HAVi: Home Audio Video Interoperability

Jussi Teirikangas
Helsinki University of Technology

Jussi.Teirikangas@hut.fi

Abstract

Eight major consumer electronics manufacturers have come up with an open standard enabling home entertainment devices to communicate intelligently with each other. The HAVi (Home Audio Video Interoperability) standard promises to bring true platform independent interoperability to consumer devices using high bandwidth IEEE 1394 (FireWire) as the connecting medium. This paper studies the HAVi standard and what it has to offer to consumers.

1 INTRODUCTION

An average household nowadays contains many very complicated devices. Many of them are home entertainment devices related to handling different audio or video data. These devices are computers in essence, but just more specialized in their features than a home PC. Home networking has become very popular nowadays since a normal household might contain several PCs that need to use shared resources like printers or file shares. Home audio and video devices like VCR, TV, amplifier, tuner, DVD, CD player and set-top-box form a similar interconnected network (see Figure 1). Why couldn't these miniature computers also make use of each other's features and even control each other to make everything easier for the consumer?

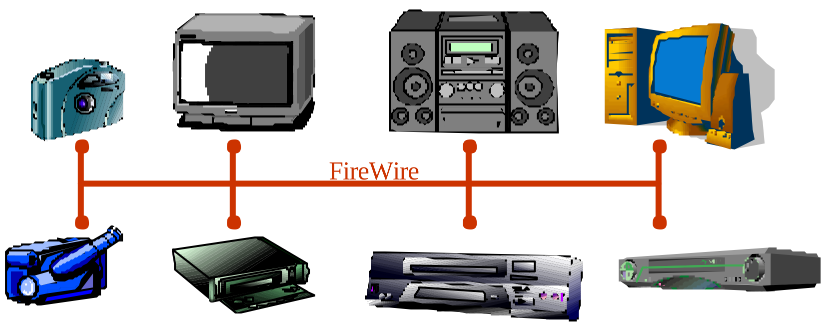

Figure 1: An Example of interconnected AV devices.

Major consumer electronics, software, semiconductor and computer manufacturers think that this should be possible and have decided to make it happen. The manufactures, namely Grundig, Hitachi, Panasonic, Philips, Sharp, Sony, Thomson and Toshiba along with now over

30 other participants, have formed a non-profit organization called HAVi (Home Audio Video Interoperability) for promoting the development of interoperable consumer products. The goal of HAVi organization is to provide a standard open architecture for intelligent audio and video devices to interoperate with each other regardless of manufacturer, operating system, CPU or programming language used for implementation (HAVi, Inc., 2001a).

The first beta version of the HAVi standard version 1.0 was published in December 1998 while the final 1.0 version was released in December 1999. The current version of the specification is 1.1 (HAVi, Inc., 2001b) and it was published in May 15th 2001.

This paper presents the basic architecture and promises the HAVi standard offers. Various problems and questions still to be answered will also be discussed. Although HAVi is still to come into living rooms as a de facto standard, a brief look at the future of HAVi will be made. The paper is mainly based on the information offered by HAVi organization (HAVi, Inc., 2001a) and naturally the HAVi specification version 1.1 itself (HAVi, Inc., 2001b). Another main source of the paper is a HAVi introduction by Rodger Lea, Simon Gibbs, Alec Dara-Abrams and Edward Eytchison (Lea et al., 2000).

2 PROMISES OF HAVI

The idea of an open standard sounds very promising, but how can a normal consumer benefit from it? How can it make lives easier and what kind of things, not possible before, can be achieved by using it?

The simplest example might be time synchronization between different devices. TV set might get the correct time from the broadcast stream and the other devices can query the TV and set their own clocks according to it. Setting the VCR to record a program is a familiar situation users usually have problems with. With HAVi enabled devices this task can be made very easy. User can select the program (s)he wishes to record with the Electronic Program Guide (EPG) residing on a digital TV set (or set-top-box). The task can be as simple as just browsing the program information, selecting the desired program and pressing one button to activate recording. The TV then locates an available recorder (e.g., a VCR or a recording DVD device) and commands it to record the program supplying it with the time, length and channel parameters taken from the EPG. Thus, the user doesn’t need to program or touch the recording device in any way.

One of the more advanced scenarios might be automatic directing of an oncoming videophone call to the TV screen or part of it and muting all other sounds. Similarly, if a camera placed outside the door detects movement, the picture is automatically connected to the TV screen notifying the user about a possible visitor. All this could also be aided by giving voice commands to the devices. These are only some of the possible use cases. A lot more can be possible, especially when the HAVi devices are connected to other home appliances, PCs or even Internet.

The possibilities HAVi offers seem endless and many of them might sound like science fiction or at least not likely in the near future, but that might not be the case. Many products have already been announced and several working demos have been presented at various consumer electronics fairs.

The interoperability of HAVi devices seems pretty extensive and complex. Will the installation and configuration of the network be as complicated as in computer networks? Fortunately no, since devices are hot-pluggable and they are supposed to automatically announce their presence and capabilities to other devices and configure themselves when connected to the network saving the user from reading installation instructions and configuring network addresses and drivers.

Finally, HAVi standard promises to be future proof by maintaining current functionality while making it easy to upgrade and add new capabilities. Non-HAVi devices can also be connected to the network if at least one of the HAVi devices supports the interface the legacy device provides.

3 HAVI ARCHITECTURE

The HAVi architecture can be divided into several different layers (see Figure 2). On the bottom there is always vendor specific hardware and software Application Programming Interface (API) that HAVi is built upon. Also, on the hardware level there is the connecting IEEE 1394 wiring, which HAVi devices use as a connecting medium. Next, a media manager for IEEE 1394 is needed as well as a messaging system. On top of the messaging system, there are several software modules: Registry, Event Manager, Stream Manager, Resource Manager, Device Control Modules (DCM) and DCM managers. These layers compose the basic services for building portable distributed applications. Basic services provided by the system, include:

- Automatic discovery or devices added or removed from the network
- Addressing scheme and message-transfer service
- Lookup service for discovering resources
- Posting and receiving local or remote events
- Automatic installation and configuration of DCMs
- Streaming and controlling isochronous data streams
- Reserving devices and performing scheduled actions
- Device control via DCMs and FCMs
- User interaction with UI mechanisms

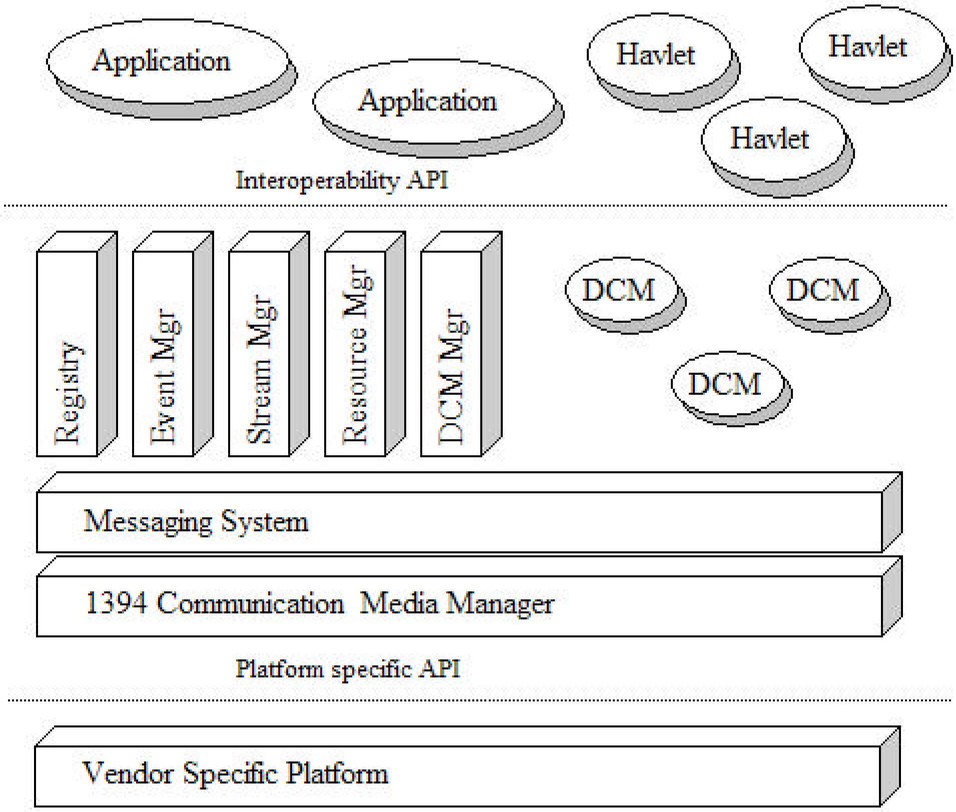

Figure 2: The architecture of a fully compliant HAVi device. (after HAVi, 2001b)

3.1 HAVi Devices

HAVi devices are classified into four categories (see Table 1): Full AV devices (FAV), Intermediate AV devices (IAV), Base AV devices (BAV) and Legacy AV devices (LAV). HAVi compliant devices fall into the first 3 categories and all other devices belong to the 4th category. FAVs and IAVs are controlling devices in the HAVi network while BAVs and LAVs are the devices they control.

Table 1: Device classes and software elements required for the four HAVi device classes. The use of parentheses indicates optional component.

Software element	FAV	IAV	BAV	LAV
Java	x			
DDI Controller	(x)	(x)		
Resource Manager	x	(x)		
Stream Manager	x	(x)		
DCM Manager	x	(x)		
Registry	x	x		
Event Manager	x	x		
Messaging System	x	x		
Communications Media Manager	x	x		
Device Control Module	x	(x)	x	x
IEEE 1394	x	x	x	(x)

IEEE 1212r	x	x	x	(x)
IEC 61883.1 FCP	x	x	(x)	(x)
IEC 61883.1 CMP	x	(x)	(x)	(x)

A Full AV device has the most complete set of HAVi features. FAV contains a runtime environment for Java bytecode allowing it to upload bytecode from other devices. This feature provides much enhanced capabilities for controlling devices. Common FAV devices might be set-top-boxes, digital TV receivers, general-purpose home-control devices, residential gateways or even PCs.

IAVs are generally a bit cheaper and do not contain the Java environment, thus having more limited capabilities for controlling other devices. IAVs may provide native support for controlling particular devices on the home network. Home theater amplifiers or DVD players might fall into this category.

BAVs do not contain any of the HAVi software modules. However, their configuration Rom must contain uploadable Java bytecode that makes it possible for FAV devices to control them. They can also be controlled by a IAV device using native code. Most likely BAV devices include portable audio players, camcorders and mass storage systems.

LAV devices do not recognize HAVi architecture. They use proprietary control protocols. LAV devices can be divided into two categories, IEEE 1394 devices and those not supporting it. They can be connected to the HAVi network, but they need a FAV or IAV device acting as a gateway for them.

3.2 IEEE 1394

To meet the requirements for real-time transfer of high-data-rate streams, self-management and autoconfiguration, low-cost cabling and interfaces, a natural choice was to adopt the IEEE 1394 (FireWire) standard (IEEE, 1996) first conceived by Apple Computer. IEEE 1394 meets all these requirements. Its high data rate of 400 Mbps (upgradeable to 800 Mbps or even 1600 Mbps) is quite enough for several simultaneous data streams. Data can also be full-duplex, i.e., both data and control instructions can flow to both directions at the same time. Scalability of up to 63 devices in the same bus should be quite enough for normal consumer electronics. FireWire can also connect almost any kind of computer peripherals such as printers, scanners, keyboards, displays and hard drivers. However, so far it isn't very popular in normal PC environment except connecting workstations with digital video cameras for video editing. Naturally Apple products utilize it more commonly since it is a built-in feature in their workstations.

On top of the IEEE 1394 layer, HAVi uses a simple Function Control Protocol (FCP) defined in IEC 61883.1 (IEC, 1998,) for the transport of command requests and responses. IEC 61883.1 also specifies a Connection Management Protocol (CMP) for creating, breaking, overlaying and restoring isochronous connections and for Common Isochronous Packet (CIP) format.

3.3 Software Modules

HAVi software elements are self-contained entities that communicate with each other in peer-to-peer fashion. Each software element has a well-defined API through which the services can be accessed. Elements also have a unique identifier assigned by the Messaging System before they register to the Registry. Since this unique identifier is used to pass messages between different modules, there is no distinction whether the modules reside within the same device or different devices on the same network. In addition to assigning unique ID to the software elements, the messaging system fragments the messages into multiple FCP packets and reassembles them. Software elements can request the messaging system to supervise other elements and notify if they become unavailable. The HAVi messaging system supports both acknowledged and unacknowledged messaging.

Registry acts as the directory service of the network. It enables objects to locate other objects on the same network. In addition to the unique identifiers, the registry contains a small set of software element attributes. Clients can then, e.g., query the attributes of a specified element or locate an element matching a search predicate. Registry also forwards the queries to all remote registries and returns the replies to the client.

Device Control Modules allow the controlling device to control other devices. A DCM might contain several function specific Functional Component Modules (FCM). HAVi 1.1 defines the following FCMs: tuner, VCR, clock, camera, AV disc, amplifier, display, AV display, modem and web proxy. DCM of a controlled device can be embedded in the controller or dynamically added by, e.g., uploading. HAVi also provides standard DCMs for controlling various kinds of devices, but proprietary DCMs can add vendor specific features and enhanced capabilities. DCM Manager is needed to make sure that each target device has only one DCM on the network. The hosting device is chose by a voting process. Device candidates can affect the voting process by setting preferences. Some device might be better suited to control the device than other candidates.

In some cases it is useful for objects to notify any changes its state to other objects. Event manager monitors these events and posts a message to local software elements that have subscribed to that event. It can also forward the events to other event managers for global posting.

Stream manager is responsible for managing transfer of real-time streams on the network. The transfer can happen internally or between different devices. Applications create the streams by defining a source and destination and then invoking the local stream manager. Stream manager verifies that the source and destination type are compatible and then allocates the needed resources.

Resource manager is used to reserve and release FCMs and arrange scheduled actions. Managing is needed to prevent conflicts between devices. Only one device should command a controlled device at any time. However, this only applies to commands that require, e.g., a change in the state of the device, but not commands that only get or view information.

3.4 User Interfaces

HAVi devices can be controlled through other HAVi devices on the same network. Thus, the user interface also needs to be portable. Device manufacturers can define graphical user interfaces (GUI) that can be rendered on a device with display capability. There are two ways to achieve this Level 1 UI and Level 2 UI.

Level 1 UI is called Data Driven Interaction (DDI). DDI user interface elements (buttons, panels, etc.) are typically obtained from DCMs. DDI controller then connects to the controlled device and sends user action messages to it. The target notifies the state changes to the DDI controller. DDI controller just provides a way to remotely command a device; it doesn't have any intelligence about actions and their consequences. Target device can suggest a preferred layout of the user interface, but the DDI controller might modify it depending on the display capabilities. If more than one DDI controller is connected to the target, all the views are synchronized by the target.

Level 2 UIs are constructed with Java and support more advanced features based on a subset of Java AWT 1.1. HAVi also defines some extensions, such as support for different screen sizes and aspect ratios, alpha blending and video/image layering, support for remote control input and support for UI components patterned after Level 1 DDI elements.

3.5 Security

All devices can send messages and events to each other without any restrictions. To avoid some of the problems that may arise, HAVi specifies what type of messages and events each software element is allowed to send. Receiving system component will then check if the sender is allowed to send this message or event. For example, some system components might only accept messages from other system components. Protection from hostile or flawed applications is left to the device manufacturer.

HAVi protection scheme has only two levels, trusted and untrusted. Vendors should thoroughly test all system and other software elements before granting them trusted level. All dynamically installed software, like updates and software patches, should be verified to make sure they are valid and come from verified sources. This verification is done by digital signature algorithm specified in the HAVi standard.

4 PROBLEMS WITH HAVI

HAVi as a technology hasn't been widely tested and utilized in real environments. Naturally, until it is proven that everything works as it is supposed to, there are several problems to be foreseen.

One of the most important goals is platform-independent interoperability. However, it has been seen that FireWire still isn't as solid as you would think considering that it has been on the market for several years. In fact, FireWire alone has proven to be very complicated to implement. The only guarantee is that devices of the same brand and same and same kind of proprietary programming will most likely work together. Until all the major problems with FireWire have been solved, they will hinder HAVi.

Also, distributing audio and video data is still quite difficult since there's no one format that all devices understand. For example the data formats of a VCR, minidisc, CD player and MP3 player are quite different.

One important issue when dealing with home entertainment is digital rights management. Without any copyright info data might be transferred freely between devices or even outside the home network. While it is legal to copy material for your self, the situation becomes more complicated when the HAVi network is connected to, for example, the Internet.

Big entertainment corporations will not like the fact that their material could be freely distributed. This raises the question about how the devices are protected from intrusions from outside. HAVi specification leaves this issue mostly to the device manufacturers. It certainly wouldn't be nice if someone could disable your home network from outside, thus rendering video cameras, set to watch your apartment, useless.

In another scenario, an intruder might get access to your personal home videos. These scenarios might seem far-fetched, but more common situations might cause problems too. The network must withstand an attack from within too. Faulty device might act in a wrong way sending invalid messages or monopolizing the network with traffic. Also, since devices can obtain bytecode from another devices or even Internet, it should be made sure that faulty program code won't cause too much problems. Again, it can be visualized that one device gets an update containing a virus which it then uploads to every other device.

Due to the growing complexity of the devices and various standards, the initial models will most likely be priced quite highly for some time. A set-top-box implementing both HAVi and Multimedia Home Platform (MHP) will probably start with the price of a full-featured multimedia PC, which is too much for the majority of consumers. As long as these high-priced devices suffer from incompatibilities, they aren't too tempting. Users will not like to have their expensive digital TV displaying error messages or crash once in a while. The high complexity of the protocols makes the verification of products very difficult and even the smallest mistakes in mass production of consumer electronics can cost quite a lot, both financially and in credibility.

Engineers have nice visions about how devices can be connected with each other and the Internet, but the reason to do this just shouldn't be 'because we can'. Sure, a lot of things not possible before can be realized with HAVi. HAVi promises easiness of use, which is welcomed since even setting the VCR to record can be a very formidable task sometimes. However, so far the discussion and development has been quite technology driven and issues, such as usability and what consumers actually want, haven't been discussed enough. Hopefully this situation will change when the technology is ready and devices become more common.

Finally, Microsoft also has its position in the development of home networks and has introduced a competing standard, Universal Plug and Play (UPnP). UPnP is more tilted towards linking PCs with all kinds of home appliances together. HAVi and UPnP complete each other in many ways, but clashes between the two efforts might make situation a bit more complicated and slow down the development of devices.

5 THE FUTURE OF HAVI

HAVi technology was first demonstrated at Winter CES (Jan 2001) in Las Vegas. After that, many of the participating companies have announced HAVi enabled products during the last year. However, most of the products are not publicly available yet. Naturally, the first HAVi compatible devices are targeted towards high-end markets, i.e., to professionals and home theater enthusiasts. It is most likely that the situation won't change much with next 2-3 years at least, not until HAVi technology is more mature and cheaper to implement.

To battle the high cost of the products and time to develop them, T. Nakajima (Nakajima et al., 2001) presents a cost effective way of developing HAVi appliances. This solution uses Linux as the operating system and Java as the programming language. Transferring the system and applications to the product is easy since the development platform and the final product can both use Java and Linux. Linux was a natural choice, because it has recently gained popularity in embedded systems. Since virtually no modifications are needed, the development time and cost should be greatly reduced. However, Linux still has some problems to be solved, such as real-time resource management and making the memory footprint small enough.

Another development area is connecting HAVi to other networks, namely the Internet. This allows appliances to be remotely commanded with any device with a browser connected to the Internet. For example, it could be possible to use a PDA to remotely program the VCR. R.G. Wendorf and M.P. Boedlander (2000) have demonstrated this in their paper. In this solution, a Messaging System Proxy encodes HAVi messages into Extensible Markup Language (XML) and SOAP. The encoded messages are then sent to Internet using Hypertext Transfer Protocol (HTTP). Vice versa, incoming messages are stripped from XML and SOAP and put on the HAVi network. All this functionality can be achieved without modifying the HAVi specification itself or adding features to Internet connected devices.

6 CONCLUSIONS

HAVi comes with great promises, promises that seem to be quite troublesome to fulfill. However, with so many large companies working for a common goal, there is a good chance that HAVi will eventually prove to be what it promised. In the end, it is up to customers if they want to adopt the new technology. When new features and possibilities are as great as with HAVi, there's no question about the fact that HAVi will be something that customers want, provided that it comes with a reasonable price tag and solid functionality.

7 REFERENCES

Bodlaender, M.P.; Wendorf, R.G., 2000, Adding full internet protocol functionality to HAVi, *Proceedings of the ICCE International Conference on Consumer Electronics 2001*, pp. 300-301

IEEE, IEEE standard 1394, 1996, *Standard for a High Performance Serial Bus*, Piscataway, N.J., IEEE Press, parts 1 - 5

IEC, IEC standard 61883, 1998, *Consumer Audio/Video Equipment – Digital Interface*, Geneva, Switzerland , Int'l Electrotechnical Commission,

HAVi, Inc., 2001a, The HAVi Homepage [online] <URL:http://www.havi.org> [referenced Nov-22, 2001]

HAVi, Inc., 2001b, *The HAVi Specification Version 1.1*, 529 p. Also available at <URL:http://www.havi.org/techinfo/index.html>

Lea, R.; Gibss, S.; Dara-Abrams, A.; Eytchison, E., 2000, Networking Home Entertainment Devices with HAVi, *Computer*, Vol. 33 Issue 7, pp 171-178

Nakajima, T.; Soejima, K.; Matsuda, M.; Iino, T.; Hayashi, T, 2001, Design and implementation of distributed object-oriented infrastructures for networked home appliances on commodity operating systems, *Proceedings of the Fourth International Symposium on Object-Oriented Real-Time Distributed Computing 2001*, pp. 171-178

Wendorf, R.G.; Bodlaender, M.P., 2000, Remote execution of HAVi applications on internet-based devices, *Proceedings of the ICCE International Conference on Consumer Electronics 2001*, pp. 232-233