

특

다중 프로토콜을 지원하는 홈 네트워크 디바이스 간의 서비스 바인딩 구조

박준호* 배대호** 강순주***

목 차

1. 서 론
2. 관련 연구 및 요구 사항
3. 제안하는 홈 네트워크 하드웨어 및 소프트웨어 구조
4. 홈 네트워크 디바이스의 네트워크 객체화 및 바인딩 서비스
5. 홈 네트워크 적용 사례
6. 결 론

1. 서 론

홈 네트워크는 다양한 종류의 가전기기들과 다양한 네트워크가 혼재해 있는 유비쿼터스 네트워크 환경[13]이다. 홈 네트워크를 구성하는 가전기기들은 기능과 특징에 따라 크게 멀티미디어 가전기기, 단순 제어 가전기기로 구분할 수 있다. 멀티미디어 가전기기의 경우 영상 및 음성 정보를 전송하기 위해 큰 대역폭을 가지는 하부 네트워크를 이용하고 있으며 영상 및 음성 정보의 전송과 제어를 위해 많은 제어 모듈을 가지고 있다. 반면 단순 제어 가전기기들은 제어 및 상태 정보의 신뢰성 있는 전송을 위한 하부 네트워크를 사용하고 있으며 단순한 제어 모듈을 가지고 있다. 이와 같이 다양한 특징을 가지는 가전기기들이 홈 네트워크에서 상호 연동하기 위해서는 모든 가전기기 정보를 공통으로 정의하고 표준화된 정보 및 제어명령 전달 인터페이스를 정의하는 것이 필수적이다[9,12]. 홈 네트워크를 위한 많은 미들웨어들이 디바이스들을 제어하고 정보를 감시하기 위해 제안되었다. 특히

멀티미디어 홈 네트워크 구축을 위해 IEEE1394 [5]를 기반으로 제안된HAVi[1], 단순 가전기기의 제어 및 감시를 위해 제안된 UPnP[2] 미들웨어가 현재 주목을 받고 있다. 각 미들웨어는 가전기기 정보를 나타내기 위해 고유의 다른 가전기기 정보의 표현 방법을 정의하고 있지만 표준화되어 있지 않고 특정 서비스를 기반으로 설계되어 홈 네트워크를 구성하는 모든 디바이스 간의 연동이 어렵게 설계되어 있다. 예를 들어 HAVi의 경우 디바이스 종류마다 고유한 코드를 부여하고 디바이스를 제어하기 위한 각 기능에 고유의 코드를 부여하고 있다. 하지만 이는 IEEE1394[4]를 기반으로 하는 멀티미디어 가전기기에 한정되어 있으며 단순히 디바이스를 제어하기 위한 코드의 열거로 되어 있다. 또한 UPnP의 경우 홈 네트워크에서 사용되는 디바이스 정보를 표현하기 위한 일반화된 형식은 제공하고 있지만 실질적인 제어나 정보의 전달을 위해 각 디바이스의 기능을 세부적으로 정의하고 있지 않다. 또한 홈 네트워크를 구성하는 가전기기들은 단순히 하나의 가전기기를 제어하는 것 이상으로 여러 가전기기들이 상호 운용되어 동작하는 복잡한 네트워크다. 멀티미디어 가전기기의 경우 하

* 경북대학교 전자공학과 박사과정

** 경북대학교 전자공학과 석사과정

*** 경북대학교 전자전기공학부 정보통신전공 부교수

나의 영상 정보가 다중의 스크린에 전송되어 방송될 수 있으며, 여러 단순 가전기기들이 상호 연동하여 하나의 작업을 수행할 수도 있다. 하지만 기존의 미들웨어들은 디바이스 상태 정보 정의에서 이러한 가전기기 간의 상호 연동을 위한 상태 정보의 정의 및 제어는 전혀 고려하지 않고 있다.

본 논문에서는 홈 네트워크 환경에서 기존의 미들웨어들이 가지고 있는 가전기기 정보의 표현 방법과 상호 연동의 문제점을 해결하기 위해 디바이스 상태 정보의 정의방법을 제안하고 상태 정보를 전달하기 위한 미들웨어 구조를 제안할 것이다.

제안하는 디바이스 정의 구조는 홈 네트워크의 모든 가전기기들의 정보를 표현하기 위해 객체지향 방법론을 바탕으로 단순하며 일관성 있는 구조를 가지도록 설계되었으며, 새로운 상태 정보의 추가, 삭제, 갱신 등과 같은 디바이스 상태 정보 변경 및 제어 정보의 전달을 유연하게 표현할 수 있도록 설계되었다. 정의된 디바이스의 상태 및 제어 정보 전달은 다양한 하부 네트워크 프로토콜을 연동할 수 있는 미들웨어를 바탕으로 디바이스, 에이전트 및 미들웨어 소프트웨어 컴포넌트 사이의 통신을 위해 이벤트 서비스를 이용한다. 마지막으로 홈 네트워크 사용자들이 원하는 서비스를 더욱 편리하게 구성할 수 있도록, 홈 네트워크 디바이스들이 제공하는 서비스들을 상호 연결하기 위한 바인딩을 지원하는 구조를 제안한다.

본 논문은 다음과 같이 구성된다. 2장에서는 기존에 제안된 미들웨어들을 살펴보고 이를 바탕으로 홈 네트워크의 디바이스들을 표현하고 상태 정보를 전달하기 위한 요구사항을 살펴볼 것이다. 3장에서는 제안하는 홈 네트워크의 하드웨어 및 소프트웨어 구조를 개략적으로 살펴보고, 4장에서는 디바이스의 네트워크 객체화 및 바인딩 서비스에 대해 상세하게 살펴볼 것이다. 5장에서는 제안하는 미들웨어를 실제의 홈 네트워크 환경에 적용한 사

례에 대해 살펴보고, 6장에서 결론을 맺도록 한다.

2. 관련 연구 및 요구 사항

2.1 관련 연구

홈 네트워크 환경을 위해 많은 미들웨어들이 제안되었다. 제안된 미들웨어 중에 주목을 받고 있는 것은 HAVi, UPnP, Jini[3,4] 등이 있다.

HVAi는 IEEE1394를 기반 네트워크로 사용하며 멀티미디어 가전기기를 위해 제안된 미들웨어다. HAVi는 디바이스 상태 정보를 관리하기 위해 디렉토리 서비스를 제공하기 위한 저장소(Registry) 서비스와 자원의 공유를 위한 자원 관리자(Resource Manger)를 정의하고 있으며, 각 디바이스 기능을 제어하기 위한 인터페이스를 제공하는 DCM(Device Control Module)을 정의하고 있다. 또한 각 소프트웨어 요소들이 상호 통신하기 위해 메시징 시스템을 정의하고 있으며 각 객체의 변화를 전달하는 이벤트 서비스를 정의하고 있다. HAVi는 미들웨어의 각 소프트웨어 모듈들이 서로 정보를 주고 받기 위해 이벤트 타입, 디바이스 타입, 및 제어 타입을 미리 정의하고 있다. 그러나 HAVi는 디바이스의 타입과 API는 미리 정해진 특정 코드를 가지고 있어 새로운 기능의 추가나 갱신에 유동적으로 대체할 수 없다는 문제점을 가지고 있으며, 각 디바이스 종류마다 방대한 디바이스 코드, 기능(Operation) 코드, 속성 코드와 이벤트 타입을 정의하고 있어 디바이스를 원격에서 관리하는 에이전트 구현에 어려움을 가질 수 있다. 또한 HAVi는 기본적으로 멀티미디어 디바이스를 기반으로 설계되어 다른 종류의 가전기기에 대해서는 전혀 고려하지 않고 있다.

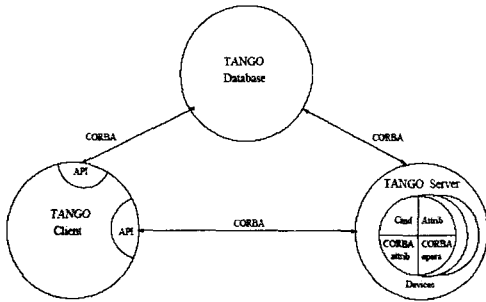
UPnP는 Microsoft에 의해 제안되었으며 가정과 사무실 네트워크에서 기기들의 상호 연동을 위한 TCP/IP 기반의 미들웨어다. UPnP는 홈 네트워크

의 관리 및 감시를 위해 디바이스의 등록, 다른 디바이스의 제어 및 상태 정보의 획득 방법, 통신 방법 등에 대해 정의하고 있다. 구체적인 방법으로 DHCP(Dynamic Host Configuration Protocol)을 이용한 주소할당, SSDP(Simple Service Discovery Protocol)을 이용한 디바이스 디스커버리(Discovery), SOAP(Simple Object Access Protocol)을 이용한 디바이스 컨트롤, GENA(General Event Notification Architecture)를 이용한 이벤트 전달 방법 등을 정의하고 있다. 하지만 각 과정에서 사용되는 디바이스의 정보 기술 방법은 XML을 이용한 자율적인 디바이스 표현 방법을 취하고 있어 광범위한 디바이스 정보의 표현은 보장해 주지만 동일한 디바이스를 표현함에도 여러 가지 다른 표현 방법이 나올 수 있어 통일성을 제공하지 못한다. 또한 각 디바이스의 상태 변경으로 발생하는 이벤트 정보를 수신하기 위해서 각 디바이스마다 따로 수신자를 등록하게 되어 있어 여러 디바이스가 상호 연동하여 복잡한 기능을 수행하기에는 문제가 있다.

Jini 또한 가정과 사무실 네트워크를 위해 제안된 미들웨어로서 중앙 집중식 관리 구조를 가지고 있다. Jini는 서비스의 등록과 검색을 위한 인터페이스를 정의하고 있지만, 서비스의 등록이 디바이스의 세부적인 기능 정보를 저장하고 있는 것이 아니라 간략한 디바이스의 설명을 저장하고 있으며 실질적인 제어 코드는 다운로드 되어 동작하도록 되어 있다. 이러한 접근 방법은 디바이스를 사용하기 위한 코드를 따로 설치할 필요가 없어 편리한 사용을 보장해 주지만 디바이스의 상세한 정보를 검색하기는 어려우며 디바이스 간의 상호 연동은 불가능하다. 또한 디바이스 정보의 기술 방법은 표준화되어 있지 않으므로 홈 네트워크에서 다양한 서비스를 제공하는 에이전트들은 디바이스 검색에 어려움을 가질 수 있다.

제어 시스템을 위한 디바이스 정보의 추상화 및 제어 인터페이스를 제공하는 미들웨어로 TANGO [8]가 있다. TANGO는CORBA[14]를 기반으로 설계되었으며 분산 제어 시스템 환경에서 일관된 디바이스를 읽고 쓰는 접근방법을 제시하고 있다. TANGO에서 각 디바이스는 하나의 CORBA 네트워크 객체로 표현되며 동기 및 비동기 통신을 이용한 이벤트 통신 방법을 제공하고 있다. 또한 디바이스를 표현하는 방법이 단순하며 모든 제어 디바이스를 표현할 수 있는 공통적인 방법을 제시하고 있다. CORBA 객체로 표현되는 모든 디바이스 형태는 TANGO에서 사용하는 모든 디바이스에 대하여 동일하며 CORBA로 표현된 디바이스 객체를 상호 운용을 위한 통신 프로토콜로 사용한다. (그림 1)은 TANGO의 기본적인 구성 요소인 TANGO 서버, TANGO 데이터 베이스, TANGO 클라이언트를 나타낸다. TANGO 서버는 디바이스를 CORBA 네트워크 객체 형태로 속성(Attribute)과 명령(Command)으로 표현한다. 또한 CORBA 인터페이스를 이용하여 TANGO 데이터베이스에 디바이스에 대한 정보를 저장하게 된다. TANGO 클라이언트는 TANGO 데이터베이스로부터 TANGO 서버에 대한 정보를 가져와서 TANGO 서버의 디바이스에 대한 제어를 수행한다. TANGO의 디바이스 정보 관리 구조는 정보를 저장하는 저장소와 실질적인 디바이스 제어코드를 분리하는 구조를 가지며 TANGO 클라이언트들이 쉽게 디바이스 정보를 검색할 수 있도록 한다. TANGO는 디바이스를 통일된 구조를 가지는 네트워크 객체 형태로 표현하고 디바이스 제어를 위한 통일된 인터페이스를 제공함으로써 클라이언트 및 서버 프로그램의 작성의 용이함과 실제적인 디바이스 제어의 편리함을 제공하고 있다.

본 논문에서는 TANGO의 기본 개념인 디바이스를 네트워크 객체로 표현하는 방법 및 상호 통신



(그림 1) TANGO 구조도

의 주체로 사용하는 방법을 홈 네트워크에 적용하여 볼 것이다.

2.2 요구사항

앞서 살펴본 홈 네트워크의 미들웨어들은 디바이스 관리 및 감시를 위해 디바이스 구조를 정의하고 있다. 그러나 각각의 미들웨어는 홈 네트워크에서 사용하는 모든 디바이스에 대한 일반화된 형태를 제공하지 못하고 있다. 그리고 홈 네트워크에서 제공하는 서비스는 디바이스 하나만을 제어하는 단순한 서비스로부터, 여러 가지 디바이스가 제공하는 서비스들을 연결하여 제공할 수 있는 상위의 복잡한 서비스가 존재한다. 이러한 시도는 UPnP에서 멀티미디어 가전기기를 연결하기 위한 표준을 제안하는 등 최근 활발히 진행되고 있으나 아직까지 정확한 표준은 없는 상태다.

홈 네트워크는 다양한 디바이스와 디바이스가 제공하는 서비스로 구성되고 사용자들은 이러한 서비스를 편리하게 사용하기를 원한다. 따라서 홈 네트워크를 관리하는 미들웨어는 디바이스 자원을 편리하게 관리할 수 있는 서비스를 제공하고 홈 네트워크의 디바이스를 제작하는 개발자들을 위해 확장하기 쉬우며 단순하고 일관성 있는 인터페이스를 제공할 필요가 있다. 이러한 홈 네트워크에서 디바이스 관리 미들웨어의 요구사항을 살펴보면 다음과 같다.

2.2.1 통일된 디바이스 표현 방법

디바이스 표현 방법은 홈 네트워크를 구성하는 디바이스를 다른 디바이스나 에이전트에게 전달하기 위해 어떻게 표현할 것인지를 정의하는 것이다. 디바이스를 표현 하는 방법은 홈 네트워크에서 동작하는 모든 소프트웨어 객체들이 상호 통신하기 위해 다양한 디바이스들을 모두 포함할 수 있도록 일관성 있는 표현 방법을 가져야 할 것이며 개발자들의 편의를 위해 단순한 형태를 가지는 네트워크 객체 구조를 가져야 할 것이다.

2.2.2 통일된 디바이스의 통신 방법

홈 네트워크에서 사용하는 하드웨어적인 프로토콜은 IEEE1394, LonWorks[4.6], TCP/IP, Bluetooth[4.7] 등과 같이 매우 다양하다. 이렇게 다양한 프로토콜들은 홈 네트워크에서 특정의 서비스를 제공하며, 단일의 프로토콜로써 홈 네트워크의 모든 서비스는 제공할 수 없다. 따라서 홈 네트워크는 다양한 프로토콜들이 사용될 것이며 다양한 하드웨어적 네트워크 프로토콜을 연동할 수 있는 미들웨어를 이용하는 방법을 고려하여야 한다. 따라서 홈 네트워크 미들웨어는 하부 프로토콜들에 대한 추상화를 통해 통일된 제공하며 상위의 소프트웨어 객체들(디바이스 제어 객체, 에이전트 객체, 미들웨어 객체)이 하부 프로토콜에 고려없이 상호 운용될 수 있는 소프트웨어 구조를 요구한다.

2.2.3 디바이스의 기능들을 상호 연결하기 위한 구조

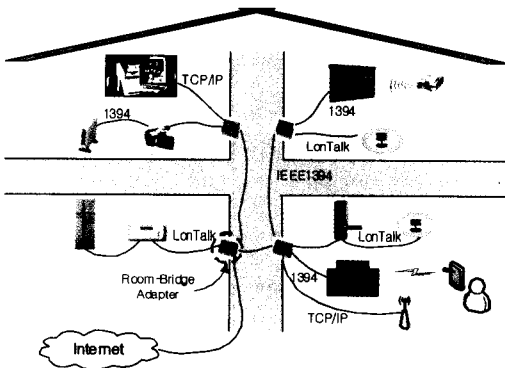
홈 네트워크의 서비스는 전등 스위치를 켜는 단순한 서비스부터, DVD 플레이어를 통해 TV로 영상을 오디오로 음성신호를 보내는 것과 같이 여러 개의 디바이스가 제공하는 서비스를 연결해야 하는 서비스가 존재한다. 이러한 디바이스 사이의 서비스를 서로 연결하고 연결된 서비스의 트랜잭션을 관리하는 구조가 필요하다. 또한 디바이스 사이의 연결 상태를 저장하고 관리할 수 있는 공간이 필요

하다. 여러 디바이스를 서로 연결하는 서비스를 본 논문에서는 바인딩이라 정의한다.

3. 제안하는 홈 네트워크 하드웨어 및 소프트웨어 구조

3.1 방 단위의 물리적 홈 네트워크 구조

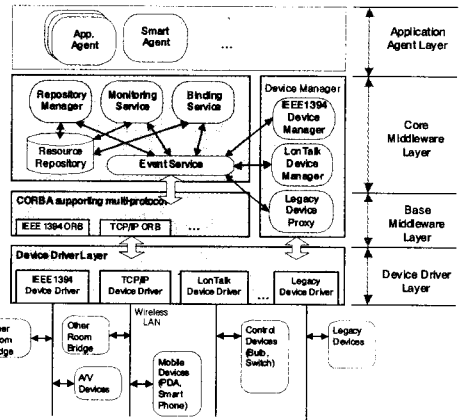
본 논문에서 고려한 홈 네트워크의 하드웨어 구성도는 (그림 2)와 같다



(그림 2) 홈 네트워크 하드웨어 구성도

홈 네트워크 환경은 (그림 2)와 같이 방 단위로 다양한 디바이스들이 연결된 하드웨어 구조를 가진다. 각 방에는 룸 브릿지 어댑터가 방의 디바이스를 연결하는 브릿지 역할을 하여 다양한 프로토콜을 사용하는 디바이스들을 상호 연결한다. 단순 제어 디바이스들인 리모트 컨트롤러, 전등, 스위치 등은 LonTalk 프로토콜을 이용하여 룸 브릿지에 연결되며 멀티미디어 디바이스들은 IEEE1394 프로토콜을 이용하여 연결된다. 각 방의 룸 브릿지들은 백본 네트워크인 IEEE1394를 통해 상호 연결되며 사용자는 무선랜이나 Bluetooth 등의 무선 네트워크를 이용하여 홈 네트워크의 가전기기들을 제어할 수 있다.

3.2 소프트웨어 구조 개요



(그림 3) 제안하는 브릿지의 전체적인 소프트웨어 구조

(그림 3)에서 보이는 것처럼, 제안하는 미들웨어는 디바이스 드라이버, 미들웨어, 어플리케이션 계층으로 구성된다. 또한 미들웨어 계층은 크게 다섯 가지 소프트웨어 요소(자원 관리자, 감시 서비스, 이벤트 전송 서비스, 바인딩 서비스, 디바이스 관리자)로 구성된다.

3.2.1 디바이스 드라이버

디바이스 드라이버는 네트워크 상황을 감시하여 새로운 가전기기가 연결되거나 연결을 끊는지를 감시한다. 또한 가전기기의 상태 변경을 감시하여 상위 미들웨어 계층에 변화를 알리고 미들웨어 계층에서 전달되는 제어 명령을 가전기기에 직접적으로 전달하는 역할을 한다.

3.2.2 다중 프로토콜을 지원하는 CORBA

CORBA[14]는 프로그래밍 언어, 운영 체제, 및 프로토콜에 관계없이 서버와 클라이언트 사이의 상호운용을 지원하는 미들웨어다. 따라서 제안하는 미들웨어에서는 CORBA를 이벤트 전송 및 디바이스 상태 정보 전달을 위한 기반 미들웨어로 사용하고 있다. 홈 네트워크는 다양한 프로토콜을 사용하는 가전기기들로 구성되기 때문에 기반 미들웨어로 사용되는 CORBA는 다중 프로토콜을 지원할 필요가 있다[11]. 각 ORB는 디바이스 드라이버와

연결되어 CORBA 패킷을 하부 네트워크를 위한 패킷으로 변환한다. 예를 들어 CORBA의 TCP/IP를 위한 GIOP/IOP를 지원하는 TCP/IP ORB와 IEEE1394를 위한 GIOP/1394IOP를 지원하는 1394 ORB 계층을 지원하여 하부 네트워크를 위한 패킷으로 변경하는 역할을 수행한다.

3.2.3 자원 저장소 및 저장소 관리자

홈 네트워크의 서비스들은 각 서비스를 수행하기 위해 가전기기들의 상태 정보를 지속적으로 감시할 필요가 있다. 따라서 홈 네트워크의 상태 정보를 감시하고 관리하기 위해 제안하는 미들웨어는 자원 관리자를 제공한다. 자원 관리자는 홈 네트워크에 연결되어 있는 모든 가전기기들의 상태 정보를 관리하고, 어플리케이션이나 에이전트의 요청에 상태 정보를 전달하는 역할을 한다.

3.2.4 이벤트 서비스 및 감시 서비스

홈 네트워크에서 전송되는 메시지는 디바이스 상태 정보 및 제어 명령의 두 가지로 분류된다. 디바이스 상태 정보는 자원 관리자에 의해 관리되며 제어 명령과 디바이스 상태 정보의 변경과 같은 이벤트는 이벤트 서비스에 의해 전송된다. 제안하는 미들웨어에서 이벤트 서비스는 CORBA의 이벤트 서비스를 홈 네트워크 환경에 적합하게 확장하여 사용하였다. 감시 서비스는 홈 네트워크에 연결된 많은 디바이스들의 동작 유무를 감시한다. 각 디바이스들이 미들웨어에 서비스 중단을 알리지 않고 갑작스럽게 중단된다면 미들웨어는 이를 인지할 수 없다. 따라서 감시 서비스는 이러한 디바이스의 동작 유무를 지속적으로 감시하여 자원 관리자에 저장된 디바이스 상태 정보의 유효성을 보장하게 된다.

3.2.5 바인딩 서비스

바인딩 서비스는 서로 연결되어 서비스가 필요한 디바이스들을 연결하는 역할을 한다. 디바이스

의 연결 정보는 자원 저장소에 저장되며 바인딩 서비스는 디바이스 연결 및 해제 요청 이벤트와 바인딩 관련 이벤트를 처리한다. 바인딩 이벤트 처리과정에서 사용되는 디바이스 정보는 다른 서비스나 디바이스들이 변경하지 못하도록 하여 바인딩 처리과정의 트랜잭션을 보장한다.

3.2.6 디바이스 관리자

디바이스 관리자는 디바이스 드라이버를 통해 물리적인 디바이스를 직접적으로 제어하는 역할을 수행한다. 예를 들어 IEEE1394 디바이스 관리자는 하부에 연결된 모든 IEEE1394 디바이스를 관리하는 모듈이다. IEEE1394 및 LonWorks 네트워크는 모두 필드 버스 프로토콜이므로, 각 네트워크는 유일한 주소 체계와 IEEE1394의 경우 IEC61883, LonWorks의 경우 SNVT(Standard Network Variable Type)와 같은 표준화된 명령체계를 가지고 있다. 따라서 각 디바이스 관리자는 네트워크에 연결된 모든 디바이스를 관리할 수 있으며 디바이스의 추가와 제거 및 상태의 변경과 같은 변화를 자원 관리자에게 전달하여 자원 저장소에 반영되게 한다. 또한 다른 소프트웨어 모듈로부터 전달되는 제어 명령을 디바이스 드라이버를 통해 물리적인 디바이스에게 전달하여 실질적인 동작이 일어나게 한다.

(그림 3)의 소프트웨어 구조는 동일한 저자에 의해 작성된 참조 논문[10]에서 상세하게 설명되어 있으며 본 논문에서는 디바이스의 정보를 표현하기 위한 구조와 바인딩 서비스를 4장에서 상세하게 설명할 것이다.

4. 홈 네트워크 디바이스의 네트워크 객체화 및 바인딩 서비스

본 절에서는 홈 네트워크의 다양한 디바이스를 일관성 있게 표현하기 위한 디바이스의 네트워크

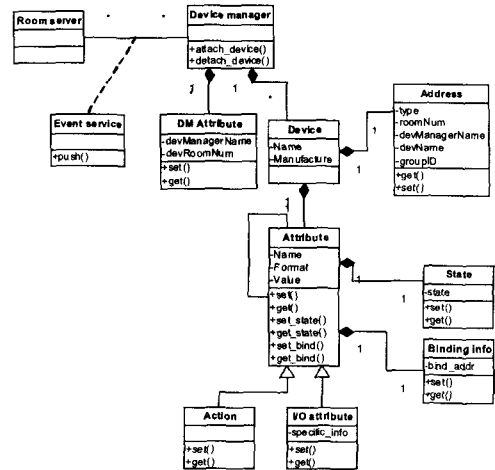
객체화 방법을 설명하고 각 소프트웨어 요소들 사이의 통신을 위한 이벤트 서비스에 대해 설명할 것이다. 또한 디바이스의 초기화, 추가, 제거, 상태 모니터링, 제어 등의 기본적 서비스를 제공할 수 있는 구조를 정의하고 마지막으로 각각의 디바이스의 서비스를 연결할 수 있는 소프트웨어 구조를 살펴볼 것이다.

4.1 홈 네트워크 디바이스들의 네트워크 객체화

가정에서 사용되고 있는 디바이스들은 서로 다른 기능들과 다른 특징을 가지고 있다. 이러한 이유로 홈 네트워크에는 많은 프로토콜들이 적용되고 있으며 각기 다른 서비스들을 제공하고 있다. 이러한 다양성에도 불구하고 홈 네트워크를 구성하기 위해서는 모든 디바이스를 표현하는 통일된 방법이 필요하며 디바이스의 기능들을 표현할 수 있어야 한다. 본 논문에서는 TANGO의 네트워크 객체 표현 방법과 LonWorks의 SNVT(Standard Network Variables Types)를 바탕으로 홈 네트워크 디바이스들의 표현 방법을 설계하였다.

(그림 4)는 제안하는 홈 네트워크 디바이스 형태를 객체 다이어그램으로 표현한 것이다. 디바이스 관리자 클래스는 동일한 하부 프로토콜을 관리하기 위한 클래스이며 프로토콜 특성을 표현하는 변수 및 하부 디바이스를 연결하기 위한 주소 체계 등을 표현하는 변수를 가지고 있다. 또한 하부 디바이스들을 관리를 위해 디바이스 리스트를 가지고 있다. 실질적인 홈 네트워크 디바이스를 표현하는 디바이스 클래스에는 디바이스의 입출력 속성 클래스와 액션 클래스를 가지고 있다. 액션 클래스는 사용자가 스위치를 누르는 것과 같은 행동을 추상화 한다. 이러한 액션 클래스는 다른 디바이스의 액션 클래스 또는 디바이스 내부의 입출력 속성과 연결 관계가 있을 수 있으며, 이러한 관계는 룸 브릿지의 바인딩 서비스를 이용하여 생성한다. 이러

한 바인딩 정보는 액션 클래스와 입출력 속성에 포함 되어 있는 바인딩 클래스에 보관된다. 디바이스의 액션과 입출력 속성은 디바이스가 초기화 되고 홈 네트워크에 추가될 때 이벤트 서비스를 이용하여 룸 브릿지로 전달된다.



(그림 4) 홈 네트워크 디바이스의 네트워크 객체화 클래스 다이어그램

홈 네트워크를 구성하는 디바이스들은 (그림 4)의 클래스 다이어그램에서 표현된 방법을 통해 모두 표현될 수 있으며 소프트웨어 모듈 사이의 상호 통신을 위해 사용될 수 있다.

4.2 소프트웨어 모듈 간의 통신 방법

(그림 3)의 소프트웨어 구조에서 각 소프트웨어 모듈들 간의 통신은 룸 브릿지의 이벤트 서비스를 이용한다. 이러한 이벤트 서비스는 디바이스 및 제어 에이전트들 사이를 약한 연관관계(loosely-coupled connection)를 가지게 해준다. 홈 네트워크의 모든 디바이스와 에이전트들은 룸 브릿지의 이벤트 서비스에 자기 자신을 이벤트 수신자로 등록하여 이벤트 서비스를 이용하게 된다. <표 1>은 룸 브릿지에서 사용하는 이벤트 데이터의 구조를 나타낸다.

〈표 1〉 이벤트 데이터 구조

Type name	Field name	Type
DevAddr	addrType	short
	roomNum	short
	devManagerName	string
	devName	string
	attrName	string
	groupID	short
EvHeader	eventType	short
	srcAddr	DevAddr
	dstAddr	DevAddr
DevAttribute	name	string
	direction	short
	attrType	short
	value	string
	specificInfo	string
	bindAddr	DevAddr
	status	short
DIS	IOR	string
	name	string
	manufacture	string
	devAddr	DevAddr
	devAttr	<sequence> of DevAttribute
DevBind	srcAttrAddr	DevAddr
	dstAttrAddr	DevAddr

룸 브릿지에서 사용하는 주소 구조체인 DevAddr은 홈 네트워크를 구성하는 모든 객체들을 지정할 수 있다. addrType 필드는 주소의 형태를 나타낸다. 주소의 형태에는 특정 객체를 지정하는 일반 주소가 있고 속해있는 그룹을 지정하는 그룹 주소가 있다. 일반 주소를 지정하기 위해서 roomNum, devManagerName, devName, attrName의 필드가 있고, 그룹 주소를 지정하기 위해 groupID 필드가 존재한다. 또한 addrType 필드의 형태 설정을 통하여 방 단위, 디바이스 관리자 단위의 그룹 주소 지정도 가능하다. EvHeader 구조체는 이벤트 메시지의 정보를 표현한다. EvHeader 구조체의 eventType 필드는 이벤트 메시지의 종류를 알려준다. 메시지의 종류에는 디바이스 추가, 제

거, 바인딩, 속성 변화 등이 있다. 이벤트 메시지의 종류를 이벤트 헤더를 통해서 알 수 있기 때문에 룸 브릿지의 이벤트 서버는 이벤트 메시지의 내용을 확인할 필요가 없이 이벤트 메시지를 처리해야 하는 룸 브릿지의 객체로 전송하면 된다. srcAddr 필드는 이벤트를 생성한 디바이스나 에이전트의 주소 정보를 가지고 있으며 dstAddr 필드는 수취하는 디바이스나 에이전트의 주소 정보를 포함하고 있다.

디바이스가 가지고 있는 액션과 입출력 속성을 표현하기 위한 DevAttribute 구조체는 액션 또는 입출력 속성의 이름을 나타내는 name 필드와 속성의 상태 전달 방향인 direction 필드를 포함한다. 그리고 value 필드는 속성의 현재 데이터를 나타내며 attrType 필드는 value 필드의 데이터 형태를 알려준다. specificInfo 필드는 IEEE1394나 LonWorks에 연관되는 세부 정보를 보관하며 bindAddr 필드는 속성이 바인딩된 주소를 보관한다. 마지막으로 status 필드는 트랜잭션의 상태와 같은 속성의 현재 상태를 표시한다.

디바이스는 초기화 되어 홈 네트워크에 추가될 때 디바이스가 가지고 있는 모든 정보를 DIS 메시지 구조체에 저장하여 룸 브릿지의 자원 관리자에 등록한다. DIS 메시지 구조체의 IOR 필드는 룸 브릿지 객체들의 CORBA 원격 호출을 위한 객체 참조를 보관하며, devAddr 필드는 디바이스의 주소 정보를 가진다. 그리고 DIS 메시지 구조체는 디바이스의 모든 속성들을 포함한다.

4.3 디바이스간의 서비스 연결을 위한 구조

홈 네트워크 서비스는 두 가지 이상의 서비스가 서로 연동하여 동작하는 바인딩 서비스를 필요로 한다. 예를 들어 보안 시스템의 경우 감시 카메라와 여러 감시 센서가 연동하여 침입을 감시하고 경보기를 통해 침입을 알리는 서비스가 이루어진다.

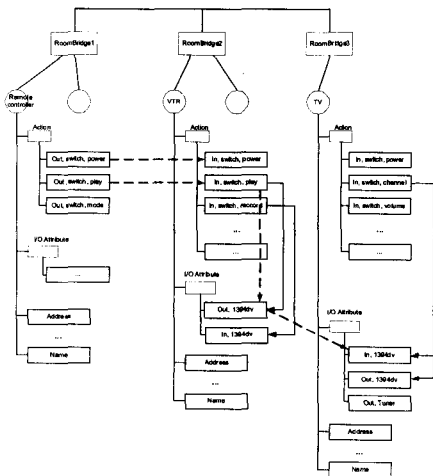
기존의 미들웨어들은 이러한 서비스들이 연동해서 동작하는 경우는 고려하지 않고 있다. 본 절에서는 서비스들을 바인딩하기 위한 데이터 구조와 바인딩 설정 과정에 대해 설명할 것이다.

4.3.1 바인딩 서비스

바인딩 서비스는 바인딩 에이전트로부터 디바이스 액션 필드 간의 바인딩 요청에 대하여 다음과 같은 작업을 수행한다.

- 바인딩 요청 시 자원 저장소에 저장되어있는 바인딩 정보에 대한 검사
- 서비스간의 바인딩 가능성 검사
- 새로운 바인딩 생성
- 기존의 바인딩에 액션 또는 입출력 속성의 바인딩 추가
- 기존의 바인딩 삭제

그리고 바인딩 관리자는 디바이스나GUI 에이전트로부터 요청된 액션의 변화에 대하여 저장되어 있는 바인딩 정보에 따르는 트랜잭션을 보장한다. 또한 서비스 바인딩에는 우선순위가 있어서 높은 우선순위를 가지는 서비스 바인딩에 대한 트랜잭션을 우선적으로 처리한다.

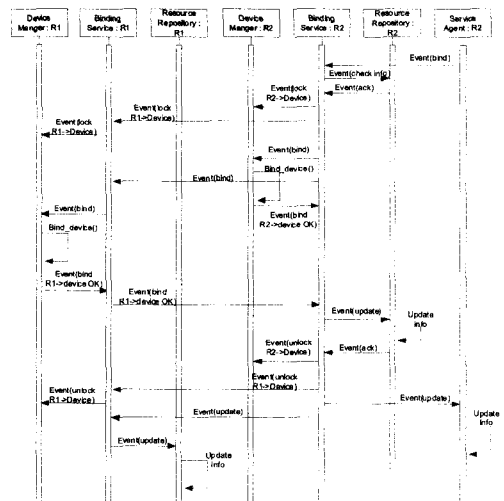


(그림 5)디바이스 바인딩 구조

(그림 5)는 룸 브릿지 1에 연결된 리모트 컨트롤러의 play 액션은 룸 브릿지 2에 연결된 VTR play 액션과 연결되어 있으며 이는 내부 속성인 1394dv와 연결되어 있어 있는 바인딩 관계를 보여준다. 또한 룸 브릿지 2에 연결된 VTR의 1394dv 속성은 룸 브릿지 3에 연결된 TV의 1394dv 속성과 연결되어 있다. 이러한 바인딩 관계에서 룸 브릿지 1에 연결된 리모트 컨트롤러의 play 버튼을 누르는 것은 룸 브릿지 2에 연결된 VTR을 play 상태로 만들고 룸 브릿지 3에 연결된 TV를 play 상태로 만들게 된다.

4.3.2 디바이스간의 서비스 바인딩 과정

(그림 6)은 디바이스 간의 액션 바인딩 과정을 나타내는 순서도다.



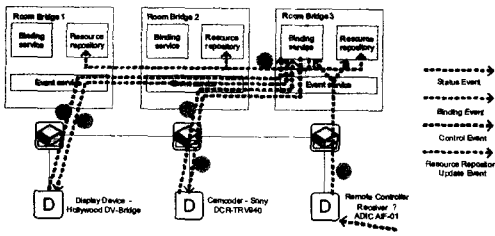
(그림 6) 디바이스간의 액션 바인딩 과정

서비스 에이전트로부터 디바이스간의 바인딩 요청이 왔을 때 룸 브릿지의 바인딩 서비스는 바인딩 가능성을 확인 후 디바이스 관리자에게 해당하는 디바이스에 대한 잠금을 요청한다. 그 후 각각의 디바이스에게 바인딩 이벤트 메시지를 전송하여 각각의 디바이스에 있는 바인딩 정보를 업데이트한다. 이 과정이 성공하면 바인딩 관리자는 자원

저장소에 있는 바인딩 리스트를 업데이트하고 각각의 디바이스 관리자에게 잠금 해제 요청을 전송한다. 그리고 다른 룸 서버의 바인딩 관리자에게 변경된 바인딩 리스트를 업데이트하도록 업데이트 이벤트 메시지를 전송한다.

바인딩으로 연결되어 있는 디바이스의 상태가 변경되면 바인딩에 관련된 모든 디바이스들은 연결된 디바이스에 이벤트를 받아 상태가 변경된다. 바인딩 서비스가 진행되는 과정에서 다른 서비스들이 바인딩으로 연결된 디바이스에게 제어 명령을 전달한다면 바인딩 서비스 과정은 보장받지 못하게 된다. 따라서 바인딩으로 연결된 모든 디바이스들이 다른 이벤트를 받지 않도록 잠금 과정이 이루어진다.

(그림 7)은 바인딩으로 연결된 디바이스들이 제어 이벤트를 받았을 경우 어떻게 동작하는지를 보여주는 순서도다. (그림 7)에서처럼 바인딩된 디바이스에게 제어 이벤트가 전달되면 바인딩 서비스가 관여하여 이벤트를 전송하게 된다.



(그림 7) 바인딩 서비스 수행 과정

5. 홈 네트워크 적용 사례

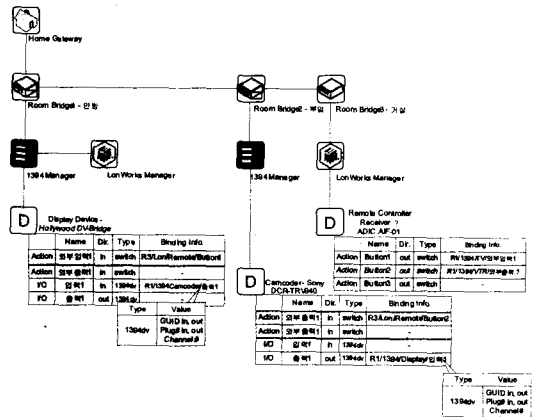
5.1 홈 네트워크 구현 환경

제안하는 홈 네트워크의 구현을 위하여 룸 브릿지로 인텔 x86 기반의 SBC(Single Board Computer)를 사용하였으며 Texas Instruments사

의 IEEE1394 인터페이스 보드 그리고 Gesytec사의 LonWorks 인터페이스 보드로 하드웨어를 구성하였다. 운영 체제는 Redhat Linux(kernel-2.4.18)을 사용하였으며, 기반 미들웨어로는 현재 GNOME 프로젝트로 진행되고 있는 C언어 기반의 ORBit CORBA ORB[15](ORBit-0.5.13)을 사용하여 (그림 3)의 제안하는 소프트웨어 모듈을 구현하였다. 홈 네트워크의 가전기기를 제어하기 위한 인터페이스는 Java를 이용하였으며 바인딩을 위한 에이전트도 Java를 이용하여 구현하였다.

사용된 ORBit CORBA ORB는 백본 네트워크 프로토콜로 사용하고 있는 IEEE1394를 지원하기 위해 GIOP/1394IOP 계층을 구현하였다[11]. 하부 프로토콜로 사용하는 LonWorks 디바이스 드라이버는 Gesytec사의 LonWorks Linux 디바이스 드라이버를 이용하여 상위의 LNS(LonWorks Network Service) 관련 모듈을 구현하여 사용하였으며 IEEE1394 디바이스 드라이버는 RTLInux 상에서 구현하여 사용하였다.

5.2 홈 네트워크 구성 및 바인딩 예



(그림 8) 디바이스 상태 정보 정의 예 및 바인딩 연결 구조

본 논문에서 제안한 바인딩 서비스를 지원하는 홈 네트워크 환경을 실질적으로 구현하기 위해 여

러 디바이스를 이용하였다. (그림 8)은 실제의 디바이스를 사용한 바인딩 연결 관계의 예다. (그림 8)에서 사용된 디바이스는 LonWorks 디바이스로 리모트 컨트롤러(ADIC AIF-01)를 사용하였으며, IEEE1394 디바이스로 캠코더(Sony DCR-TRV 940)와 디스플레이 장치(Hollywood DV-Bridge)를 이용하였다.

6. 결 론

본 논문에서는 홈 네트워크 환경을 위한 디바이스 구조를 제안하고 여러 디바이스들이 상호 연동하는 바인딩 서비스를 지원하는 미들웨어 구조를 제안하였다. 다양한 특징을 가지는 홈 네트워크의 디바이스를 표현하기 위해 액션과 속성을 가지는 네트워크 객체 방식으로 표현하였으며 통일된 인터페이스를 제공하였다. 제안된 디바이스 표현 방법은 홈 네트워크의 모든 디바이스를 하나의 통일된 네트워크 객체로 표현할 수 있으며 새로운 디바이스를 추가하거나 디바이스의 기능을 확장하기 쉬운 구조를 제공한다. 또한 다수의 디바이스가 참여하는 복잡한 서비스를 제공하기 위한 바인딩 구조체와 서비스를 제공하였다. 바인딩 서비스는 액션과 속성의 연결을 통해 하나의 디바이스 상태 변화가 추가적인 조작 없이 연결된 디바이스에게 전달되며 공통의 서비스를 수행하게 하도록 한다. 제안된 디바이스 구조 및 바인딩 서비스 구조는 다중 프로토콜을 지원하는 롬 브릿지 미들웨어를 기반으로 동작하며 실제의 홈 네트워크 상에서 테스트 되었다.

제안된 다중 프로토콜을 지원하는 홈 네트워크 상에서 동작하는 서비스 바인딩을 위한 미들웨어는 다양한 디바이스를 추상화하여 동일한 구조를 가지는 네트워크 객체로 표현하는 방법을 제시하였으며 바인딩 서비스를 통해 다수의 디바이스들이 공통된 서비스를 수행하기 위해 서로 연결될 수

있는 미들웨어 구조를 제안하였다.

미래의 홈 네트워크 구조는 더욱 더 다양한 네트워크 기능을 가진 디바이스들이 증가할 것이며 홈 네트워크에서 사용될 수 있는 프로토콜 또한 더욱 다양해질 것이다. 또한 홈 네트워크 환경에서 사용자가 원하는 서비스는 더욱 더 복잡해질 것이다. 따라서 여러 디바이스 및 서비스들이 상호 연동하여 상위 서비스를 수행하고 사용자의 요구 사항을 미리 인식하여 수행하는 홈 네트워크 서비스에 대한 연구가 앞으로 진행되어야 할 것이다.

참고문헌

- [1] HAVi, <http://www.havi.org>
- [2] UPnP forum, <http://www.upnp.org>
- [3] Jini Network Technology, <http://www.sun.com/software/jini>
- [4] Gerard O'Driscoll, The Essential Guide to Home Networking Technologies, Prentice Hall PTR, 2001
- [5] IEEE Std. 1394-1995, IEEE Std. for a High Performance Serial Bus, IEEE, 1995
- [6] LonTalk Protocol Specification Version 3.0, 1994
- [7] Bluetooth Resource Center, <http://www.palowireless.com/bluetooth>
- [8] TANGO - an object oriented control system based on CORBA, <http://www.esrf.fr/computing/cs/tango>
- [9] Tim Kindberg, Armondo Fox, "System Software for Ubiquitous Computing," Pervasive Computing, 2002
- [10] Soon Ju Kang, Jun Ho Park, Sung Ho Park, "ROOM-BRIDGE: A Vertically Configurable Network Architecture and Real-Time Middleware for Interoperability between

Ubiquitous Consumer Devices in Home,"
Lecture Notes in Computer Science
2218(p.232-251)

- [11] J.Y. Oh, J. H. Park, G.H. Jung and S. J. Kang, "CORBA based Core Middleware Architecture Supporting Seamless Interoperability between Standard Home Network Middlewares," IEEE Transaction on Consumer Electronics, AUG. 2003
- [12] Jun Ho Park, Soon Ju Kang, Jung Bae Lee, "Multi-Agent based Home Network Management System Using Extended Tuple Space Model," SAM'02 SCOPE
- [13] Ken SAKAMURA, Ubiquitous Computing KAKUMEI, 2002
- [14] Object Management Group, The Common Object Request Broker Architecture and Specification, Revision 2.2: Object Mangement Group, 1995
- [15] ORBit CORBA, <http://orbit-resource.sourceforge.net>

저자약력



박 준 호

1998년 2월 : 경북대학교 전자 공학과 졸업
2000년 2월 : 경북대학교 전자 공학과 석사
2000년 3월~현재 : 경북대학교 전자공학과 박사과정
관심분야 : 홈네트워크, 미들웨어, 실시간 시스템
이 메 일 : zec@palgong.knu.ac.kr



배 대 호

2000년 2월 : 경북대학교 전자 공학과 졸업
2000년 ~ 2002년 : LG전자 GSM 사업부 연구원
2002년 8월 ~ 현재 : 경북대학교 전자 공학과 석사과정
관심분야 : 홈네트워크, 실시간 시스템, 소프트웨어공학
이 메 일 : baedo@palgong.knu.ac.kr



강 순 주

1983년 2월 : 경북대학교 전자 공학과 졸업
1985년 2월 : 한국과학기술원 전자계산학과 석사
1995년 2월 : 한국과학기술원 전자계산학과 박사
1985년 ~ 1996년 : 한국원자력연구소, 핵인공지능연구실
선임연구원(과책), 전산정보실
선임연구원(실장)
1996년 ~ 현재 : 경북대학교 전자전기공학부 정보통신전공
부교수
2000년 ~ 2001년 : University of Pennsylvania. Dept. of
CIS. 방문연구교수
관심분야 : 실시간 시스템, 임베디드 시스템,
지식기반시스템
이 메 일 : sjkang@ee.knu.ac.kr