

홈네트워크 서비스와 데이터방송 서비스의 컨버전스를 위한 구조 설계 및 구현

배 유 석[†] · 오 봉 진[†] · 문 경 덕^{**} · 김 상 욱^{***}

요 약

본 논문에서는 홈네트워크 기기의 상호 운용성을 제공하는 UMB와 데이터방송 서비스를 지원하는 ACAP 미들웨어를 기반으로 홈네트워크 서비스와 방송 서비스의 상호 운영을 위한 컨버전스 구조를 제시한다. 제안된 구조는 데이터방송 서비스처럼 홈네트워크 서비스에 대하여 Java TV Xlet 인터페이스와 HAVi 사용자 인터페이스를 적용하여 구성함으로써 리모콘과 TV 수상기를 통하여 홈네트워크와 데이터방송 서비스를 함께 제어할 수 있는 TV 중심의 사용자 인터페이스를 제공한다. 또한, 셋톱박스에서 홈네트워크 서비스를 접근하기 위한 서비스 프락시와 홈 서버에서 서비스 관련 데이터를 제공하는 서비스 브로커를 정의하고, ACAP 미들웨어의 애플리케이션 매니저는 서비스 프락시에서 애플리케이션 매니저의 서비스 이벤트를 통하여 홈네트워크 서비스에 대한 라이프사이클을 관리한다.

키워드 : 홈네트워크, 데이터방송, 컨버전스, UMB, ACAP

Design and Implementation of Architecture for Convergence of Home Network Services and Data Broadcasting Services

Yu-Seok Bae[†] · Bong-Jin Oh[†] · Kyeong-Deok Moon^{**} · Sang-Wook Kim^{***}

ABSTRACT

This paper describes the design and implementation of architecture for convergence of home network services and data broadcasting services based on UMB and ACAP middleware; the one is used for interoperability of home networked devices, and the other for data broadcasting services. The proposed architecture provides TV-centric user interface capable of managing home network services and data broadcasting services together using a remote controller and TV sets by applying Java TV Xlet interface and HAVi user interface for home network services like data broadcasting services. Besides, we define the service proxy for accessing home network services in a set-top box and the service broker for providing data related to home network services in a home server. Finally, the application manager in ACAP middleware manages the lifecycle of home network services through service events from the service proxy to the application manager.

Key Words : Home Network, Data Broadcasting, Convergence, UMB, ACAP

1. 서 론

홈네트워크 기술과 디지털 방송 기술의 컨버전스가 가속 화됨에 따라 홈네트워크 기기를 제어하고 상호 운용하는 홈네트워크 서비스와 디지털 방송을 통하여 고품질의 영상 서비스뿐만 아니라 고부가가치의 정보를 제공하는 데이터방송 서비스를 접목한 융합 서비스에 대한 요구가 증가되고 있다. 따라서, 미래의 홈네트워크 환경에서는 이러한 홈네트워크 서비스와 데이터방송 서비스를 융합한 새로운 형태의 서

비스를 보다 편리하게 사용자에게 효과적으로 제공하는 기술의 개발이 중요하게 될 것이다.

홈네트워크 환경에서는 다양한 기기들이 존재하며, 또한 기기별로 지원하는 미들웨어가 구분됨으로써 상호 운용하거나 이들 기기를 연동한 서비스의 제공이 쉽지 않다. UMB(Universal home network Middleware Bridge)[1, 2]는 HAVi(Home Audio/Video interoperability)[3], Jini[4], LonWorks[5], UPnP(Universal Plug and Play)[6] 등 다양한 홈네트워크 미들웨어를 연동할 수 있는 구조를 제시함으로써 홈네트워크에 연결된 이종의 기기간의 상호 운용성을 제공한다.

데이터방송 서비스를 제공하기 위해서는 셋톱박스에서 디지털 방송 신호로부터 수신되는 MPEG-2 트랜스포트 스트

† 정 회 원 : 한국전자통신연구원 선임연구원
 ** 정 회 원 : 한국전자통신연구원 책임연구원
 *** 정 회 원 : 경북대학교 컴퓨터학과 교수
 논문접수 : 2006년 8월 28일, 심사완료 : 2006년 9월 11일

림을 필터링하여 서비스 정보를 분석하고 애플리케이션의 다운로드, 애플리케이션의 실행, 종료 등 애플리케이션 라이프사이클 관리, 셋톱박스의 자원을 관리하는 미들웨어를 필요로 한다[7]. 특히, 미들웨어 중에서 애플리케이션의 실행 및 상태 관리를 위해 애플리케이션 매니저의 역할이 중요하다. 애플리케이션 매니저는 애플리케이션 시그널링 정보에 따라 애플리케이션의 라이프 사이클을 관리하고 셋톱박스의 제한된 시스템 자원을 여러 애플리케이션들이 공유할 수 있도록 효율적으로 관리하는 역할을 한다. ACAP(Advanced Common Application Platform)[8]은 국내 지상파 데이터방송 미들웨어 표준으로 지정되어 각 방송사에서 다양한 데이터방송 서비스를 제공하고 있다.

따라서, 본 논문에서는 미들웨어 측면에서 홈네트워크 기간 상호 운용성을 제공하는 UMB와 데이터방송 미들웨어 표준인 ACAP을 접목하여 홈네트워크 서비스와 데이터방송 서비스를 상호 연동할 수 있는 컨버전스 구조를 제시한다. 본 논문에서 제안된 구조는 홈서버, 셋톱박스, 홈네트워크 기기로 구성된 홈네트워크 환경에서 홈네트워크 서비스에 대하여 데이터방송 서비스처럼 TV와 같은 가전 기기를 위해 경량화된 사용자 인터페이스를 제공하는 HAVi 사용자 인터페이스[9]와 서비스의 실행, 종료 등의 애플리케이션 라이프사이클 관리를 위해 Java TV[10]에 정의된 Xlet 인터페이스를 적용함으로써 ACAP 미들웨어의 애플리케이션 관리자를 통하여 데이터방송 서비스와 더불어 홈네트워크 서비스를 실행하고 관리할 수 있으며, 홈네트워크 서비스와 데이터방송 서비스가 결합된 새로운 형태의 융합형 서비스를 구축할 수 있는 기틀을 제공한다. 또한, 홈네트워크 기기의 연결 상태에 따라 셋톱박스에서 홈네트워크 서비스를 접근하고 이벤트를 수신하는 서비스 프락시(Service Proxy)와 홈서버에서 서비스 관련 정보 및 Xlet 애플리케이션을 전달하는 서비스 브로커(Service Broker)를 정의한다. 서비스 프락시는 홈네트워크 서비스에 관련된 서비스 이벤트를 ACAP 미들웨어의 애플리케이션 매니저에게 전달하여 홈네트워크 서비스와 데이터방송 서비스를 통합 관리할 수 있는 환경을 지원한다.

본 논문은 제 2 장에서 본 논문과 관련된 선행연구에 대하여 간략히 언급하며, 제 3 장에서는 UMB와 ACAP을 기반으로 홈네트워크 서비스와 데이터방송 서비스를 연동하기 위한 서비스 컨버전스 구조를 제시하며, 제 4 장에서는 구현 환경 및 결과를 언급하며, 제 5 장에서 결론을 맺는다.

2. 관련 연구

홈네트워크 기술의 발전으로 이종의 미들웨어를 지원하는 다양한 종류의 홈네트워크 기기들이 태내에 존재한다. 현재 HAVi, Jini, LonWorks, UPnP 등 이용 가능한 홈네트워크 미들웨어들은 물리적으로 연결이 되어 있다고 할지라도 명령어나 통신 프로토콜의 차이로 인해 서로간에 호환이 되지

않는다. UMB는 다른 홈네트워크 미들웨어에 속하는 물리적인 기기들을 동적으로 UMB 영역내의 가상 추상 기기로 매핑하여 이종의 홈네트워크에 존재하는 모든 홈네트워크 기기들을 같은 미들웨어 영역내의 가상 추상 기기로 보여지도록 함으로써 UPnP, LonWorks, IEEE1394, USB 등의 이종 홈네트워크 기기들의 상호 접속 및 운용성을 보장한다.

데이터방송 미들웨어는 전송 방식에 따라 미국의 DASE(Digital TV Application & Software Environment) [11], 유럽의 DVB-MHP(Digital Video Broadcasting-Multimedia Home Platform)[12], 미국 CableLabs에서 제정한 케이블 방송 표준인 OCAP(OpenCable Application Platform)[13]으로 나눌 수 있다. ACAP은 ATSC를 주축으로 하여 DASE의 후속으로 DVB-MHP의 응용 프로그램 수행 환경 표준인 GEM(Globally Executable MHP)[14]과 데이터방송 서비스의 구성 정보를 나타내는 표준인 PSIP(Program and System Information Protocol)[15]을 결합한 미들웨어이다. ACAP은 애플리케이션 환경 언어로 Java 기술을 언급하며, Java TV, HAVi UI, DAVIC(Digital Audio Visual Council)[16], DVB(Digital Video Broadcasting) [17], DSM-CC(Digital Storage Media Command and Control) [18] 등의 API 집합을 포함하며, 소비자에게 고부가가치의 상호작용적인 데이터방송 서비스를 제공하기 위해 제정된 표준이다. ACAP은 국내 지상파 데이터방송 미들웨어 표준으로 제정되어 각 방송사를 중심으로 다양한 데이터방송 서비스가 제공되고 있다.

DLNA(Digital Living Network Alliance)[19]는 제조사의 기기들 사이의 상호 운용성을 보장하기 위한 개방형 표준 프레임워크에 기반한 가이드라인과 미디어 레벨에서 상호 운용성을 보장하기 위한 공통적인 베이스라인을 제공한다. DLNA의 가이드라인은 홈네트워크에서 기기들이 자동적으로 인식하는 것을 허용하기 위해 UPnP 표준을 수용한다.

또한, 미국 CableLabs에서는 OCAP 미들웨어의 확장된 부분으로 홈네트워크를 결합하기 위한 연구를 진행하고 있다. 홈네트워크 상에서 공통적인 프로토콜을 사용하여 OpenCable 기기간의 상호운용성을 제공하기 위해 물리적인 통신망이나 전송 프로토콜에 독립적인 OpenCable HN(Home Networking) API를 정의하고, 이를 UPnP 디바이스 아키텍처[20]나 UPnP AV 아키텍처[21]로 매핑함으로써 기기 검색 및 제어, 미디어 관리, 미디어 전송 기능을 제공하는 OpenCable 홈네트워킹 아키텍처를 제시하고 있다[22].

따라서, 본 논문에서 제시하는 홈네트워크 서비스와 데이터방송 서비스의 컨버전스를 위한 구조는 미들웨어 측면에서 UMB와 ACAP을 유연하게 접목함으로써 태내에서 홈네트워크 서비스뿐만 아니라 데이터방송 서비스를 하나의 단말에서 제공할 수 있고, 홈네트워크 서비스와 방송 서비스를 접목한 융합형 서비스를 효과적으로 수용할 수 있는 구조를 제시한다.

3. 홈네트워크 서비스와 데이터방송 서비스의 컨버전스 구조

디지털 홈 환경은 다양한 종류의 홈네트워크가 존재하며, 이들 홈네트워크에는 다양한 홈네트워크 미들웨어를 지원하는 다수의 기기들이 연결되어 있다. 이러한 이질적이고 복잡한 미들웨어 환경에서 다수의 기기를 연동하여 서비스를 구성하기 위해서는 상호 운용성을 제공하는 모듈이 필요하며, UMB는 유선 또는 무선으로 연결된 홈네트워크 환경에서 이종의 홈네트워크 기기들의 상호 운용성을 지원한다.

또한, 데이터방송 서비스를 제공하기 위해서는 셋톱박스에서 방송 스트림으로부터 MPEG-2 트랜스포트 스트림을 필터링하여 서비스 정보를 분석하고 애플리케이션을 실행 및 관리할 수 있는 미들웨어를 필요로 한다. ACAP 미들웨어에서의 데이터방송 서비스용 애플리케이션은 GEM 표준에서 기술된 Java TV Xlet기반의 DVB-J를 모델로 한 ACAP-J 애플리케이션과 XHTML 기반의 DVB-HTML을 모델로 한 ACAP-X 애플리케이션으로 구분된다. 현재 ACAP 표준에서는 ACAP-X는 선택 사항으로 기술하고 있으며, 본 논문에서 제시하고 있는 ACAP 데이터방송 미들웨어에서는 ACAP-J 애플리케이션만을 취급한다.

ACAP 데이터방송 미들웨어는 MPEG-2 트랜스포트 스트림에 포함된 AIT(Application Information Table)를 추출하여 애플리케이션 관련 정보를 획득하고, DSM-CC 오브젝트 캐로셀 및 데이터 캐로셀을 통하여 ACAP-J 애플리케이션을 다운로드하며, 애플리케이션 매니저(Application Manager: AM)을 통하여 Xlet 애플리케이션을 수행하고 라이프사이클을 관리한다.

(그림 1)은 홈네트워크 기기간의 상호 운용성을 보장하여 홈네트워크 서비스를 제공하는 UMB와 데이터방송 서비스를 지원하는 방송 미들웨어 ACAP을 통하여 홈네트워크 서비스와 방송 서비스를 유연하게 접목하여 상호 연동하기 위한 소프트웨어 구조를 보여준다.

ACAP 셋톱박스는 지상파 튜너를 통하여 MPEG-2 트랜스포트 스트림을 수신하여 데이터방송 서비스를 제공할 뿐만 아니라 UMB와 통신하여 홈네트워크 서비스를 제공하는 단말이다. 홈서버는 이종의 홈네트워크 기기들을 연결하는 중앙 허브로서 UMB를 통하여 원격으로 가전기기들을 제어하고 기기들의 상태를 모니터링한다. 또한, 셋톱박스와 홈서

버간의 이벤트 시그널링 및 데이터 교환을 위해 서비스 프락시와 서비스 브로커가 사용된다.

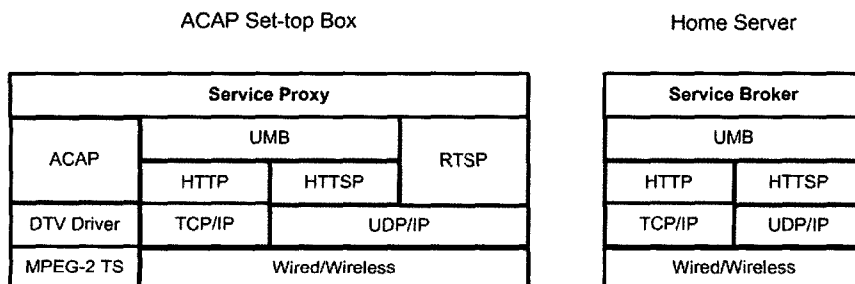
ACAP 셋톱박스에서 홈네트워크 서비스와 방송 서비스를 유연하게 연동하기 위해서는 데이터방송 서비스처럼 홈네트워크 서비스에 대하여 일관된 형태의 인터페이스를 적용하는 것이 효율적이다. 그러므로 본 논문에서는 홈네트워크 서비스에 대하여 Java TV Xlet 인터페이스와 HAVi UI를 적용하여 애플리케이션을 구성함으로써 AM으로 하여금 홈네트워크 서비스의 라이프사이클과 제한된 자원을 효율적으로 관리할 수 있도록 한다. Java TV Xlet 인터페이스는 셋톱박스에서 애플리케이션의 라이프사이클을 제어하기 위해서 사용되며, HAVi UI는 Java TV Xlet 애플리케이션이 디스플레이 기기의 사용자 인터페이스 기능을 결정하고 사용자로부터 리모콘 입력을 받고 스크린상에 각종 드로잉을 지원하는 경량 컴포넌트로 구성되어 있다.

ACAP 셋톱박스는 MPEG-2 트랜스포트 스트림으로부터 색션 필터링을 이용하여 AIT 테이블을 추출하며, DSM-CC를 통하여 AIT와 연관된 ACAP-J 애플리케이션을 다운로드하고 애플리케이션을 구동함으로써 데이터방송 서비스를 제공한다.

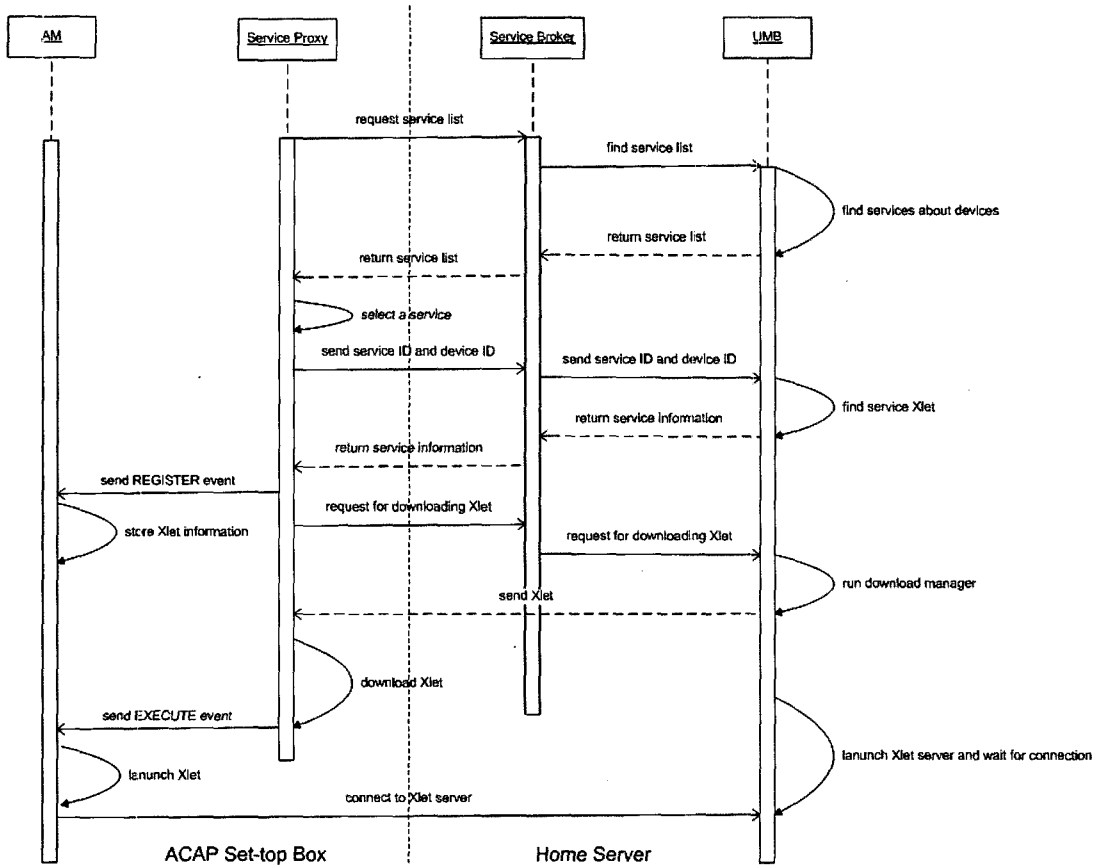
한편, (그림 2)는 홈네트워크 서비스를 제어하기 위해 ACAP 셋톱박스와 홈서버 사이의 이벤트 시그널링을 보여주는 시퀀스 다이어그램이다.

홈네트워크 서비스를 제어하기 위해서는 ACAP 셋톱박스내의 서비스 프락시는 SSDP(Simple Service Discovery Protocol)를 통하여 홈서버의 서비스 브로커에게 연결하고, 모든 홈네트워크에 존재하는 기기와 서비스 목록을 요청한다.

서비스 브로커는 UMB를 통하여 모든 기기와 서비스의 목록을 구성하며, 목록을 서비스 프락시에게 전달한다. 만일 사용자가 목록으로부터 어떤 서비스를 선택하면, 서비스 프락시는 해당 서비스에 대한 서비스 ID와 기기 ID를 서비스 브로커에게 전달하며, 서비스 브로커는 해당하는 서비스 Xlet애플리케이션을 찾아서 관련 정보를 서비스 프락시에게 통보한다. 서비스 프락시는 AM에게 새로운 서비스의 도착을 알리고 등록하기 위해 REGISTER 서비스 이벤트를 전달한다. 한편, 서비스 브로커가 UMB를 통하여 기기의 플러그인을 탐지할 때마다 해당 기기와 관련된 서비스 Xlet애플리케이션을 찾아서 관련 정보를 서비스 프락시에게 전달하며, 서비스 프락시는 REGISTER 서비스 이벤트를 통하여



(그림 1) 홈네트워크 서비스와 데이터방송 서비스의 상호 연동 소프트웨어 구조



(그림 2) 이벤트 시그널링 시퀀스 다이어그램

해당 서비스를 AM에 등록한다.

<표 1>은 홈네트워크 기기의 연결이나 해제시 홈네트워크 서비스 실행을 위해 서비스 프락시와 AM 사이에 발생하는 서비스 이벤트 분류를 보여준다.

<표 2>는 홈네트워크 서비스를 등록하기 위해서 사용되는 REGISTER 서비스 이벤트에 포함되는 필드의 세부사항을 보여준다.

REGISTER 이벤트는 이벤트 타입, 애플리케이션 ID, 서비스 이름, 디렉토리, 시작 클래스 이름, 패러미터로 구성되며, 이러한 정보는 AIT 테이블 정보를 대체하며, 애플리케이션 ID는 홈네트워크 서비스 Xlet과 매핑되며, AM에서 Xlet을 구분하고 관리하기 위한 키로서 사용된다.

REGISTER 이벤트를 AM에게 전달하여 서비스를 등록한 이후, 서비스 프락시는 홈서버의 UMB로부터 서비스 Xlet

<표 1> 서비스 이벤트 분류

이벤트	의미	발생 시점
REGISTER	서비스 추가	사용자가 서비스를 선택하는 경우 기기가 플러그인 되는 경우
UNREGISTER	서비스 삭제	사용자가 서비스를 종료하는 경우 기기가 플러그아웃 되는 경우
EXECUTE	서비스 실행	Xlet 애플리케이션 다운로드가 완료되는 경우

<표 2> REGISTER 서비스 이벤트의 세부사항

필드	의미	예제
이벤트 타입	서비스 이벤트 유형	REGISTER
애플리케이션 ID	애플리케이션 ID (organization id, application id)	(0x01, 0x01)
서비스 이름	홈네트워크 서비스 이름	Home Security
디렉토리	기본 디렉토리 경로	/
시작 클래스 이름	Java TV Xlet 인터페이스를 구현한 클래스	HSSXlet
패러미터	Xlet을 구동하기 위한 패러미터	null

애플리케이션을 다운로드하며, 다운로드가 완료되면 EXECUTE 서비스 이벤트를 AM에게 전달하여 새로운 애플리케이션을 구동하도록 한다. 이때 AM은 그래픽, 메모리 등 자원의 점유 상태와 다른 Xlet의 수행 상태를 확인하여 필요하다면 현재 수행중인 Xlet을 종료하고 새로운 Xlet을 구동한다.

또한 사용자가 서비스를 종료하거나 서비스 브로커가 UMB를 통하여 어떤 기기의 플러그아웃을 탐지하게 되면 서비스 프락시에게 플러그아웃 정보를 전달하고, 서비스 프락시는 AM에게 UNREGISTER 서비스 이벤트를 전달하여 AM으로 하여금 애플리케이션 ID를 이용하여 해당되는 서비스 Xlet을 삭제하도록 한다.

AM에서는 이러한 서비스 이벤트 처리를 위하여 이벤트 별로 필요한 값들을 설정하여 이벤트 객체를 생성할 수 있는 AppsEvent 클래스와 AppsEvent 발생시 이벤트 리스너로 등록된 객체에 이벤트를 전달하기 위한 AppsEventManager 클래스를 정의하고 있으며, AM에서는 AppsEventManager에게 리스너를 등록하여 서비스 이벤트 발생시 이벤트를 전달받아 홈네트워크 서비스를 등록하고 실행하는 과정을 수행한다.

<표 3>은 AppsEvent 클래스의 일부로서 서비스 이벤트 관련 부분을 보여준다.

<표 4>는 AppsEventManager 클래스의 일부로서 서비스 이벤트 리스너 등록과 삭제 및 서비스 이벤트 포스팅 관련 부분을 보여준다.

또한 AM 클래스의 handleEvent 함수 내부에서는 REGISTER_XLET, UNREGISTER_XLET, EXECUTE_XLET 이벤트에 대한 처리 함수를 구현하여 서비스의 등록, 삭제, 실행 과정을 수행한다.

(그림 3)은 ACAP 셋톱박스과 홈서버 사이의 홈시큐리티 서비스의 동작 모델을 보여준다.

홈시큐리티 서비스는 센서를 이용하여 태내에서 외부의 침입을 탐지하고 센서가 위치한 곳의 카메라 영상을 제공해주는 서비스로 USB 카메라와 IEEE1394 카메라를 포함한다. UMB에서는 UPnP-USB 프락시와 UPnP-1394 프락시를 제공하여 USB 카메라와 IEEE1394 카메라를 UPnP 기기로 보이도록 하여 UPnP 어댑터에 연결되고 UPnP 어댑터가 UMB에 연결된다. UPnP-USB 프락시는 USB 기기를 UPnP 네트워크에 연결된 가상 기기로 매핑하고 UPnP-1394 프락시는 IEEE1394 기기를 매핑하기 위해 사용된다. UMB에서는 홈네트워크 미들웨어를 지원하지 않는 저수준 기기의 경우 특정 어댑터로 매핑할 수 있는 프락시 개념을 적용함으로써 새로운 기기의 출현이나 기존 홈네트워크 기기들을 손쉽게 연동할 수 있는 확장성을 제공한다.

<표 3> AppsEvent 클래스

```
import org.dvb.appliation.AppID;

public class AppsEvent {

    public final static int REGISTER_XLET = 21;
    public final static int UNREGISTER_XLET = 22;
    public final static int EXECUTE_XLET = 23;

    private int kind = -1;
    private AppID appid = null;
    private String appname = null;
    private String dir = null;
    private String classname = null;
    private String[] params = null;
    private Object source = null;

    // REGISTER_XLET
    public AppsEvent(int kind, AppID appid, String appname, String dir,
                    String classname, String[] params, Object source) {
        this.kind = kind;
        this.appid = appid;
        this.appname = appname;
        this.dir = dir;
        this.classname = classname;
        this.params = params;
        this.source = source;
    }

    // UNREGISTER_XLET, EXECUTE_XLET
    public AppsEvent(int kind, AppID appid, Object source) {
        this.kind = kind;
        this.appid = appid;
        this.source = source;
    }
}
```

〈표 4〉 AppsEventManager 클래스

```

public class AppsEventManager {

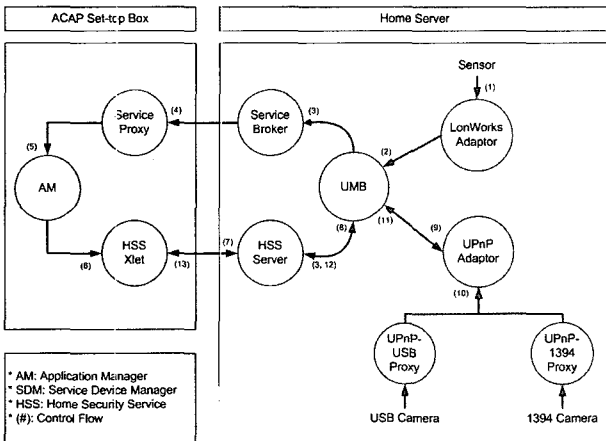
    ArrayList listeners = new ArrayList();

    public void addAppsEventListener(AppsEventListener listener) {
        listeners.add(listener);
    }

    public void removeAppsEventListener(AppsEventListener listener) {
        listeners.remove(listener);
    }

    synchronized public void postEvent(final AppsEvent event) {
        int type = event.getEventId();

        switch (type) {
            case AppsEvent.REGISTER_XLET:
            case AppsEvent.UNREGISTER_XLET:
            case AppsEvent.EXECUTE_XLET:
                for (int i = 0; i < listeners.size(); i++) {
                    AppsEventListener listener =
                        (AppsEventListener) listeners.get(i);
                    if (listener instanceof AM) {
                        listener.handleEvent(event);
                    }
                }
                break;
        }
    }
}
    
```



(그림 3) 홈시큐리티 서비스 동작 모델

센서에서 이벤트가 탐지되면 서비스 브로커는 센서와 관련된 서비스 관련 정보를 서비스 프락시에게 전달하고, UMB는 홈시큐리티 서비스와 관련된 HSSXlet을 찾고 대응하는 HSS 서버를 구동한다. 한편, 서비스 프락시는 HSS 서비스를 ACAP 미들웨어의 AM에 등록하고 HSSXlet을 다운로드하며, 다운로드 완료 후 홈시큐리티 서비스를 실행하기 위해 HSSXlet의 구동을 AM에게 요청한다. AM은 데이터방송 서비스를 시청하는 중간에 홈시큐리티 서비스의 실행 요청을 받으면 현재 수행중인 데이터방송 애플리케이션 Xlet의 destroyXlet 함수를 호출함으로써 데이터방송 서비스의

실행을 종료하며, HSSXlet의 initXlet 함수와 startXlet 함수를 호출하여 HSSXlet을 구동한다. HSSXlet은 HSS 서버와 통신하면서 UMB를 통하여 UPnP 어댑터에게 제어 명령어를 전달하여 홈시큐리티 서비스를 수행한다.

따라서, 홈시큐리티 서비스는 UMB에서 센서를 제어하기 위한 LonWorks 미들웨어와 AV 기기를 위한 UPnP 미들웨어의 2가지 홈네트워크 미들웨어를 상호 연동하고 있으며, HSSXlet의 경우 Java TV Xlet 인터페이스와 HAVi UI를 적용함으로써 ACAP 미들웨어를 통하여 홈네트워크 서비스를 제어함으로써 홈네트워크 서비스와 방송 서비스의 컨버전스를 보여준다.

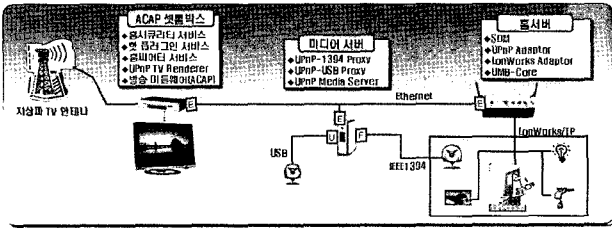
4. 구현

(그림 4)는 홈네트워크 서비스와 방송 서비스를 상호 운용하기 위한 프로토타입 시스템 구성도를 보여준다.

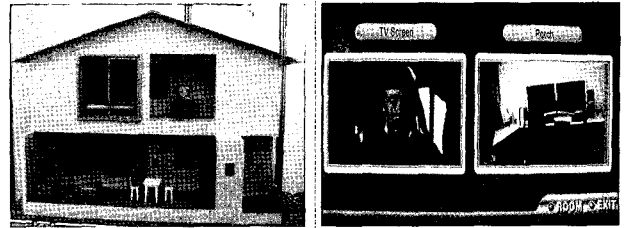
프로토타입 시스템은 ACAP 셋톱박스, 미디어 서버, 홈서버, 그리고 홈네트워크 가전 기기들로 구성되며, Ethernet, IEEE1394, USB, LonWorks와 같은 다양한 홈네트워크가 존재한다. ACAP 셋톱박스는 인텔 모바일 펜티엄 733MHz 프로세서와 하드웨어 디코더 칩으로 ATI Xilleon 225 프로세서를 내장하고 있으며, 임베디드 리눅스 운영체제(Qplus)를 기반으로 하고 있다. ACAP-J 미들웨어를 위한 Java VM으로 CDC-PP 기반의 CVM을 사용하고 있다. 셋톱박스는 8VSB 튜너를 통하여 디지털 지상파 방송을 수신하고 ATI 프로세서를 이용하여 수신된 MPEG-2 트랜스포트 스트림으로부터 PAT(Program Association Table)와 PMT(Program Map Table) 테이블의 색션 필터링을 수행하여 오디오/비디오를 디코딩하며, AIT 색션 필터링을 통해 애플리케이션 정보를 추출하고 해당 애플리케이션을 다운로드하여 ACAP 기반의 데이터방송 서비스 애플리케이션을 수행할 뿐만 아니라 홈서버의 UMB와 통신함으로써 홈네트워크 서비스 애플리케이션을 구동할 수 있다. 또한 홈시큐리티 서비스가 구동될 때 카메라 영상을 TV 화면에 렌더링 기능을 제공하는 UPnP 미디어 렌더러의 역할을 수행한다.

미디어 서버는 미디어 콘텐츠를 제공하는 UPnP 미디어 서버의 역할 뿐만 아니라 USB 카메라나 IEEE1394 카메라를 UPnP 어댑터에 연결할 수 있도록 프락시를 제공한다. 한편, 홈서버는 UMB를 통하여 UPnP, IEEE1394, USB, LonWorks 등의 홈네트워크에 연결된 이종의 홈네트워크 기기들을 제어하고 상태를 모니터링한다.

홈네트워크 서비스와 데이터방송 서비스를 유연하게 연동하기 위해서 본 논문에서는 홈네트워크 서비스에 대하여 방송 서비스와 유사하게 Java TV Xlet 인터페이스와 HAVi UI를 적용하여 TV를 통하여 데이터방송 서비스뿐만 아니라 홈네트워크 서비스를 함께 제공할 수 있도록 하였으며, 홈네트워크 서비스와 방송 서비스의 연동 구조를 검증하기 위해 전동, 밸브, 카메라, 센서 등 디지털 가전기기로 구성된 미니어처 버전의 테스트베드를 구성하였다.



(그림 4) 프로토타입 시스템 구성도



(그림 5) 테스트베드 및 홈시큐리티 서비스 실행 결과 화면

〈표 5〉 홈시큐리티 서비스를 위한 HSSXlet 클래스

```

import javax.tv.xlet.Xlet;
import javax.tv.xlet.XletContext;
import org.havi.ui.HScene;
import org.havi.ui.HSceneFactory;

public class HSSXlet implements Xlet {
    XletContext ctx = null;
    HScene scene = null;
    HSS_UI hss = null;

    public void initXlet(XletContext ctx) {
        this.ctx = ctx;

        HsceneFactory factory = HSceneFactory.getInstance();
        scene = factory.getDefaultHScene();

        scene.addFocusListener(this);
        scene.setVisible(false);

        // HSS: Home Security Service
        hss = new HSS();
        scene.add(hss);
    }

    public void startXlet() {
        scene.setVisible(true);
        hss.setVisible(true);
        hss.validate();
        hss.requestFocus();
    }

    public void pauseXlet() {
        scene.setVisible(false);
    }

    public void destroyXlet() {
        scene.setVisible(false);
        scene.removeAll();
        scene.removeFocusListener(this);
        scene.dispose();

        ctx.notifyDestroyed();
    }
}
    
```

〈표 5〉는 홈시큐리티 서비스를 위한 HSSXlet 클래스의 일부분으로 HAVi UI와 Java TV Xlet 인터페이스 API의 사용 예를 보여준다.

Xlet 인터페이스는 셋톱박스에서 애플리케이션 라이프사이클을 제어하기 위해 Java TV API에서 도입되었으며, HAVi UI는 Xlet에서 알파블랜딩을 통하여 데이터 방송 서비스를 제공할 수 있도록 그래픽 영역을 위한 AWT 루트 컨테이너(root container)를 획득하기 위해 사용된다.

(그림 5)는 홈네트워크 서비스를 위한 테스트베드와 현관에서 발생한 이벤트에 대한 홈시큐리티 서비스의 실행 결과 화면을 보여준다.

홈시큐리티 서비스는 적외선 센서와 USB 또는 IEEE1394 카메라를 사용하며, 방송 서비스를 시청하는 중간에 현관에 있는 센서로부터 이벤트가 탐지되면 자동적으로 구동되며, 현관에 있는 USB 카메라를 통하여 영상 이미지를 획득하고 UPnP-USB 프락시에 연결된 UPnP 어댑터를 사용하여 영상 이미지를 셋톱박스로 전송하고 셋톱박스에서 미디어 렌더러를 통하여 화면에 출력한다.

홈시큐리티 서비스 외에 가전 기기가 플러그인될 때 가전 기기의 목록 및 기기가 설치될 수 있는 공간과의 연결관계를 제시하여 기기의 위치 정보를 설정하고 관리하는 핫플러그인 서비스, TV를 통하여 벨브, 전등, 카메라와 같은 가전 기기를 선택하여 직접 제어하는 기기 제어 서비스, 미디어 콘텐츠와 방송 스트림 콘텐츠를 UPnP 기반의 미디어 스트리밍을 통하여 홈네트워크 기기와 셋톱박스 사이에 콘텐츠를 서로 공유하고 디스플레이 장치와 같은 공간에 있는 전등, 커튼 등을 제어하여 콘텐츠를 시청하기 좋은 환경으로 자동으로 설정해주는 홈씨어터 서비스 등을 제공한다.

5. 결 론

디지털 컨버전스 환경에서 대표적인 통신 방송 융합 사례로 방송과 홈네트워크의 결합을 들 수 있다. 본 논문에서는 미들웨어 측면에서 UMB와 ACAP을 기반으로 홈네트워크 서비스와 데이터방송 서비스를 상호 연동할 수 있는 컨버전스 구조를 제시하였으며, 테스트베드를 통하여 제안된 구조를 검증하였다. 홈네트워크 서비스와 데이터방송 서비스를 유연하게 접목하기 위해서 홈네트워크 서비스의 라이프사이클 관리를 위해 Java TV Xlet 인터페이스를 적용하고, TV 중심의 편리한 사용자 인터페이스를 제공하기 위하여 HAVi UI와 같은 방송 미들웨어의 표준 인터페이스를 적용함으로써 ACAP 미들웨어의 애플리케이션 관리자를 통하여 데이터방송 서비스와 더불어 홈네트워크 서비스를 실행하고 관리할 수 있는 메커니즘을 제공한다. 또한 UMB와 ACAP의 정보 교환을 위해 서비스 프락시와 서비스 브로커를 정의함으로써 홈네트워크 미들웨어와 방송 미들웨어를 연동하는 모델을 제시하였다.

본 논문에서 제안된 구조는 미래의 홈네트워크 환경에서

홈네트워크 기술과 방송 기술의 융합 과정에서 발생할 수 있는 상호 운용성 문제에 대한 유연한 해결책을 제시할 수 있으며, 향후 연구로는 우선 순위를 고려하여 서비스 요청 시 지능적인 애플리케이션 라이프사이클 관리가 될 수 있도록 하여 보다 편리한 서비스 실행 환경을 제공하는데 초점을 둘 예정이다.

참 고 문 헌

[1] K. D. Moon, Y. H. Lee, and C. E. Lee, "Design of a Universal Middleware Bridge for Device Interoperability in Heterogeneous Home Network Middleware," IEEE Transaction on Consumer Electronics, Vol.51, No.1, pp.314-318, Feb., 2005.

[2] D. H. Kim, J. H. Park, P. Yevgen, K. D. Moon, and Y. H. Lee, "IEEE1394/UPnP Software Bridge," IEEE Transaction on Consumer Electronics, Vol.51, No.1, pp.319-323, Feb., 2005.

[3] The HAVi Organization, "Havi Version 1.1 Specification," <http://www.havi.org>.

[4] Sun Microsystems, "JINI Architecture Specification," <http://www.sun.com/jini>.

[5] Echelon Co., "LonTalk Protocol Specification version 3.0," 1994.

[6] UPnP Forum, Universal Plug and Play Forum, <http://www.upnp.org>.

[7] C. Peng and P. Vuorimaa, "Digital Television Application Manager," ICME 2001, pp.685-688, 2001.

[8] ATSC, DTV Application Software Environment Level 1 (DASE-1), Nov., 2002.

[9] HAVi, Specification of the Home Audio/Video Interoperability (HAVi) Architecture Chapter 8 - Level 2 User Interface Version1.1, May, 2001.

[10] Bart C., Jon C., Bill F., Linda K., David R., James V., and Tao Y., "Java TV API Technical Overview," The Java TV API Whitepaper Version 1.0, Nov., 2000.

[11] ATSC, DTV Application Software Environment Level 1 (DASE-1), Nov., 2002.

[12] ETSI, TS 101 812: DVB Multimedia Home Platform (MHP) Specification 1.0.3, Jun., 2003.

[13] CableLabs, OpenCable Specification Home Networking Protocol, Jun., 2006.

[14] ETSI, TS 102 819: DVB Globally Executable MHP version 1.0.2 (GEM 1.0.2), Oct., 2005.

[15] ATSC Standard A/65B, Program and System Information Protocol (PSIP), Mar., 2003.

[16] DAVIC Standard 1.4.1p9, DAVIC 1.4.1 Specification Part 9, Complete DAVIC Specifications, Jun., 1999.

[17] Digital Video Broadcasting Project, <http://www.dvb.org>.

[18] ISO/IEC Standard 13818-6, Information technology - Generic coding of moving pictures and associated audio information: Extensions for Digital Storage media Command and Control, 1998.

[19] Digital Living Network Alliance, <http://www.dlna.org>.

[20] UPnP Forum, UPnP Device Architecture 1.0, Dec., 2003.

[21] UPnP Forum, UPnP AV Architecture 0.83, Jun. 2002.

[22] CableLabs, OpenCable Home Networking Protocol, Jun., 2006.



배 유 석

e-mail: baeyoung@etri.re.kr
 1995년 경북대학교 컴퓨터학과(학사)
 1997년 경북대학교 컴퓨터학과(석사)
 1997년~현재 한국전자통신연구원
 선임연구원
 관심분야: 데이터방송 미들웨어, 홈네트워크
 미들웨어, IPTV, 유비쿼터스
 컴퓨팅



오 봉 진

e-mail: bjoh@etri.re.kr
 1993년 부산대학교 전산학과(학사)
 1995년 부산대학교 전산학과(석사)
 1995년~1996년 시스템공학연구소 연구원
 1997년~현재 한국전자통신연구원
 선임연구원
 관심분야: 데이터방송 미들웨어, 홈네트워크 미들웨어, Java,
 편제형 컴퓨팅



문 경 덕

e-mail: kdmoon@etri.re.kr
 1992년 한양대학교 전산학과 졸업 (석사)
 2005년 ICU 전산학과 졸업 (박사)
 1992년~1996년 시스템공학연구소
 선임연구원
 1997년~현재 한국전자통신연구원
 책임연구원
 관심분야: 홈네트워크 미들웨어, 데이터방송 미들웨어, 액티브
 네트워크, 편제형 컴퓨팅



김 상 옥

e-mail: swkim@cs.knu.ac.kr
 1981년 서울대학교 컴퓨터학과(석사)
 1989년 서울대학교 컴퓨터학과(박사)
 1988년~현재 경북대학교 컴퓨터학과
 교수
 관심분야: 모바일 멀티미디어 컴퓨팅,
 멀티미디어 콘텐츠 제작, 인간과
 컴퓨터 상호작용, DMB 시스템