Implementation of Jini Service supporting interoperable with HAVI Home Network

Dong-Hwan Park\(^1\) Tai-Yeon Ku Kyeong-Deok Moon
Dept. of Office, Electronics & Telecommunication Research Institute

1. 서론


2. 관련 연구

2.1 Jini의 개요

Jini는 Java를 기반으로 네트워크 상의 기기나 소프트웨어를 동적으로 생성하고 심화성 있는 방식으로 네트워크에 연결된 각종 서비스를 동적으로 연계하기와 같은 모듈을 가지고 있다. 먼저 서비스를 구성하는 참여자들이 서로에 대한 접근이 없이 Jini 네트워크에 연결되어 있으며, 이러한 설정 작업은 요구되지 않고도 서비스를 인식하여 연결되고 서비스를 제공할 수 있도록 해준다. Jini에서 사용되는 서비스는 하드웨어와 소프트웨어의 구별이 존재하지 않는다.
네트워크상 존재하는 서비스는 이것이 하드웨어로 구현되어 있거나, 소프트웨어로 구현되어 있거나, 혹은 두 가지 형태로 구성되어 있다. Jini 서비스는 이러한 서비스를 제공하고 이를 통해 네트워크에서 Jini 기술을 적용한 기기들이 서로 연계되어 Jini 커뮤니티에 등록하고 서로 사용할 수 있게 하기 위해 제공된다. Jini는 각 기기 안에 기반을 둔 기기지만, 이상적인 분산 컴퓨팅 환경을 위한 기술을 포함하고 있다.

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

2.2 HAVI의 개요

HAVI[5]는 지연에 있는 네트워크를 통해 연결된 다양한 제조품의 상호작용의 디지털 오디오와 비디오 장치 간의 상호운용성(Interoperability)을 제공해주는 디지털 오디오 및 비디오 관련 제품의 유효한 연결을 지원하기 위해 이들 사이의 공용 표준의 하나이다. 이는 Sony, Philips등의 회사들이 주도하여 공 밖 네트워크 제품을 만들기 위해 개발한 HAVI 프로토콜은 고급 스마트공구를 사용한 프로토콜을 API로 가지고 있고 IEEE 1394[5, 6]을 기반으로 한다. HAVI 구조는 이러한 서비스들이 소프트웨어 요소(Software Element)가 스택의 각계로 배열된다. 모든 소프트웨어 요소는 80비트의 SEID(Software Element Identifier)로 정의되며 네트워크인 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 API로 제공하며, 이 API를 이용하여 홈 네트워크상에서 분산 어플리케이션을 구현할 수 있다. HAVI 메시지 서비스의 상호운용성을 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 Manager)는 다른 소프트웨어 요소들이 IEEE1394를 통하여 비동기 통신과 동시에 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 컨포넌트 간의 AV스트림의 시간적 연속성을 관리하며 레지스트리(Registry)는 홈 네트워크와 다른 오브젝트를 찾는 데로써 서비스를 제공한다. DCM(Device Control Module)는 디바이스를 관리하는 소프트웨어 요소이며 DCM 관리자(DCM Manager)는 디바이스의 DCM코드를 설치하고 제어하는 기능을 한다.

3. HAVI 네트워크의 제어를 위한 Jini 서비스

3.1 제어하는 Jini서비스의 구조

제어하는 Jini서비스는 크게 HAVI Network Manager Application(HNMA)과 Jini 네트워크에서 서비스 프락시 객체로 사용될 HAVI Network Manager Jini Service(HNMS)로 구분된다. HNMA는 일부의 HAVI 어플리케이션으로 HAVI 미들웨어의 여러 시스템의 소프트웨어 요소(System Software Element) - Registry, Event Manager, DCM Manager, Stream Manager, Resource Manager, Messaging System - 등에 접근하기 위해 HAVI 천사소재에 제안된 HAVI Java API(HJIA)를 이용한다. HNMA는 HNMS에서 HAVI 네트워크를 관리하고 DCM들을 제어하기 위해 사용할 여러 기능을 정의하고 HAVI 미들웨어와 Jini 서비스를 연결해주는 역할을 수행하며 HNMS는 Jini 서비스의 객체로 HAVI 측의 HNMA를 이용하여 HAVI 네트워크를 관리하게 된다.

다음 그림은 제어하는 Jini 서비스의 구조를 나타낸다.
HNMA는 HAVI 네트워크에서 발생하는 각종 이벤트를 수신하기 위해 HAVI Network Event Listener(HNEL)를 요구한다. HNEL은 Jini서비스를 필요로 하는 네트워크 이벤트를 지정하여, 특정 이벤트가 HAVI의 Event Manager에 수신되었을 때 HNEL에 전달되도록 자동으로 동작하고 이 HNEL은 수신된 이벤트를 Jini 서비스에게 전달한다. 이러한 HNEL은 Jini 서비스가 알 수 있도록 이벤트를 직접하여 전달한다.

다음으로, HNMA는 HAVI에서 사용하는 DCM을 사용할 수 없기 때문에 HAVI Device Proxy(HDP)를 이용한다. HAVI에서 제공하는 DCM의 재현될 수 있는 모듈 또는 플랫폼이며, DCM을 이용하여 DCM Manager에 의해 설치, 관리된다. DCM은 다이나믹으로 자동을 시도할 수 있는 모듈 또는 플랫폼이며, DCM Manager 특성으로 자동을 시도할 수 없기 때문에, IEEE1394의 어플리케이션 프로토콜로 정의되는 각종 프로토콜을 이용하여 직접 제어를 하게 한다. 이러한 방식은 제어기별로 제공되는 다양한 기능을 모두 이용할 수 없게 되지만, 기본적으로 제어기 자체에 저장된 기능을 사용할 수 있다. 이를 위해서 HAVI 네트워크에 연결된 제어기의 GUID(Globally Unique Identifier)를 기반으로 기기의 종류를 분류해 볼 수 있는 HAVI Device Classifier(HDC)가 필요하다. HDC는 GUIDE에서 다이나믹으로 제어를 할 수 있는 IEEE1394 어플리케이션 프로토콜이 어떤 것인지 파악해 볼 수 있다.

3.2 HAVI 네트워크 관리 서비스 시나리오

HAVI 네트워크 관리 서비스는 HAVI 네트워크와 Jini 네트워크에 동시에 참여할 수 있는 홍서바나 PC등에서 제공될 수 있다. 우선 HNMS는 Jini 네트워크에서 채널 서비스(lookUp Service)를 찾아 자신의 프로토시 객체(Proxy Object)를 채널 서비스에 등록한다. 이렇게 등록된 프로토시 객체는 자동으로 HAVI 네트워크에 관리할 수 있는 정보를 지정하고, 이러한 서비스를 사용하는 Jini Client는 채널 서비스를 통해 관리한다. 선택의 결과로 서비스를 찾은 Jini Client는 이 프로토시 객체를 가져오게 된다. 이러한 서비스를 제공하기 위해 채널 서비스는 Discovery, Join, Registration, LookUp 메커니즘을 제공한다. 프로토시 객체는 Jini Client는 채널 서비스없이 직접 HNMS와 연결하여 통신이 이루어지게 된다. 프로토시 객체는 HAVI 네트워크를 관리하고 기기를 제어할 수 있는 여러 서비스를 제공한다. 이 서비스는 HNMS를 통해 HAVI에서 제공된다. 서비스의 구조를 알기 위해 HNMS는 HAVI를 통해 GUI/GM 및 SDFB를 받아온다. 이러한 정보를 이용하여 HNMS는 서비스의 구조가 연결되어 있는지 알 수 있게 된다. 또한 HNMS는 HAVI에서 제공되는 HAVI의 Registry 관련 API를 이용하여 원하는 기기를 제어할 소프트웨어 모듈을 검색할 수 있으며, HAVI FCM을 이용한 제어가 가능할 경우에는 HNMS에서 제공하는 FCM 관련 API를 이용하여 제어할 수 있다.

4. 구현 및 고찰


5. 결론

조금은 통신망의 복잡한 보급과 가전기기의 디지털화 속에 이어져 최근 홈 네트워크의 구현이 지속적으로 진행되고 있는 추세이다. 이러한 Jini 어플리케이션을 이용한 프로토콜, Jini 서비스의 등록과 사용을 위한 HAVI 어플리케이션과의 연동을 위해 Jini 서비스를 구현함으로써 Jini 어플리케이션을 이용한 Jini 서비스에서 홈 어플리케이션의 연동을 가능할 것이다.

[참고문헌]