

# HAVi 홈 네트워크와 연동을 지원하는 Jini 서비스의 구현

박동환<sup>0</sup> 구태연 문경덕  
한국전자통신연구원  
(dhpark, kutai, kdmooon)@etri.re.kr

## Implementation of Jini Service supporting interoperate with HAVi Home Network

Dong-Hwan Park<sup>0</sup> Tai-Yeon Ku Kyeong-Deok Moon  
Dept. of Office, Electronics & Telecommunication Research Institute

### 요 약

홈 네트워크 미들웨어 중 디지털 멀티미디어 데이터의 전송을 위한 표준인 HAVi는 IEEE1394를 네트워크 프로토콜로 사용함으로써 디지털 오디오/비디오 데이터의 효과적인 송수신이 가능하고 동적인 네트워크 재구성이 가능하다. 하지만, 이들 디지털 A/V 기기를 제외한 거의 모든 홈 네트워크 정보 단말은 TCP/IP를 통신 프로토콜로 사용하고 있으며, 이를 위한 미들웨어로 Jini 기술이 제안, 적용되고 있다. Jini는 자바 기술에 기반을 뒀으로써 플랫폼 독립적인 특성과 코드의 이동성이 보장되는 특징이 있으며 network-and-play를 지원한다. 본 논문은 Jini 네트워크에서 HAVi네트워크에 존재하는 디지털 AV기기의 제어를 위한 HAVi 네트워크 제어 가능한 Jini 서비스의 구조와 제어 기법을 제안한다.

### 1. 서 론

홈 네트워크는 디지털 가전 기기와 정보 단말기기의 급속한 개발과 보급으로 인해 점차 현실로 다가오고 있다. 하지만, 홈 네트워크의 보급을 위해서는 홈 네트워크의 구성과 서비스를 위한 요구 사항인 네트워크 관리의 용이성, QoS(Quality of Service)의 보장, 설치 및 사용의 용이성, 디지털 오디오와 비디오 같은 멀티미디어 서비스에 대한 사용자의 요구 등을 만족시켜야 한다. 이러한 요구사항을 만족시키기 위해서 홈 네트워크를 위한 미들웨어의 연구와 개발이 활발히 이루어지고 있다.

홈 네트워크 미들웨어 기술은 각각 제어 네트워크, 데이터 네트워크, AV 네트워크를 위한 미들웨어로 구분하고 있으며, 제어 네트워크 미들웨어로는 LonWorks 기술이, 데이터 네트워크 기술은 Jini와 UPnP가, AV 네트워크 미들웨어로는 HAVi(Home Audio/Video Interoperability)가 각각 제안되었다. 제어 미들웨어인 LonWorks기술은 네트워크 미디어로 TP, 전력선, 무선 등 다양한 미디어를 사용할 수 있으며, 주로 기기의 제어에 활용되기 때문에 낮은 전송속도를 가진다. 반면 데이터 네트워크 미들웨어는 현재 가장 보편적으로 사용되고 있는 TCP/IP를 기반으로 네트워크를 구성한다. UPnP가 주로 Microsoft사의 윈도우 운영체제를 기반으로 미들웨어의 구성이 이루어지고 수행이 되는 반면에, Jini기술은 Java언어를 기반으로 구성되므로 플랫폼 독립적인 특성을 가지며, 어떠한 환경에서라도 JVM(Java Virtual Machine)이 갖추어진 환경에서는 구성될 수 있다는 장점이 있다. AV네트워크 미들웨어인 HAVi는 IEEE1394 프로토콜을 네트워크

프로토콜로 사용을 한다. 이는 IEEE1394의 높은 대역폭과 실시간 전송이 가능한 점을 적극 활용하는 것으로, 이를 이용한 멀티미디어 서비스와 QoS, 그리고 네트워크 관리의 용이성을 동시에 제공해 줄 수 있다. 하지만, HAVi 미들웨어가 디지털 AV기기에 국한된 서비스를 제공하기 때문에, 가정 외부에서 HAVi 미들웨어로 동작되는 홈 AV 네트워크 기기의 제어와 멀티미디어 콘텐츠를 이용할 수 있는 방법이 전무하다. 이러한 목적을 위해서는 외부 망에서 홈 네트워크로의 접근이 가능한 데이터 네트워크 미들웨어가 다른 미들웨어의 기기를 제어하기 위한 서비스를 제공하는 것이 바람직하다.

본 논문에서는 홈 데이터 네트워크 미들웨어인 Jini기술로 홈 AV 네트워크 미들웨어인 HAVi 미들웨어를 제어하고 이들 미들웨어간 연동을 지원하기 위한 Jini서비스의 구조를 제안한다.

### 2. 관련 연구

#### 2.1 Jini의 개요

Jini[1]는 Java를 기반으로 네트워크 상의 기기나 소프트웨어를 단순하고 신뢰성 있는 방식으로 네트워크에 연결된 각종 서비스를 동적으로 연계시키는 구조를 가지고 있다. Jini는 서비스를 구성하는 참여자들이 서로에 대한 사전 지식이 없이 Jini 네트워크에 연결만 되면 어떠한 설정 작업이 요구되지 않고도 서로를 인식하여 통신하고 서비스를 제공할 수 있도록 해준다[2, 8]. Jini에서 사용되는 서비스는 하드웨어와 소프트웨어의 구별이 존재하지 않는다.

네트워크상에 존재하는 서비스는 이것이 하드웨어로 구현되어 있거나, 소프트웨어로 구현되어 있거나, 혹은 둘의 조합으로 구성되어 있더라도 Jini에서의 서비스 접근 방식과 사용방식은 항상 동일하다. Jini는 룩업서비스(LookUp Service: LUS)를 중심으로하여 Discovery, Lookup, Remote Event, Leasing, Transaction의 서비스를 제공하고 이를 통해 네트워크에서 Jini 기술을 적용한 기기들이 서로 연결되어 Jini 커뮤니티에 등록하고 서로 자원을 공유하는 수단을 제공하게 된다[3]. Jini는 자바 언어에 기반을 둔 기술이지만, 이상적인 분산 컴퓨팅 환경을 위한 기술을 포함하고 있다.

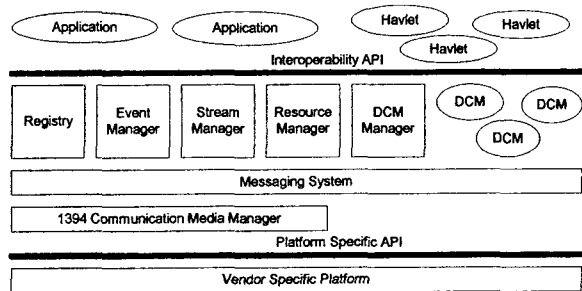
지니 서비스 매커니즘은 우선 지니 서비스가 룩업 서비스를 찾아 자신의 서비스 프락시 객체(Service Proxy Object)를 룩업 서비스에 등록하고, 서비스를 제공 받고자 하는 지니 클라이언트(Jini Client)는 자신이 원하는 서비스가 룩업 서비스에 존재하는지 검색한다. 검색의 결과로 지니 클라이언트는 서비스 프락시 객체를 가져오게 되며 이를 위해 룩업 서비스에서는 지니 서비스와 지니 클라이언트의 통신을 위해 Discovery, Join, Registration, Lookup를 제공한다. 서비스 프락시 객체가 지니 클라이언트에게 전달되면 지니 클라이언트와 서비스 사이의 통신은 룩업 서비스를 거치지 않고 직접 연결되어 서비스가 이루어진다[4].

## 2.2 HAVi의 개요

HAVi[7]는 가정에 있는 네트워크를 통해 연결된 다양한 제조사와 상표의 디지털 오디오와 비디오 장치 간의 상호 운용성(Interoperability)을 제공해주는 디지털 오디오 및 비디오 관련 전자 제품의 유연한 연동을 지원하기 위한 미들웨어에 대한 산업 표준의 하나이다. 이는 Sony, Philips등의 회사들이 주축이 되어 홈 네트워크 제품을 만들기 위해 개발한 미들웨어 소프트웨어 구조로 독자적인 프로토콜과 API를 가지고 있고 IEEE1394[5, 6]를 기반으로 한다. HAVi 구조에서는 서비스들이 소프트웨어 요소(Software Element)라 불리는 객체로 모델링 된다. 모든 소프트웨어 요소는 80bit의 SEID(Software Element Identifier)로 접근되며 네이밍 서비스인 Registry 서비스를 이용해 다른 객체를 찾을 수 있다. 또한 모든 객체는 메시지 전달을 통해 통신을 하며, 목적 객체(target object)는 SEID에 의해 결정된다. 모든 소프트웨어 요소는 상호 운용 가능한 API를 제공하며, 이 API를 이용하여 홈 네트워크 상에서 분산 어플리케이션을 구현할 수 있다. HAVi 디바이스 사이의 상호 운용성을 제공하기 위해 필요한 소프트웨어 요소에는 메시징 시스템, 레지스트리, 이벤트 관리자, 자원 관리자, DCM(Device Control Module) 관리자 등이 있다.

그림 1은 HAVi의 시스템 구조에 관한 것으로, 1394CMM(Communication Media Manager)는 다른 소프트웨어 요소들이 IEEE1394를 통하여 비동기 통신과 동시성 통신이 가능하게 하며 메시징 시스템(Messaging System)은 소프트웨어 요소들 간의 메시지 전달을 담당한다. 이벤트 관리자(Event

Manager)는 이벤트 전달 서비스를 제공하고 여기서의 이벤트는 홈 네트워크나 오브젝트의 상태 변화를 의미한다. 스트림 관리자(Stream Manager)는 각 컴포넌트 간의 AV스트림의 실시간 전송을 관리하며 레지스트리(Registry)는 홈 네트워크상의 다른 오브젝트를 찾는 디렉토리 서비스를 제공한다. DCM(Device Control Module)은 디바이스를 제어하는 소프트웨어 요소이며 DCM 관리자(DCM Manager)는 디바이스에 DCM코드를 설치하고 제거하는 기능을 한다.



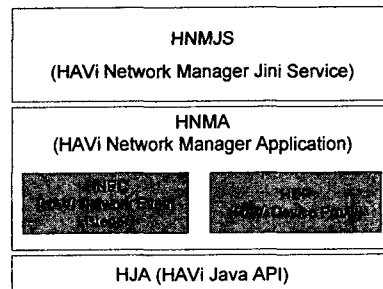
<그림 1> HAVi 소프트웨어 구조

## 3. HAVi 네트워크의 제어를 위한 Jini 서비스

### 3.1 제안하는 Jini서비스의 구조

제안하는 Jini서비스는 크게 HAVi Network Manager Application(HNMA)와 Jini 네트워크에서 서비스 프락시 객체로 사용될 HAVi Network Manager Jini Service(HNMJS)로 구분된다. HNMA는 일종의 HAVi 어플리케이션으로 HAVi 미들웨어의 여러 시스템 소프트웨어 요소(System Software Element) - Registry, Event Manager, DCM Manager, Stream Manager, Resource Manager, Messaging System - 등에 접근하기 위해 HAVi 컨소시엄에서 제안된 HAVi Java API(HJA)를 이용한다. HNMA는 HNMJS에서 HAVi 네트워크를 관리하고 DCM들을 제어하기 위해 사용할 여러 기능을 정의하고 HAVi 미들웨어와 Jini 서비스를 연결해주는 역할을 수행하며 HNMJS는 Jini 서비스 객체로 HAVi 측의 HNMA를 이용하여 HAVi 네트워크를 관리하게 된다.

다음 그림은 제안하는 Jini 서비스의 구조를 나타낸다.



<그림 2> 제안하는 Jini 서비스 구조

HNMA는 HAVi 네트워크에서 발생하는 각종 이벤트를 수신하기 위해 HAVi Network Event Listener(HNEL)가 요구된다. HNEL은 Jini서비스에서 필요로 하는 네트워크 이벤트를 지정하여, 특정 이벤트가 HAVi의 Event Manager에 수신되었을 때 HNEL에 전달되도록 자신을 등록하고 이 HNEL은 수신된 이벤트를 Jini 서비스에게 전달한다. 이때 HNEL은 Jini서비스가 알 수 있도록 이벤트를 객체화하여 전달한다.

다음으로, HNMA는 HAVi에서 사용하는 DCM을 사용할 수 없기 때문에 HAVi Device Proxy(HDP)를 이용한다. HAVi에서는 각 기기마다 DCM이 제공된다. 이 DCM은 디바이스 자신을 제어할 수 있는 모듈 또는 어플리케이션으로 DCM Manager에 의해 설치, 관리된다. 하지만, 이러한 DCM을 Jini서비스에서는 직접 사용할 수 없기 때문에, IEEE1394의 어플리케이션 프로토콜로 정의되는 각종 프로토콜을 이용하여 직접 제어를 하게 된다. 이러한 방식은 각 기기별로 제공되는 다양한 기능을 모두 이용할 수 없게 되지만, 기본적으로 각 기기가 제공해야 하는 기능을 사용할 수 있다. 이를 위해서 HAVi 네트워크에 연결된 기기의 GUID(Globally Unique Identifier)를 기반으로 기기의 종류를 분류해 낼 수 있는 HAVi Device Classifier(HDC)가 필요하다. HDC는 GUID에서 디바이스의 종류와 제조사를 알아내어 필요한 IEEE1394 어플리케이션 프로토콜이 어떤 것인지 파악해 낼 수 있다.

### 3.2 HAVi 네트워크 관리 서비스 시나리오

HAVi 네트워크 관리 서비스는 HAVi 네트워크와 Jini 네트워크에 동시에 참여할 수 있는 홈서버나 PC등에서 제공될 수 있다. 우선 HNMJS는 Jini 네트워크에서 룩업 서비스(LookUp Service)를 찾아 자신의 프락시 객체(Proxy Object)를 룩업 서비스에 등록한다. 이렇게 등록된 프락시 객체는 자신이 HAVi 네트워크를 관리할 수 있다는 정보를 지니게 되고, 이러한 서비스를 원하는 Jini Client는 룩업 서비스를 통해 검색한다. 검색의 결과로 서비스를 찾은 Jini Client는 이 프락시 객체를 가져오게 된다. 이러한 서비스를 제공하기 위해 룩업 서비스는 Discovery, Join, Registration, LookUp 메커니즘을 제공한다. 프락시 객체를 가져온 Jini Client는 룩업 서비스없이 직접 HNMJS와 연결하여 통신이 이루어 지게 된다. 프락시 객체는 HAVi 네트워크를 관리하고 기기를 제어할 수 있는 여러 서비스가 제공한다. 이 서비스는 HNMJS를 통해 HNMA에서 이루어진다.

우선, HAVi 네트워크의 구조를 알기 위해 HNMJS는 HNMA를 통해 GUID 맵과 SelfID 맵을 받아오게 된다. 이러한 정보를 이용하여 HNMJS에서는 어떠한 기기가 연결되어 있는지 알 수 있게 된다. 또한 HNMJS에서는 HNMA에서 제공되는 HJA의 Registry 관련 API를 이용하여 원하는 기기를 제어할 소프트웨어 모듈을 검색할 수 있으며, HAVi FCM을 이용한 제어가 가능할 경우에는 HNMA에서

제공하는 FCM관련 API를 이용하여 제어할 수 있다.

### 4. 구현 및 고찰

제안된 HAVi 미들웨어와의 연동을 위한 Jini 서비스는 Qplus 운영체제를 사용하는 ETRI 홈 서버에서 구현되었다. HAVi 미들웨어는 Java언어로 구현되었으며, HJA는 HAVi 미들웨어 위에서 여러 소프트웨어 요소와 리스너 객체를 관리한다. 홈 네트워크 상에서 HAVi 네트워크는 디지털 VCR, 1394 카메라, 캠코더로 구성되었으며, Jini 네트워크는 PC, PDA 등으로 구성되었다. HAVi 미들웨어와의 연동을 위한 Jini서비스를 구현하기 위해, HJA를 이용하는 HAVi 어플리케이션을 구현하였고, 이 어플리케이션을 이용하는 Jini 서비스 객체(HNMJS)를 구현하였다. HNMA에서는 HNEL이 HJA를 이용하여 리스너 객체가 추가되었다. 이 서비스를 이용하여 Jini 네트워크 상의 정보 기기에서 HAVi 네트워크의 상태를 알 수 있으며, 연결된 기기에 대한 정보를 얻고 제어를 할 수 있다.

### 5. 결론

초고속 통신망의 급속한 보급과 가전기기의 디지털화 고급화 추세에 힘입어 최근 홈 네트워크를 구성할 수 있는 기반이 다져지고 있다. Jini는 홈 데이터 네트워크 미들웨어로 자바언어를 기반으로 플랫폼 독립적인 특성을 가지며, 코드의 이동성을 제공하기 때문에 다른 미들웨어에 비해 진보적인 특징을 지니고 있다. 이러한 Jini 미들웨어 기술을 바탕으로 홈 엔터테인먼트 네트워크에 적용되는 HAVi 미들웨어와 연동이 가능한 Jini 서비스를 구현함으로써 Jini 미들웨어 기술을 기반으로 LonWorks나 HAVi와 같은 다른 미들웨어의 연동이 가능함을 보였다.

### [참고문헌]

- [1] Sun Microsystems, "Jini Architecture Specification." [http://www.sun.com/jini/specs/jini1\\_1.pdf](http://www.sun.com/jini/specs/jini1_1.pdf)
- [2] Sun Microsystems, Jini Technology Core Platform Specification. [http://www.sun.com/jini/specs/core1\\_1.pdf](http://www.sun.com/jini/specs/core1_1.pdf)
- [3] W. Keith Edwards, "Core Jini" 2nd Edition, Prentice Hall, 2001
- [4] Scott Oaks & Henry Wong, "Jini In a nutshell", O'Reilly, 2000
- [5] IEEE1394, Std for High Performance Serial Bus, 1995.
- [6] Don Anderson, "FireWire System Architecture" 2nd Edition, Addison Wesley, 1999
- [7] Specification of the Home Audio/Video Interoperability (HAVi) Architecture, Version 1.1, May 15, 2001.
- [8] K. Arnold et al., The Jini™ Specification, Addison-Wesley Longman, Reading, Mass 1999.