタフェースを通してアクセス可能で、そのオブジェクトが動作する機器のホストのソフトウェア実行環境で実行される、自己完結型のエンティティである。種々の機器が種々の実行環境をホストできることに注意する必要がある。サービスは、通信インフラストラクチャーを使って、それらの規定されたインタフェースを経てアクセスされる。HAViアーキテクチャのサービスは機器メーカから提供されるか、またはサードベンダにより付加できる。

4・4・2 ソフトウェア要素識別子

個々のオブジェクトはユニークに名前が付けられる。システムサービス構築のオブジェクトとアプリケーションサービスのオブジェクトの間で区別はされない。すべてのオブジェクトは、彼ら自身を、Registryとして知られているシステム全体の名前付けサービスによって登録／管理され、システムのオブジェクトは、他のオブジェクトを見つけるためにRegistryに質問し、メッセージをそれらのオブジェクトに送信するために、その質問の結果を用いることができる。オブジェクトを登録する前に、オブジェクトに割り当てられた識別子がMessaging Systemによって作成される。

これらの識別子はSEID (Software Element Identifier: ソフトウェア要素識別子) として参照される。たとえオブジェクトに割り当てられたSEIDがホームネットワークの再設定、すなわち機器のプラグインや除去の結果としてどのように変わっても、SEIDはユニークであると保証される。

4・4・3 メッセージベースの通信

他のオブジェクトのサービスを利用しようとするどのようなオブジェクトも、ターゲットのオブジェクトにサービス要求を引き渡す汎用のメッセージ通信メカニズムを使って行う。ターゲットオブジェクトは、上述のユニークなSEIDを使って指定される。この汎用メッセージ通信メカニズムは物理的な位置の概念から分離している。すなわち、同じ機器上のオブジェクト

とリモート機器のオブジェクトを区別しない。メッセージ通信メカニズムの実際の実現方法は機器毎に、またメーカの間で異なる。しかしながら、相互運用性の保証のために、HAViメッセージのフォーマットとその伝達のために使うプロトコルは共通にする必要がある。オブジェクトモデルとメッセージングシステムの一般的な意図は、各種のソフトウェアシステムと言語によって複数の実現方法を許すために、十分に柔軟な完全に一般的なソフトウェアモデルを提供することである。メッセージとそれらを扱うコード間の結び付けの詳細は、システムの実現方法に委ねられる。

4・5 ソフトウェア要素

HAViアーキテクチャのソフトウェア要素は、

- (1) ネットワーク管理
 - (2) 機器抽象化
 - (3) 機器間通信
 - (4) 機器ユーザインタフェース (UI) 管理
- の基礎的な概念をサポートする。

全体として、これらのソフトウェア要素は、ホームネットワークのポータブルな分散したアプリケーション構築のための相互運用APIと、サービスのセットを開示する。ソフトウェア要素自身はリアルタイムオペレーティングシステム (RTOS) などのベンダ特定のプラットフォーム上に存在することになる。図1はFAV機器におけるソフトウェア要素の配置を示している。ここでは特定の実装方法は意図しておらず、HAViソフトウェア要素が、プラットフォームの特定APIとプラットフォーム非従属なアプリケーションの間の中間レイヤを形成していることのみを示している。

以下でHAViアーキテクチャを構成しこの規格で定義されているソフトウェア要素の概要を列記する。

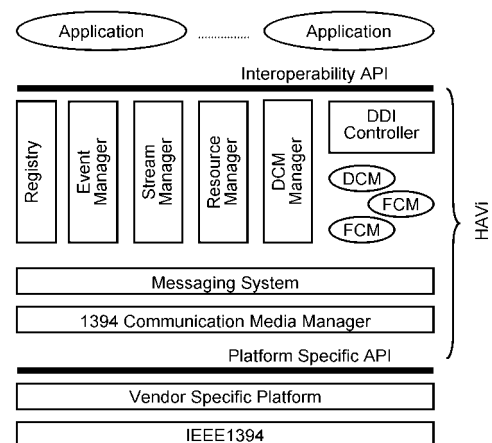


図1 FAV機器でのソフトウェア要素の構成

Fig. 1 Arrangement of software elements on a FAV device.

4・5・1 1394 Communication Media manager (CMM)

CMMはHAViネットワークにおいてネットワークに依存する要素である。CMMは通信媒体とインタフェースし、同じ機器上の他のHAVi要素やアプリケーションプログラムへのサービスを提供する。各々の物理的な通信媒体は、この目的を達成するために、各々独自のCMMを持つ。1394 CMMはIEEE1394バスのためのCMMであり、他のソフトウェア要素に、IEEE1394上の非同期と同期の通信を実行することを許す。1394CMMの主な機能は、

- (1) 他機器との間で命令を送信／受信するための伝送機構を提供する
- (2) Busの動作状態を把握し、その情報を他のソフトウェア要素に提供する

の2つである。

4・5・2 Message System (MS)

MSはすべてのFAV機器とIAV機器に組み込まれ、ネットワーク層やトランスポート層から独立し、HAViのソフトウェア要素に対して通信の機能を提供し、ネットワーク上の各機器のソフトウェア要素同士がコミュニケーションするためのAPIとして働く。したがって、メッセージ送り側と受け側では、互いに相手のネットワーク上での場所を知る必要はなく、メッセージ伝送はMSに任せることになる。MSは、その機器のソフトウェア要素に対する識別子(SEID)の割り当てを行う。その識別子は最初、登録するためにソフトウェア要素により使われる。その識別子はホームネットワーク内で、他のソフトウェア要素にメッセージを送るために、MSのAPIを呼び出す時に、お互いを識別するために利用される。

4・5・3 Registry

Registryは、ホームネットワーク内で利用可能なソフトウェア要素の一覧表を管理することを目的とするシステムサービスであり、ネットワーク上にどんな機器が接続されているか、またその機器がどんな機能を持っているかなど機器に関する情報の保持や更新を行う。また他のソフトウェア要素のために、登録するためのAPIと検索するためのAPIを提供する。RegistryサービスはIAV機器とFAV機器で存在する。1つの機器内では、どのようなローカルなソフトウェア要素でもRegistryを経由して自身を表現できる。もしソフトウェア要素が他のソフトウェア要素と関係したいならば、Registryに登録しなければならない。

またソフトウェア要素は、Registryのサービスを利用することで、ホームネットワーク上の別のソフトウェア要素を探し出すことができる。アプリケーションプログラムはRegistryから必要な情報を入手するこ

とで、異なる機器間の相互操作を可能にする。

4・5・4 Event Manager (EM)

EMはイベント配達サービスを提供する。イベントとはソフトウェア要素やホームネットワークの状態変化である。例えば機器の接続または切断は、ネットワークの状態の変化と考えられる。イベントの配達は、一つの機器内でローカルに行われるかまたは、ネットワークのすべての機器にグローバルに行われる。このサービスをサポートするために、EMは、ソフトウェア要素により投函されたイベントが、イベントを望むすべてのソフトウェア要素に達することを確実にするのに役立つように、エージェントとして機能する。もし特定のイベントが投函される時に、ソフトウェア要素がそれを通知されることを望むならば、ソフトウェア要素はそのローカルなイベントマネージャーにそのような意向を登録しなければならない。個々のEMは、ソフトウェア要素により登録されたイベントのリストを含んでいる内部テーブルを管理する。ソフトウェア要素がイベントを投函する時に、それは、EMから提供されるサービスを経て行われる。EMはその内部テーブルをチェックし、このイベントを登録したソフトウェア要素へ通知する。イベントを登録していないソフトウェア要素は、通知を受け取らない。もしイベントがグローバルで投稿されたならば、ローカルなEMはイベントをネットワークのすべてのリモートなEMに中継する。個々のリモートなEMは同じチェックを実行し、登録されているローカルなソフトウェア要素に通知する。EMはHAViのMSを利用して、通知されるソフトウェア要素へ通知メッセージを送ることで通知する。EMにより、ネットワークの状態変化を他のソフトウェア要素に知らせることで、プラグアンドプレイなどの利便性を実現できる。

4・5・5 Stream Manager (SM)

SMは、エンドエンド間の同期ストリーム接続を形成するためにAPIを使う容易さを提供する。接続は一对一かまたは放送である。SMは

- (1) 機器内の内部接続と機器間の外部接続の両方の設定
- (2) 転送のために必要なシステムリソースの要求と開放
- (3) グローバルな接続情報の提供
- (4) ネットワークリセット後の接続復元のサポート等を行うことで、ネットワーク上での映像や音声などのストリームデータの転送を監視／管理する。

4・5・6 Resource Manager (RM)

RMはリソースの共有や機能動作のスケジュールの管理を行い、ホームネットワークにおけるリソース及び動作のスケジューリングの共有を容易にする。ネッ

トワークのアプリケーションは、ユーザにサービスを提供するために、一般的に、FCMの組を用いる。FCMはこの状況において機器リソースと呼ばれてリソース管理される。またネットワークリソース(IEEE1394のバンド幅とチャンネル数)もリソース管理に含まれる。なぜなら、これらは、HAViネットワークのDCMやFCM間のオーディオやビデオストリームを転送することに必要なリソースだからである。

4・5・7 Device Control Module (DCM), Functional Component Module (FCM)

DCMはそのネットワークで認識されている個々の機器のために存在する、機器をコントロールするために使用されるソフトウェア要素である。DCMは、HAViアーキテクチャのソフトウェア要素としてそれを提示することで機器に対してインタフェースを提供する。DCMはDCMコードユニットから獲得される。DCMコードユニット内には、DCM自身のためのコードに加えて、機器内の個々の異なった機能構成要素に対するFCMのためのコードがある。さらに、DCMコードユニットは、機器及びその機能構成要素のユーザ制御を可能にするhavletを含んでいる。

DCMはHAViアーキテクチャの中心的概念で、新しい機器や機能を収容する柔軟性の源である。

DCMには

- (1) embedded DCM
- (2) uploaded DCM

の2つの主要なタイプがある。embedded DCMは、IAV機器及び場合によってはFAV機器にプリインストールされたDCMで、uploaded DCMはJavaバイトコードで実現された、FAV機器にアップロードされ、FAV機器においてのみ動作する、DCMである。

DCMは、APIを機器種別及び特定モデルの両方のコントロールに提供できる。一般に、前者のAPIは用法のより広い範囲を持ち、後者はベンダ特定の特徴と能力のコントロールを許す。

HAViアプリケーションは独自のコードで開発され、FAVまたはIAVに埋め込まれることが可能である。また、Javaバイトコードの形態であることも可能で、HAViにより定義されたメカニズムを使って、インターネット等の外部のソースや、または接続された機器のソフトウェア要素から得られる。アプリケーションプログラムはこのDCMを介して機器の制御を行う。このため、アプリケーションプログラム自体は個々の機器の違いを考慮する必要がなく、その結果、ホームネットワーク上の機器は、他の機器の機能を認識でき、離れた場所からの他機器の操作が可能とな

る。

4・5・8 DCM Manager (DCMM)

DCMの管理はFAV機器とIAV機器のDCMMによって実行される。DCMMはFAV機器とIAV機器にBAV機器とLAV機器の制御を行うためのDCMコードユニットのインストールとアンインストール及び更新を行う。新しい機器がネットワークに追加接続されると、その機器に必要なDCMを新たに加え、ネットワークの更新に自動的に対応する。なおFAV機器とIAV機器に対するDCMコードユニットは独自の方法によりその機器自身で管理される。

4・5・9 Data Driven Interaction (DDI) Controller

DDIはユーザインタラクションによって必要とされるソフトウェア要素である。DDI Controllerはユーザ入力操作を取り扱い、翻訳してDDI要素とする。DDI Controllerは機器の表示部のGUI(Graphical User Interface)を担当し、テキストだけの表示からグラフィックスの表示まで、多様なディスプレイに対応した表示を行う。つまり、DDIコントローラーは、ユーザと機器操作を助ける役目を果たすソフトウェア要素である。

4・5・10 HAVi Self Describing Device (SDD) data

HAVi機器は、機器及びその能力について記述された情報を保持している。HAVi SDDデータと呼ばれるこの情報は、ユーザインタフェース要素を構成するためのDCMコードユニットやデータを含みうる。

注：Javaは、米国及びその他の国における米国Sun Microsystems, Inc.の商標または登録商標です。

むすび

HAViアーキテクチャは、機器をネットワークで接続した場合に、異なったメーカーの機器間の相互運用を可能とし、ユーザに対して、簡単な操作で快適で便利な環境を提供するための通信ミドルウェアプロトコルの仕様であり、ホームネットワークシステムやそこに接続される機器を開発していく上で、必要不可欠な技術である。当社はその標準化活動に参画すると共に、ソフトウェア開発を進めている。

参考文献

- 1) The HAVi Specification Version:HAVi1.0beta, October 5, (1998).

(1999年10月5日受理)