

碩士學位論文

JavaTV API를 이용한
디지털 TV 셋톱박스 미들웨어의 구현

지도교수 : 이 승 룡

慶熙大學校 産業情報大學院

電子工學科

이 세 우

2002年 8月 日

JavaTV API를 이용한 디지털 TV 셋톱박스 미들웨어의 구현

지도교수 : 이 승 룡

이 論文을 工學 碩士學位 論文으로 提出함

慶熙大學校 産業情報大學院

電子工學科

李世雨

2002年 8月 日

李世雨의 工學碩士學位 論文을 認准함

주심교수 印

부심교수 印

부심교수 印

慶熙大學校 産業情報大學院

2002年 8月 日

국문요약

JavaTV API를 이용한 디지털 TV 셋톱박스 미들웨어의 구현

이 세 우
전자 공학과
경희대학교 산업정보대학원

디지털 TV는 고화질, 다채널 방송 프로그램을 제공함과 동시에 디지털 TV용 응용프로그램을 제공하여 다양한 형식으로 부가적인 방송 서비스를 제공할 수 있다. 디지털 방송 시청자는 단 방향의 단순한 영상 및 음성 서비스를 제공받는 것뿐만 아니라, 방송프로그램 관련 응용프로그램, 독립형 응용프로그램, 프로그램 안내 응용프로그램, 광고 응용프로그램 등의 여러 형식의 부가정보 서비스를 제공 받을 수 있다.

이러한 부가정보 서비스는 MPEG-2 TS를 통해 영상, 음성 정보와 함께 전송되어 지는 응용프로그램을 디지털 TV 셋톱박스에 다운로드한 후 실행되어 제공될 수 있으며, 응용프로그램의 내용과 디지털 TV 셋톱박스의 기능에 따라 상향채널을 이용한 대화형 서비스도 가능하다.

그러나, MPEG-2 TS를 통해 전송되어 지는 응용프로그램은 실행되는 환경에 해당하는 디지털 TV 셋톱박스의 제품이 다양하여 각각의 플랫폼의 차이로 인해 정상적인 실행이 어렵다. 또한 방송네트워크 프로토콜이 다양하여 전송단과 동일한 프로토콜을 수신단에서 사용하여야 한다.

본 논문에서는 이를 해결하기 위해 플랫폼 독립적인 동작을 보증하고 응용프로그램에게 방송프로토콜의 추상화를 제공하는 JavaTV API를 이용하여 디지털 TV 셋톱박스의 미들웨어를 구현하였다.

본 논문에서 구현한 디지털 TV 셋톱박스 미들웨어는 유럽방식으로 알려진 DVB 규격에 맞게 구현되었고, 차후 ATSC나 ARIB규격에 맞는 구현이 가능하며, 각 규격과 플랫폼에 맞는 구현을 용이하게 하는 틀셋이나 프레임웍의 개발이 가능하다.

주요어 : MPEG-2 TS, 디지털 TV 셋톱박스 미들웨어, JavaTV API, DVB

<차 례>

<차 례>	1
<표 차 례>	2
<그림 차례>	3
1. 서 론	5
2. 관련연구	7
2.1 미들웨어 표준 규격	7
2.2 상용 미들웨어 제품	9
3. 관련 배경 기술	11
3.1 디지털 TV 의 구성	11
3.2 MPEG-2 TRANSPORT STREAM.....	13
3.3 DVB 표준규격	17
3.4 디지털 TV 용 응용프로그램	20
3.5 JAVATV API.....	22
4. 구현환경.....	24
4.1 셋톱박스 하드웨어 환경	24
4.2 셋톱박스 소프트웨어 환경	25
4.3 개발 환경	26
5. 구현내용.....	27
5.1 시스템 구조	27
5.2 DVB SI 구현부	29
5.3 데이터 방송 구현부	31
5.4 JAVATV API 구현부	33
5.5 실험 결과 및 고찰	39
6. 결론 및 향후과제	43
참고문헌.....	45

<표 차례>

표 3-1. PID 테이블	15
표 3-2. 서비스 정보 테이블	18
표 5-1. SIHANDLLER 의 SI 데이터 처리 알고리즘.....	30
표 5-2. 객체 캐리셀 분석 알고리즘.....	33

〈그림 차례〉

그림 2-1. MHP 프로파일	8
그림 2-2. DASE 구조	9
그림 3-1. 일반적인 DTV 수신기 모델.....	13
그림 3-2. 일반적인 셋톱박스의 소프트웨어 구성	14
그림 3-3. TRANSPORT STREAM SYSTEM DIAGRAM	14
그림 3-4. 프로그램 특성 테이블을 이용한 역 다중화	15
그림 3-5. 캐러셀 개념도	17
그림 3-6. 기타 영역과의 관련성	20
그림 3-7. JAVATV 를 이용하는 디지털 TV 수신기	23
그림 4-1. 셋톱박스의 하드웨어 구성도	25
그림 4-2. 구현 목표 셋톱박스	26
그림 4-3. 셋톱박스의 소프트웨어 구성	26
그림 4-4. 개발환경 구성도	27
그림 5-1. 구현된 디지털 TV 셋톱박스 전체 구성도	28
그림 5-2. DTV 셋톱박스 미들웨어 구현 구조	29
그림 5-3. DVB SI 구현부 구조.....	30
그림 5-4. 객체 캐러셀 계층도	32
그림 5-5. 데이터 캐러셀 구조도	32
그림 5-6. 객체 캐러셀 분석 과정	33
그림 5-7. SI 패키지간의 의존성	35
그림 5-8. SERVICECONTEXT 의 상태 전이	37
그림 5-9. JMF LITE 구현	38
그림 5-10. XLET 생명주기 모델	39
그림 5-11. JAVATV 기본 화면	40
그림 5-12. 응용프로그램 메뉴	40
그림 5-13. 선수명단 응용프로그램	41
그림 5-14. 신상 명세 제공 응용프로그램	41

그림 5-15. 프로그램 가이드 응용프로그램	42
그림 6-1. JAVATV 포팅 툴셋의 개념도	44

1. 서 론

디지털 방송은 기존의 방송과는 달리 시청자와 더불어 호흡하는 방송으로 단방향의 정보 전달 뿐 만 아니라 상향채널(Return Channel)을 이용한 실시간 시청자참여, TV전자상거래 (T-commerce), 홈뱅킹(Home Banking), 전자우편, 퀴즈 등을 할 수 있으며, 게임, 증권정보, 날씨정보 등의 다양한 부가정보를 제공하는 응용프로그램을 제공할 수 있는 방송이다. 즉, 시청자가 원하는 정보를 방송으로부터 얻을 수 있고, 방송에 직접 참여하여 의견을 제시할 수도 있으며, 물건구매도 할 수 있는 방송이다. 또한, 디지털 방송의 특징은 다채널/다매체 방송으로 기존 아날로그 방송과는 달리 고화질/고음질 고품질 서비스를 제공하는 것뿐만 아니라 방송의 다양화, 개인화, 양방향화, 네트워크화를 가지는 다기능성을 가진다[18].

이렇게 방송 서비스는 관련 산업 전반에 걸쳐 새로운 고부가가치를 창출할 것으로 예상되어, 세계 각국에서 서비스 실현을 위한 규격화 작업을 진행하고 있다. 대표적인 디지털 방송 규격으로는 유럽의 DVB, 미국의 ATSC, 일본의 ARIB등이 있으며, 각국에서는 이러한 방송 규격에 맞게 방송 서비스를 시작하고 있다.

썬 마이크로 시스템즈(Sun Microsystems)는 디지털 TV를 위한 응용프로그램의 개발과 관련하여 Java 언어를 이용하여 Platform 및 방송 네트워크 기술에 독립적으로 TV 수신기와 셋톱박스를 제어할 수 있는 응용프로그램의 개발을 지원하는 Personal Java 기반의 JavaTV API 를 정의하고 있다.[11]

디지털 TV 수신기를 구성하는 플랫폼에는 수 많은 하드웨어와 실시간 운영체제(RTOS)가 존재하며 이들의 규격 또한 천차만별일 수 밖에 없다. 다양한 플랫폼의 디지털 TV 수신기를 모두 만족하는 디지털 TV용 응용프로그램이 방송되어 수신된 후 셋톱박스에서 정상적으로 동작되는 것을 보증하기란 불가능에 가깝다고 볼 수 있다. 그러나, JavaTV API는 디지털 TV 응용프로그램 개발자 및 콘텐츠 제공자에게 디지털 TV 수신기의 하드웨어나 운영체제 그리고 디지털 방송에 사용되는 방송기술과 관한 내용을 전혀 알지 않아도 되게 한다. 또한 JavaTV API에 충실하게 작성된 어떠한 응용프로그램도 JavaTV API가 구현된 디지털 TV 수신기에서 정상적인 실행을 보증한다[11].

본 논문에서는 JavaTV API를 이용한 디지털 TV 셋톱박스 구현을 통해 플랫

폼 독립적으로 디지털 TV용 응용프로그램이 동작될 수 있도록 하며, 따라서 응용프로그램 개발자나 콘텐츠 제공자가 수신자의 수신기 플랫폼을 전혀 염두에 두지 않게 하게 한다.

본 논문에서는 JavaTV API를 이용하여 DVB 표준을 만족하는 DTV 셋톱박스의 소프트웨어를 구현하고, JavaTV API에 충실한 응용프로그램을 구현하여 셋톱박스에서의 동작을 확인하고 그 내용을 소개한다. 나아가서 여러 상이한 플랫폼을 갖는 셋톱박스에 JavaTV API의 구현을 용이하게 할 수 있도록 JavaTV API 포팅에 관한 방법에 접근한다.

본 논문의 구성은 다음과 같다. 2장에서는 관련연구로서 현존하는 디지털 TV 셋톱박스 미들웨어 제품 및 표준들에 대해 알아보고, 3장에서는 본 연구의 기반이 되고 있는 디지털 TV의 구조, 환경 및 표준에 대해서 소개하고, 본 연구에서 플랫폼 독립성을 지원하기 위해 구현하는 JavaTV API에 대해 설명한다. 4장에서는 본 논문에서 구현하는 디지털 TV 셋톱박스의 구현환경을 소개한다. 5장에서는 디지털 TV 구현 과정 및 관련 내용을 설명하고, 그 구현 결과를 소개하고 결과에 대해 고찰한다. 6장에서는 JavaTV API 포팅에 관한 접근을 소개하면서 결론 및 향후 과제를 제시하고 논문을 마친다.

2. 관련 연구

디지털 방송은 영상과 음성 이외에 응용프로그램을 이용한 다양한 서비스를 제공할 수 있다는 장점을 갖으며, 디지털 방송이 아날로그 방송과 차별화 될 수 있는 가장 큰 특징이라 할 수 있다.

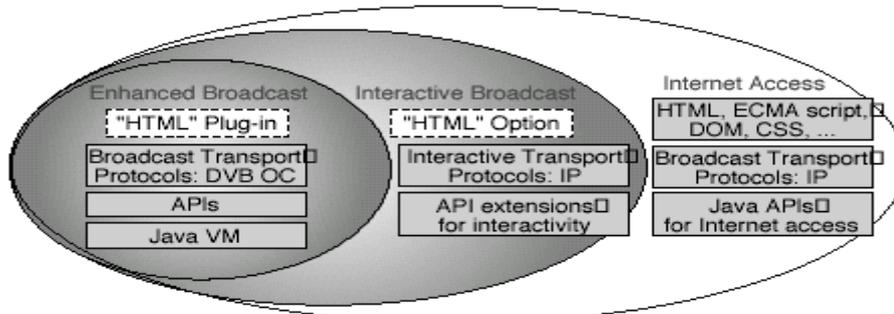
그러나, 디지털 TV 셋톱박스의 하드웨어 및 운영체제가 제조업체마다 서로 상이하여 데이터 채널을 이용해 다운로드된 응용프로그램 코드의 정상적인 동작의 보장은 불가능하다. 따라서 셋톱박스에는 부가서비스를 제공할 응용 프로그램을 상이한 셋톱박스 플랫폼에서 동작할 수 있게 해 주는 미들웨어의 필요성이 요구된다. 이 장에서는 현존하는 디지털 TV 미들웨어 제품들과 표준 규격들에 대해 알아본다.

2.1 미들웨어 표준 규격

2.1.1 MHP

MHP는 1996년 EBU의 UNITEL프로젝트에서 제안되었으며, 1997년 DVB CM(Digital Video Broadcasting Commercial Module)내에 DVB-MHP 특별위원회가 결성되면서 본격적으로 활동을 시작하였다.

DVB-MHP는 상업적인 사항들에 관심을 갖고 사용자와 시장 요구사항을 고려하여, 가정용 단말기인 STB, TV, PC와 그 주변장치, 그리고 가정용 디지털 네트워크를 모두 수용하는 수신기에서 향상된 방송(Enhanced Broadcasting), 양방향 서비스(Interactive Service), 그리고 인터넷 액세스(Internet Access)등의 서비스가 가능하도록 하는 것을 목표로 하고 있다.



[그림 2-1] MHP 프로파일

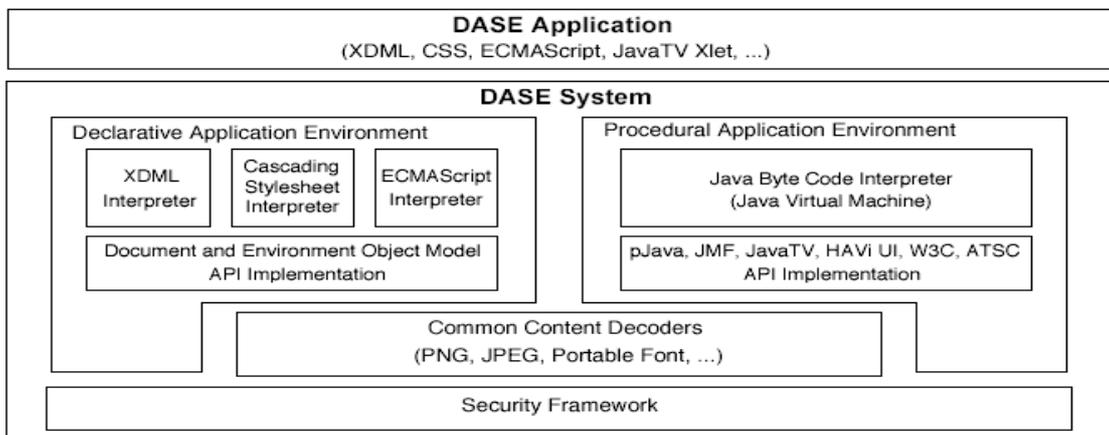
MHP는 시스템 자원, 시스템 소프트웨어, 애플리케이션의 3개의 계층으로 구분 될 수 있다. 일반적으로 시스템 자원이라 함은 MPEG 처리장치, I/O 장치, CPU, 메모리 등을 의미한다. 응용프로그램은 시스템 자원을 직접적으로 접근하지 못하며, 대신 시스템 소프트웨어가 응용프로그램과 시스템 자원간의 미들웨어 역할을 수행하고 응용프로그램 관리자(Application Manager)를 통해 MHP와 그 위에서 작동되는 응용프로그램을 제어한다.

MHP 표준의 핵심은 H/W 독립적인 자바 기반의 가상기계(Java virtual machine)의 사용이다. 응용프로그램을 구현하기 위해 다양한 기능을 제공하는 MHP API들은 자바 가상 기계상에서 작동하는 응용프로그램과 MHP 단말기와의 인터페이스를 제공한다. 또한 MHP API는 DVB-SI, DSMCC 등과 같은 전송 프로토콜, 응용프로그램에 대한 정보를 제공하기 위한 신호처리 프로토콜 및 보안성을 제공하기 위한 보안 프로토콜 등을 함께 제공한다.

MHP는 pJava를 기반으로, JavaTV API, JMF API, HAVi-UI API 등을 포함한다[17].

2.1.2 DASE

DASE(DigitalTV Application Software Environment)는 응용환경을 선언형 응용환경과 절차형 응용환경으로 나누어 규정한다. 선언형 응용환경은 xHTML 해석기, CSS 해석기, ECMAScript 해석기 및 응용프로그램에서 마크업 언어에 기반 한 콘텐츠 내용을 변경, 추가, 삭제할 수 있도록 정의하는 문서의 공통 모델인 DOM(Document Object Model)API의 구현으로 이루어지고, 절차형 응용환경은 Java 응용프로그램을 실행할 수 있는 Java 가상머신으로 이루어진다.



[그림 2-2] DASE 구조

특히 API는 pJava API, JMF API, JavaTV API, HAVi API, DAVIC API 및 DASE 고유의 API 등으로 이루어진다. [그림 2-2]는 DASE의 전체구조를 나타낸 것이다[15].

2.1.3 ATVEF

ATVEF(Advanced TV Enhanced Forum)는 HTML 기반의 향상된 TV용 프로토콜의 규격화를 위해 방송사, 가전사, PC 제조업자, 소프트웨어 개발업자 등이 모여 만든 단체이며 산업체 표준 제정을 목표로 하고 있다.

전송매체는 아날로그, 디지털 모두 대상으로 하고, 의무적인 콘텐츠 규격으로 HTML 4.0을 사용하도록 되어 있으며 그외 CSS1, ECMA Script/DOM0(JavaScript1.1) 등을 채택하고 있다. 전송 규격은 인터넷 전송 프로토콜의 변형인 UHTTP(Unidirectional HTTP)를 사용한다.

전송형태는 리턴채널 유무에 따라 두 가지로 정의하고 있다. 리턴채널이 있는 경우 트리거만 제한된 방송대역폭으로 전달하고 콘텐츠는 양방향 인터넷 전송로를 사용하며, 리턴 채널이 없는 경우 트리거, 콘텐츠 모두 방송 채널을 통해 전송한다[16].

2.1.4 MHEG

MHEG(Multimedia & Hypermedia information coding Expert Group)은 다양한 미디어를 사용해서 응용과 서비스들이 상화 호환될 수 있도록 멀티미디어 및 하이퍼미디어 정보 객체를 코드화 하여 표현하기 위한 표준이다. 그리고 이 객체들은 멀티미디어 및 하이퍼미디어 프리젠테이션 구조를 정의하고 있다[17].

MHEG는 현재 MHEG-5까지 개발되었고, 영국의 디지털 지상파 방송인 BBC와 ONdigital에서 사용 중이고 HTML보다도 디코더가 단순하다는 장점이 있다.

2.2 상용 미들웨어 제품

2.2.1 OpenTV

OpenTV는 OpenTV사에 의해 제작된 디지털 방송 셋탑박스에 특화된 미들웨어

엔진으로, 적절한 Royalty 가격과 여러 RTOS를 지원하는 Portability로 인해 많이 보급되고 있다. 현재 유럽 및 미국 등지에 20개 종류 이상의 수신기에서 200만 명 정도의 가입자를 확보하고 있다.

OpenTV는 O-Code라는 Byte Code를 실행하는 C 언어 기반의 독자적인 가상 기계 Engine을 사용하여 CPU 비 의존적인 어플리케이션이 개발 가능하다. O-code를 기반으로 각종 대화형 서비스를 개발 할 수 있어서, 현시점에서 상업적으로 대화형 서비스를 구축할 수 있는 가장 유력한 솔루션이다.

OpenTV는 셋탑박스의 미들웨어 엔진 이외에도 Open Author라는 저작도구도 제공하고 있고, Head End의 송출 시스템에 대한 솔루션도 함께 제공하고 있다[13].

2.2.2 MicrosoftTV

MicrosoftTV는 자사의 운영체제 및 미디어 기술을 활용해 HTML, 자바스크립트 위주의 미들웨어를 제공한다. Microsoft사는 TV Platform Adaptation Kit(TVPAK)를 텔레비전 업계의 소프트웨어 솔루션으로 제시하였는데, TVPAK은 Microsoft WindowsCE 와 WebTV Networks를 기반으로 네트워크 관리자에게 인터넷, 인터랙티브 텔레비전, 전자메일, 채팅, pay-per-view, VOD, 멀티플레이어 게임 등의 기능을 제공한다. TVPAK은 기존의 텔레비전과 ATVEF 표준안과 연계되도록 만들어 졌다.

Microsoft TV는 인터랙티브 텔레비전과 셋탑박스를 위한 클라이언트 소프트웨어 플랫폼이다. Microsoft TV Server는 Microsoft TV 클라이언트쪽의 제품을 관리하고 발전시키기 위해 Microsoft와 WebTV네트워크에 있는 서버 소프트웨어이다. 서버 소프트웨어는 Microsoft Windows2000 Server, Microsoft BackOffice와 WebTV 기술을 기반으로 하고 있다[14].

2.2.3 MediaHighway

MediaHighway는 Canal+ Technologies사가 1996년 디지털 대화형 TV 서비스를 제공하기 위해 개발한 제품으로 독자적인 가상 기계를 기반으로 상부의 컨텐츠 인터페이스를 추가함에 따라 HTML, MHEG-5, Java등의 컨텐츠를 흡수할 수 있는 구조를 갖는다. 현재 프랑스, 스페인, 영국, 폴란드, 네델란드, 덴마크, 브라질, 말레이시아 등의 방송사에서 사용되고 있다.

3. 연구 배경 기술

디지털 TV는 기존의 아날로그 방식의 TV와 비교하여 방송국에서 발신하는 단순한 오디오 비디오에 치중한 일방적인 TV프로그램에 국한되지 않고, 다채널, 고화질, 다양한 부가서비스를 제공하는 응용프로그램을 제공한다. 예를 들어 디지털 TV 시청자는 야구 중계를 시청하면서, 투수의 전적이나, 시합 중인 팀의 시합 일정 등의 내용을 필요 따라 응용프로그램을 통해 제공 받을 수 있으며, 중계내용 또한 관중석, 타석, 외야 등의 여러 시각에서 촬영한 내용을 시청자가 선택하여 시청할 수 있게 제공 될 수 있다.

이와 같은 디지털 TV의 기능들은 모든 데이터들이 디지털화 되어 전송되어지고, 전송된 데이터들을 일련의 작업을 통해 수행 할 수 있는 디지털 TV 수신기가 존재해야 하며, 이들 디지털 TV 수신기들은 전송단이 따르는 규격을 만족해야만 한다.

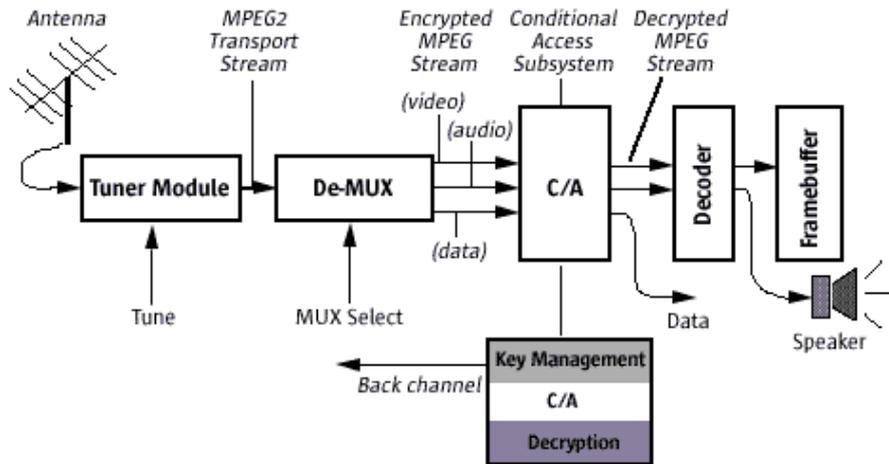
따라서 이 장에서는 일반적인 디지털 TV의 구성과 디지털 TV의 전송되는 데이터의 표준인 MPEG-2 TS(Transport Stream), 본 논문의 구현에서 따르는 DVB표준, 디지털 TV용 응용프로그램, 그리고 JavaTV API에 대해서 알아 보겠다.

3.1 디지털 TV의 구성

디지털 방송을 수신하기 위해서는 디지털 신호를 처리할 수 있는 별도의 수신기를 필요로 한다. 디지털 신호는 아날로그 신호에 비해 보다 다양한 콘텐츠를 방송 할 수 있다. 디지털 방송은 디지털화된 영상 및 음성 데이터와 함께 다양한 형태의 정보를 포함 할 수 있다.

[그림 3-1]은 일반적인 디지털 TV 수신기의 모델로서, 수신기 내에서 어떤 과정을 거쳐 데이터와 오디오 그리고 비디오가 분리되는가를 나타내고 있다. 우선 튜너가 원하는 주파수를 선택하면 이 결과가 디모듈레이터(de-modulator)를 거쳐 하나의 비트 스트림인 MPEG-2 TS를 얻어 낸다. 다음은 디멀티플렉서(de-multiplexor)인데, 이것은 멀티플렉서로 묶어 놓은 MPEG-2 TS로부터 비디오, 오디오, 데이터 서비스 정보 등에 해당하는 스트림을 분리한다. 다음은 수신제한 시스템(C/A -Conditional Access System)으로 보통 유료 방송은 비디오를 암호화해서 보내고 비용을 지불한 사람들만 복호

화해서 볼 수 있도록 하는데, 수신제한 시스템이 사용자의 권한을 점검해서 비디오를 복호화하는 역할을 한다. 이렇게 해서 나온 각각의 비디오, 오디오, 데이터 등은 각자 적합한 디코더(Decoder)를 거쳐 화면에 보여진다. 이때 비디오는 디지털 환경에서 다시 점들의 집합으로 디코딩된 후 비디오를 위한 프레임 버퍼에 그려진다.



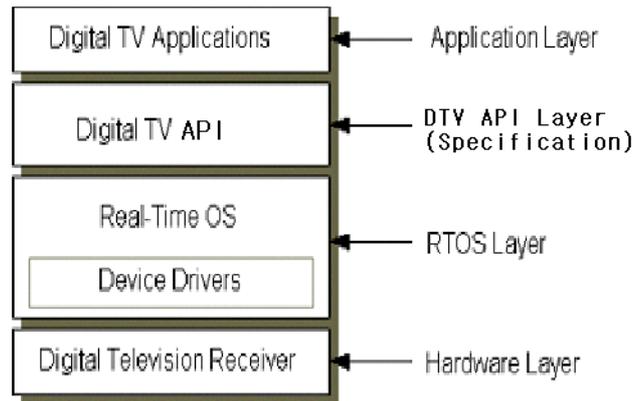
[그림 3-1] 일반적인 DTV 수신기 모델

각 방송채널은 하나의 비트 스트림이라고 볼 수 있는데, 이 하나의 비트 스트림에는 각자 정해진 속도로 비디오, 오디오, 그 외 데이터를 적절히 섞어서 보내고 나중에 정확히 다시 분리하는 것이 중요하다. MPEG-2 표준에는 방송환경과 같이 에러가 잦은 환경에서 비디오, 오디오를 한 스트림에 묶어 넣고 다시 분리하는 규약으로 MPEG-2 TS(Transport Stream)를 지정하고 있다[19].

MPEG-2 TS는 오디오, 비디오, 그리고 데이터를 포함하고 있는데, 그 중 데이터 부분은 서비스 정보와 방송 파일, 그리고 응용프로그램 코드 등으로 구성되어 있다. 이들 데이터들을 어떻게 표현하고 응용프로그램이 사용할 수 있게 하는가에 대한 정의로는 유럽의 DVB(Digital Video Broadcasting), 미국의 ATSC(Advanced Television Systems Committee), 일본의 ARIB(Association of Radio Industries and Business)를 대표적으로 들 수 있고 본 논문에서는 DVB규격을 선택하여 구현하고 있다.

디지털 TV 셋톱박스는 [그림 3-2]에서 보여지는 것과 같이 디지털 TV 수신 장치와 RTOS(Real-Time Operating System), 그리고 사용하는 방송규약을 구현하는 DTV API를 구성으로 갖으며, MPEG-2 TS로부터 전송되거나 셋톱박스

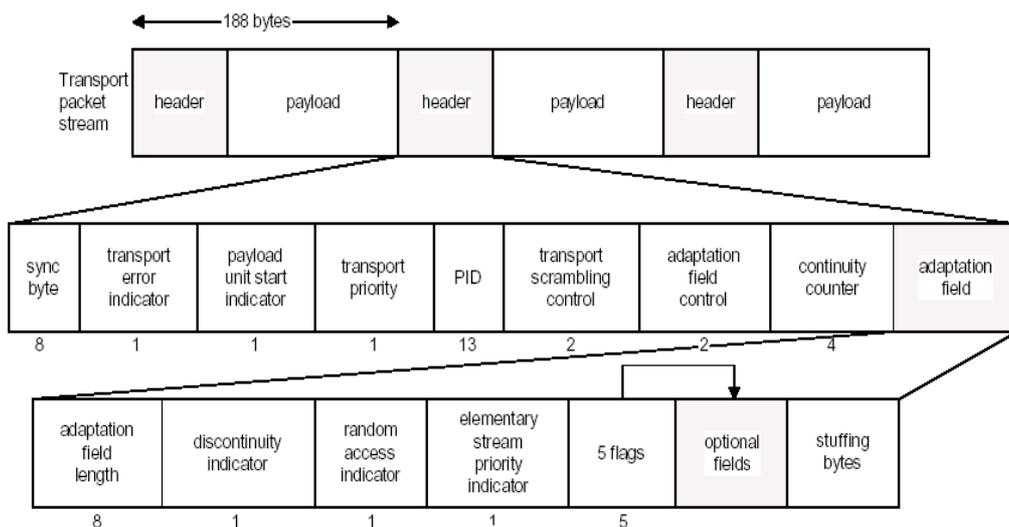
에 내장된 응용프로그램으로 구성된다. DTV API의 구현은 MPEG-2 TS로부터 전송된 데이터를 종류별로 분류해내는 디멀티플렉서(de-multiplexer)와 서비스정보를 분류해 내고 체계적인 구조를 갖게하는 SI-parser 및 SI Database, 그리고 Broadcast File System을 구성하게 하는 부분으로 나뉘어질 수 있다.



[그림 3-2] 일반적인 셋톱박스의 소프트웨어 구성

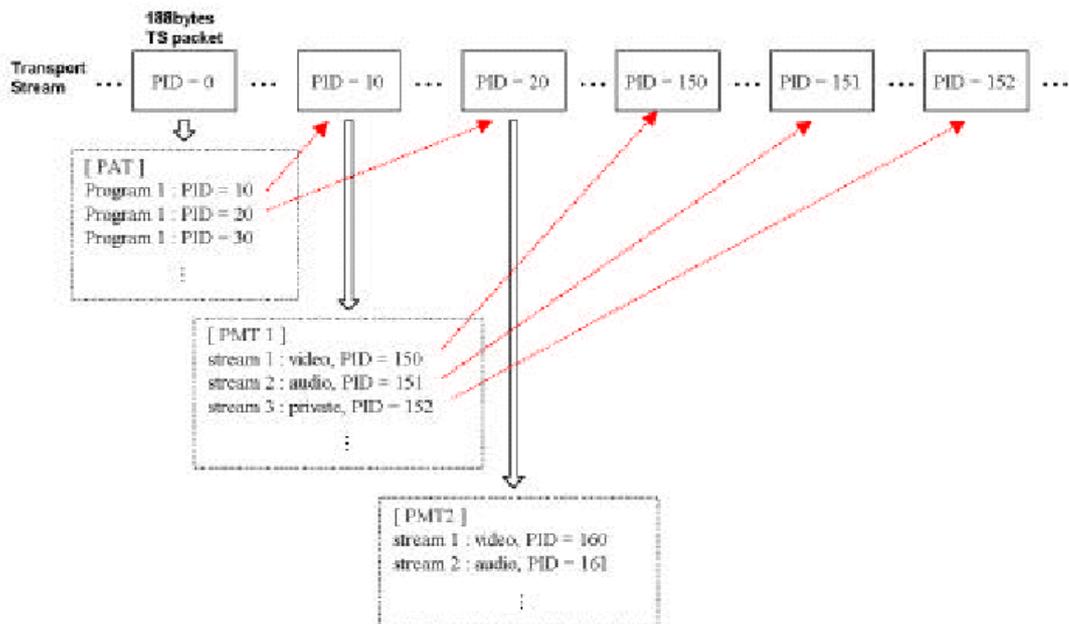
3.2 MPEG-2 Transport Stream

MPEG-2 시스템 표준에서는 영상 데이터와 음성 데이터 혹은 그 이외의 데이터들을 전송에 적합하도록 하나의 스트림으로 다중화하는 방법에 대해서 TS(Transport Stream)규격으로 규정한다[1].



[그림 3-3] Transport Stream syntax diagram

MPEG-2 TS는 [그림 3-3]에 보여지는 것처럼 188 바이트의 패킷들로 구성된다. 각 패킷은 4바이트의 고정 헤더와 가변길이 적응필드(adaptation field), 그리고 실제 전송하고자 하는 데이터를 갖는 유효부하(payload)로 구성된다. 고정헤더에는 전송 에러 지시자(transport_error_indicator), 연속 카운터(continuity_counter), 전송 우선도(transport_priority)등의 전송에 관련된 정보들과, 패킷 식별자(packet identifier, PID), 전송암호화 제어(transport_scrambling_control)등의 패킷이 전송하고 있는 데이터에 대한 정보가 포함되어 있다. 이 중에서 패킷식별자는 다중화된 TS를 역다중화하여 ES(Elementary Stream)들을 만들어 내는데 필요한 중요한 정보이다. 확장헤더는 그 용도에 따라서 크기가 가변될 수 있는데, 확장헤더의 유무는 고정헤더에 포함된 필드로부터 알 수 있다. 확장헤더에는 시스템의 동작 클럭을 동기화하기 위한 클럭참조값(Program Clock Reference, PCR)이외의 여러 정보 필드들과 TS 패킷의 길이를 188바이트로 맞추기 위한 채움바이트(stuffing_byte)가 삽입될 수 있다. 하나의 TS 패킷에서 고정헤더, 확장헤더를 제외한 나머지 부분이 실제로 전송되는 데이터인데, 부호화된 영상, 음성데이터와 사용자데이터 등이 이러한 데이터가 될 수 있다. TS가 포함하고 있는 데이터들을 종류별로 역다중화 하기 위해서는 각각의 데이터에 해당하는 패킷식별자(PID)를 알아내야 할 필요성이 생긴다. 이러한 이유로 TS 규격에서는 전송될 데이터의 패킷식별자 목록을 전달하기 위한 특별한 데이터 형식을 별도로 규정하고 있는데, 이것을 프로그램 특성 정보(Program Specific Information, PSI)라고 한다.



[그림 3-4] 프로그램 특성 테이블을 이용한 역 다중화

3.2.1 프로그램 특성 정보(PSI)

프로그램 특성 정보의 종류에는 프로그램 연결 테이블(Program Association Table, PAT), 프로그램 맵 테이블(Program Map Table, PMT), 네트워크 정보 테이블(Network Information Table, NIT), 조건부 접근 테이블(Conditional Access Table, CAT)이 있다. 이 중에서 네트워크 정보 테이블은 TS가 전달되는 네트워크에 대한 정보를 수록하는 테이블이고, 조건부 접근 테이블은 수신대상에 따라 TS에 수록된 데이터에의 접근을 제한하기 위해 사용되는 테이블이다. 그리고 나머지 두 테이블이 TS에 포함된 데이터들의 패킷 식별자를 목록화하여 전달하는데 사용되는 테이블이다. [그림 3-4]에 TS를 역다중화 하여 데이터를 추출해 내는 절차를 간략히 나타내었다.

프로그램 맵 테이블에는 하나의 프로그램을 구성하는 영상, 음성, 사용자 데이터들의 패킷식별자 목록을 수록하고 있으며, 그 자신도 유일한 패킷식별자를 가지고 있어서 다른 데이터들과 구별된다. TS에는 여러 프로그램들이 동시에 수록될 수 있으므로 수록된 프로그램의 개수와 일치하는 프로그램 맵 테이블 또한 TS내에 포함되어 있어야만 한다. 따라서 프로그램 맵 테이블을 TS로부터 추출하기 위해서는 프로그램 맵 테이블의 패킷식별자 등을 목록화 하여 전송할 필요성이 생기게 된다. 이러한 역할을 담당하는 것이 프로그램 연결 테이블인데, 이것의 패킷식별자는 '0x0000' 으로 TS규격에 정의되어 있다. 이러한 규정에 따라 역다중화 과정에서는 먼저 패킷식별자가 '0x0000' 인 프로그램 연결 테이블을 찾아 프로그램 맵 테이블의 패킷식별자를 알아내고, 그로부터 프로그램 맵 테이블을 찾아서 하나의 프로그램을 구성하는 데이터들의 패킷식별자를 알아낸후, 각각의 데이터를 분류해 내게 된다. 역다중화 과정이 이런 방법으로 진행되기 때문에 방송과 같은 응용분야에서는 프로그램 연결테이블과, 프로그램 맵 테이블을 주기적으로 전송하도록 규정하고 있다.

Value	Description
0x0000	Program Association Table
0x0001	Conditional Access Table
0x0002 - 0x000F	Reserved
0x00010 ... 0x1FFE	May be assigned as network_PID, Program_map_PID, elementary_PID, or for other purposes
0x1FFF	Null packet
NOTE - The transport packets with PID values 0x0000, 0x0001, and 0x0010 - 0x1FFE are allowed to carry a PCR.	

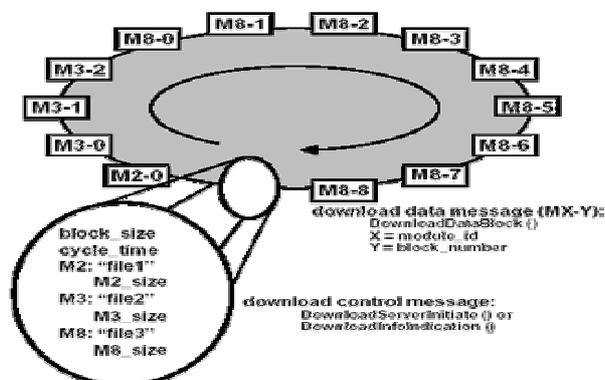
[표 3-1 PID 테이블]

영상 부호기를 통해서 만들어진 ES 데이터를 복호하여 화면에 출력하거나 음성 데이터와 동기를 맞추기 위해서는 특별한 시간정보가 필요한데, 이를 타임 스탬프(Time Stamp)라고 한다. 타임 스탬프에는 복호 타임 스탬프(Decoding Time Stamp, DTS), 표현 타임 스탬프(Presentation Time Stamp, PTS)의 두 가지 종류가 있는데, 복호 타임 스탬프는 데이터가 복호될 시간을 나타내고, 표현 타임 스탬프는 복호된 데이터가 표시될 시간을 나타낸다. 하지만 이러한 정보들은 ES 데이터에는 포함되어 있지 않기 때문에, 이러한 정보를 추가한 새로운 구조가 필요한데, 이것을 PES(packetized Elementary Stream)라 한다. 따라서 TS에 데이터를 수록할 때는, 먼저 ES를 PES로 만든 후, 이것을 다시 TS 패킷으로 만드는 과정을 수행하게 된다.

3.2.2 DSM-CC

DSM-CC(Digital Storage Media Command and Control)는 광대역 네트워크상에서 멀티미디어 서비스를 공급하기 위해 MPEG에서 제정한 규격이다. DSM-CC에서 대상이 되는 시스템은 클라이언트와 서버로 대변되는 사용자(User) 시스템과, 서비스, 자원 관리자(Service and Resource Manager, SRM)를 포함하는 네트워크 시스템으로 분류할 수 있다. DSM-CC에서는 이렇게 분류된 시스템 사이의 인터페이스를 규정하고 있는데, 사용자 시스템 사이의 인터페이스를 U-U(User-to-User), 사용자 시스템과 네트워크 시스템 사이의 인터페이스를 U-N(User-to-Network)이라 규정한다[2].

U-U인터페이스와 UN 인터페이스 사이에는 메시지(message)가 전달되는데 이러한 메시지들은 정형화된 메시지 헤더와 실제 전송되는 메시지로 이루어져 있다. U-U 메시지에서는 원격 작업 호출 메시지, 세션(session) 제어 메시지 등이 포함되며, U-N 메시지에는 네트워크 설정 메시지, 서비스, 자원 관리 메시지, 다운로드 메시지 등이 있다.



[그림 3-5] 캐러셀(Carousel) 개념도

이 중 데이터 방송(Data Broadcasting)과 관련되는 것은 데이터 캐러셀(Data Carousel), 객체 캐러셀(Object Carousel) 등이 포함되어 있다. 데이터 캐러셀은 데이터 모듈을 주기적으로 전송하는 메커니즘이며, 객체 캐러셀은 데이터 캐러셀을 통하여 DSM-CC U-U 파일과 디렉토리의 계층적 구조를 주기적으로 전송하는 메커니즘이다. [그림 3-5]은 캐러셀의 개념도를 보이고 있다.

3.3 DVB 표준

DVB(Digital Video Broadcasting)는 다양한 디지털 방송환경에서 더 많은 시청자에게 서비스할 수 있는 방안으로 유럽의 방송사, 가전사, 통신업자, 법제 기관 등이 참여한 프로젝트로써 현재, 세계 28개국의 200개 이상의 단체가 참여해 유럽을 넘어 세계 디지털 방송에 하나의 표준을 제시하여 ETSI(European Telecommunication Standardization Institute)에 의해 승인되었다.

DVB 표준의 주요 내용으로는 사용자가 선택한 서비스와 연결된 서비스 정보의 표현과 관련된 내용을 정의하는 서비스 정보(Service Information, SI) 부분과 MPEG-2 TS를 통한 방송환경에서의 데이터 전송과 관련된 내용을 정의하는 데이터 방송(Data Broadcasting)부분으로 나눌 수 있다.

3.3.1 Service Information(SI)

DVB-SI는 ISO/IEC 13818-1에서 정의 하고 있는 PSI를 기반으로 몇 개의 테이블을 추가 함으로서, 방송에서 사용될 전송체계, 내용, 그리고 스케줄 등의 정보를 표현하며, 이와 같은 테이블을 통틀어서 SI 테이블이라고 한다. MPEG2-TS의 PSI에는 멀티플렉싱된 스트림에서 다양한 스트림을 디멀티플렉싱하기 위한 테이블이 정의 되어 있고, DVB SI가 여기에 서비스와 이벤트를 식별하기 위한 정보를 추가했다[4].

[표 2-2]는 MPEG2-TS에서 정의하고 있는 PSI 테이블과 DVB-SI에서 추가적으로 정의하고 있는 테이블들과 각 테이블에 고유하게 부여된 패킷 식별자(PID)를 보여 주고 있다.

DVB에서 추가한 테이블들의 역할은 다음과 같다.

- BAT(Bouquet Association Table) : 부케(bouquet)의 이름과 부케에 속하는 서비스 목록 등 부케에 대한 정보를 표현한다.
- SDT(service Description Table) : 서비스 이름과 서비스 제공자 등 서

비스에 대한 정보를 표현한다.

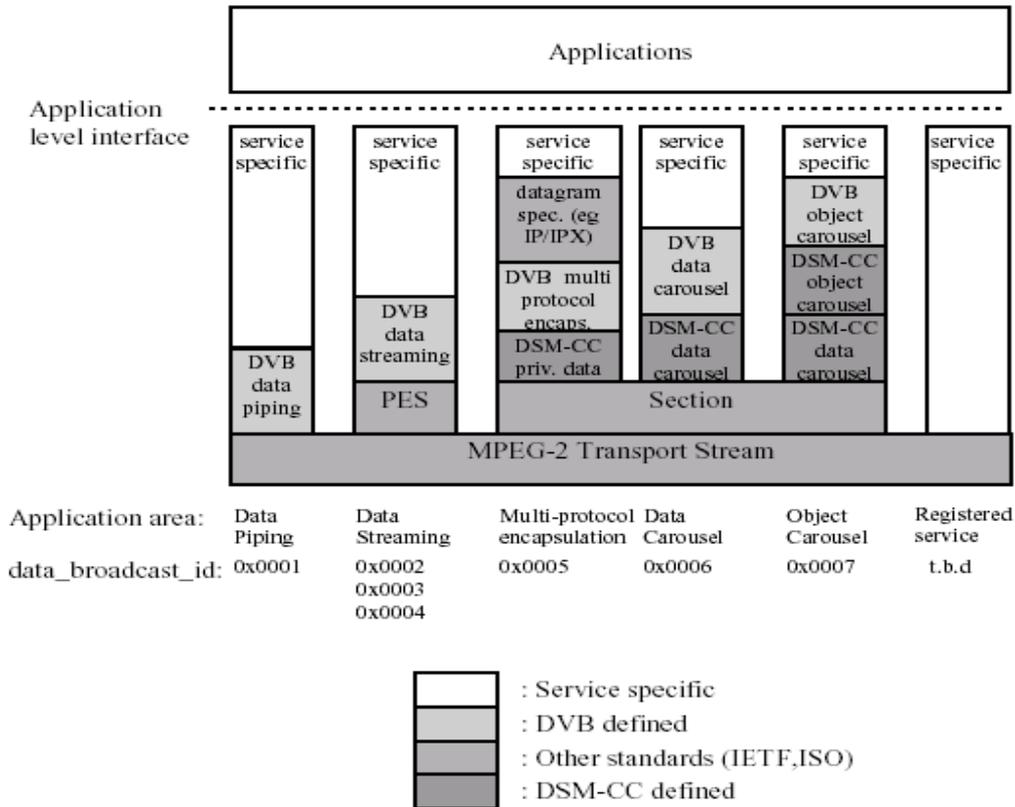
MPEG-2	DVB (mandatory)	DVB (optional)	
PAT PID=0x0000	NIT PID=0x0010	NIT PID=0x0010	Network Information
CAT PID=0x0001		BAT PID=0x0011	Bouquet Association
PMT PID=P	SDT PID=0x0011	SDT PID=0x0011	Service Description
TSDT PID=0x0002	EIT PID=0x0012	EIT PID=0x0012	Event Information
	TDT PID=0x0014	RST PID=0x0013	Running Status
	Time & Date	TOT PID=0x0014	Time Offset
		ST PID=0x0014	Stuffing

[표 3-2] 서비스 정보 테이블

- EIT(Event Information Table) : 이벤트 이름, 시작시간, 기간 등 이벤트 또는 프로그램에 대한 데이터를 표현한다.
- RST(Running Status Table) : 이벤트의 상태(Running 또는 Not Running)를 나타낸다.
- TDT(Time and Data Table) : 현재 시간과 날짜 관련 정보를 나타낸다.
- TOT(Time Offset Table) : 현재 시간과 날짜, 그리고 현지 시간과의 편차에 관련된 정보를 표현한다.
- ST(Stuffing Table) : 의미 없는 데이터 영역을 채우는데 사용된다.
- SIT(Selection Information Table) : 녹화된 스트림에서 DIT와 함께 부분적인 SI 정보를 나타내기 위해 사용된다.
- DIT(Discontinuity Information Table) : 녹화된 스트림에서 SIT와 함께 부분적인 SI 정보를 나타내기 위해 사용된다.

3.3.2 Data Broadcasting

DVB는 방송환경에서의 데이터 전송 표준으로 ISO/IEC 13818-6(DSM-CC, MPEG2 standard part 6)를 근간으로 삼고 있다. DVB는 데이터 전송 요구사항에 따라 서로 다른 네 가지 어플리케이션 영역으로 나누고 있다[6]. [그림 3-6]는 각 영역에서의 관련된 표준들과의 상관 관계를 보여 주고 있다



[그림 3-6] 기타 영역들과의 관련성

- **Data Piping**

Data piping을 위한 데이터 방송 규격은 DVB 호환 방송 네트워크를 통해서 단순하고 비동기적이며 종단 데이터 전송(end-to-end delivery)에 대한 요구사항을 지원한다. 데이터는 MPEG-2 TS 패킷의 유료부하에 직접 실어서 전송한다. DVB에서는 Data piping을 이용한 전송에 대한 규약은 별도로 되어 있지 않고 개별적으로 사용되질 수 있다.

- **Data streaming**

데이터 스트리밍을 위한 데이터 방송 규격은 DVB 호환 방송 네트워크를 통해서 비동기적이거나, 또는 동기화된 방법으로 스트리밍 지향적이고,

종단 데이터 전송(end-to-end delivery)에 대한 요구사항을 지원한다. 데이터는 MPEG-2 시스템에서 정의된 PES(Program Elementary Stream) 패킷을 이용하여 전송된다.

- **Multiprotocol encapsulation**

Multiprotocol encapsulation을 위한 데이터 방송 규격은 DVB 호환 방송 네트워크를 통해서 커뮤니케이션 프로토콜의 데이터그램 전송에 필요한 요구사항을 지원한다. 캡슐화된 데이터그램들은 MPEG-2에서 정의하는 DSM-CC 섹션을 이용하여 전송된다.

- **Data Carousel**

데이터 케로셀을 위한 데이터 방송 규격은 DVB 호환 방송 네트워크를 통해서 데이터 모듈의 주기적인 전송에 대한 요구사항을 지원한다. 모듈은 그 크기가 정해져 있으며, 데이터 카로셀로 부터 한번에 갱신, 추가, 삭제되어 질것이다. 모듈들은 필요에 따라 여러 그룹으로 나뉘어 질 수 있고, 그룹들은 슈퍼 그룹으로 또 다시 나뉘어 질 수 있다.

데이터 케로셀은 MPEG-2 DSM-CC에서 정의된 DSM-CC 데이터 케로셀을 통해 전달되어 진다.

- **Object Carousel**

객체 케로셀 규격은 DVB 호환 네트워크를 통한 주기적인 DSM-CC U-U 객체의 방송의 요구사항을 지원하기 위해 추가되었다. DVB 객체 케로셀의 전송은 MPEG-2 DSM-CC에 정의된 DSM-CC 데이터 케로셀과 객체 케로셀을 따른다.

3.4 디지털 TV용 응용프로그램

디지털 TV는 단방향의 비디오, 오디오로 구성된 단순한 방송 프로그램만 뿐만 아니라, 수신기에서 동작할 수 있는 응용프로그램을 방송채널을 통해 다운로드 받아서, 적절한 시간이나 사용자가 원하는 시간에 동작시켜 다양한 부가 서비스를 제공 받을 수 있게 한다.

디지털 TV에서 동작되는 DTV 응용프로그램은 크게 다음과 같이 나눌 수 있다.

3.4.1 EPGs(Eletronic Program Guides)

EPG의 본질적인 기능은 현재 또는 곧 방영될 프로그램에 대한 편성표이며,

시청자는 언제든지 EPG 응용프로그램을 선택하여 방영 스케줄을 알 수 있다. 기본적으로 EPG는 방송 프로그램의 스케줄을 표시하며, 방송 내용의 미리보기 등을 제공하며, 특정 프로그램의 선택에 맞게 채널을 바꾸어 주기도 한다.

EPG에 사용되는 정보들은 전송되는 서비스 정보(Service Information) 데이터를 데이터 베이스화 하여 필요한 정보들을 표현하는 것이다.

3.4.2 Program-Specific Applications

특정 방송 프로그램에 한정된 응용프로그램으로 특정 오디오, 비디오 프로그램의 서비스를 제공 받는 동안에만 사용될 수 있는 응용프로그램이다. 가장 전형적인 예로서 스포츠 프로그램을 시청하는 동안 경기중인 선수의 경력이나 전적 등의 서비스를 제공하는 응용프로그램이나 퀴즈 프로그램을 시청하는 동안 시청자 퀴즈를 응용프로그램을 통해 참여할 수 있는 예를 들 수 있다. 이와 같은 응용프로그램은 시청하고 있던 채널을 변경한 경우 동작이 정지되어야 하고, 다른 채널을 선택한 경우 선택된 채널에 맞는 응용프로그램이 동작되어야 한다는 요구사항을 갖는다.

3.4.3 독립실행(Stand-alone) 응용프로그램

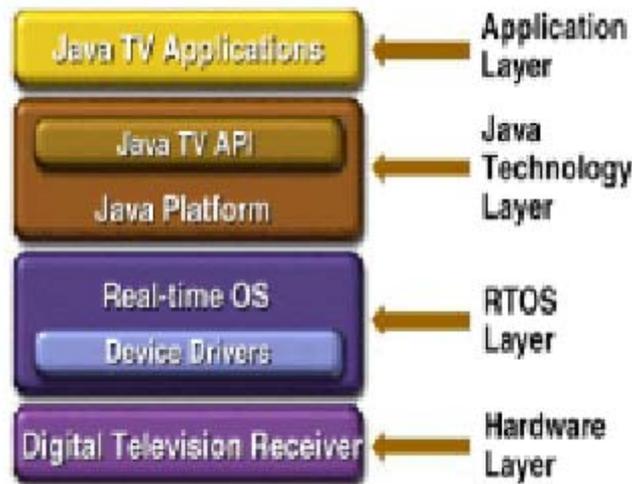
TV에서 방영되고 있는 방송 프로그램과는 전혀 관련 없이 시청자에게 정보를 제공하는 형태의 응용프로그램으로 물리적 채널 안에 하나의 가상 채널을 차지하고 동작한다는 뜻에서 가상 채널 서비스(Virtual Channel Service)라고도 한다. 흔히 뉴스, 날씨정보, 주식정보 등의 서비스를 제공하는 응용프로그램들이 이러한 형태를 띠고 있으며, 대표적인 예로 시청자가 연속극을 시청하면서 관심을 갖는 주식정보를 제공 받을 수 있도록 주식정보를 화면에 표시하는 응용프로그램을 들 수 있다.

3.4.4 광고 응용프로그램

광고 응용프로그램은 상업적인 오디오, 비디오 콘텐츠에 대한 내용을 갖는 응용프로그램이다. 일반적으로 광고시간 동안에만 실행되는 상당히 짧은 동작시간을 갖는다. 광고 응용프로그램의 실제 다운로드에는 광고시간이 시작되기 전에 미리 이루어져야 하며, 광고 시간이 지나면, 중지되고 버려진다. 광고 응용프로그램은 단순히 상업적인 콘텐츠의 제공에만 제한되지 않으며, 부가적인 기능을 추가 할 수 있다. 예를 들어 특정 광고의 시청 후 점수를 적립하여 무료전화 서비스나 물품 제공 등의 서비스를 추가 할 수 있다.

3.5 JavaTV API

방송채널을 통해 전송되는 응용프로그램이 디지털 방송 수신을 위해 존재하는 모든 셋톱박스의 플랫폼을 만족하여 정상적인 동작을 하게 하기 위해서는 디지털 방송 수신기에서 동작하고 TV 서비스와 관련된 서비스를 지원하는 미들웨어의 요구는 반드시 필요한 사항이 될 것이다. JavaTV API는 자바 플랫폼을 미들웨어로 사용하여 플랫폼 독립적인 응용프로그램의 환경을 만족시키고, 추상화된 인터페이스를 이용하여 응용프로그램이 동작되는 플랫폼이나 사용하는 방송기술에 대한 정보의 반영 없이 개발되고 동작될 수 있도록 지원한다[8].



[그림 3-7] JavaTV를 이용하는 디지털 TV 수신기

JavaTV API는 대화형 TV 서비스와 디지털 방송 수신기에서 동작하는 여러 종류의 소프트웨어를 개발하는데 사용되는 것을 목적으로 하는 Interface이며, JavaTV API가 제공하는 주요 기능은 다음과 같다.

3.5.1 서비스와 서비스 정보의 접근

JavaTV API는 방송 프로그램을 개별적인 서비스로 정의하고 Service Information API를 이용하여 서비스를 선택하는데 사용될 서비스 정보(SI)를 얻는 것을 지원한다.

SI database는 실행시간 동안 사용 가능한 서비스가 무엇인지에 대한 정보를 어플리케이션이 접근 할 수 있도록 제공하고, SI database의 접근은 SI Manager를 통해서 이루어 지도록 하고 있다. 어플리케이션이 관심있는 서비스의 탐색을 할 수 있도록 SI Manager는 필터링 연산을 제공한다. SI

database의 뷰는 탐색, EPGs, 그리고 MPEG2 전송을 제어하는데 사용되어 진다.

3.5.2 서비스 선택

Service Selection API는 표현될 서비스를 선택하는데 사용되어 진다. 선택에 대한 매카니즘은 서비스 컴포넌트(Service Component)에 의해 결정되어지고, 응용프로그램이 서비스의 일부분이라면 응용프로그램의 시작도 포함된다.

3.5.3 방송 Pipeline의 제어

수신기의 방송 Pipeline의 표현을 위해 JMF(Java Media Framework)를 사용한다. JMF는 데이터 원본(Data source)과 콘텐츠의 취급을 정의한다. JavaTV API는 Tunner-Demultiplexer-C/A 부분 시스템은 데이터 원본(Data source)로, Decoder-Frame Buffer-Audio/Video 출력은 Content Handler로 구분 짓는다.

3.5.4 방송 Data의 접근

디지털 데이터를 포함하는 스트림으로부터 Broadcast Data API는 방송 파일 시스템, 스트리밍 데이터, 그리고 Encapsulated IP 데이터에 대한 접근을 지원한다.

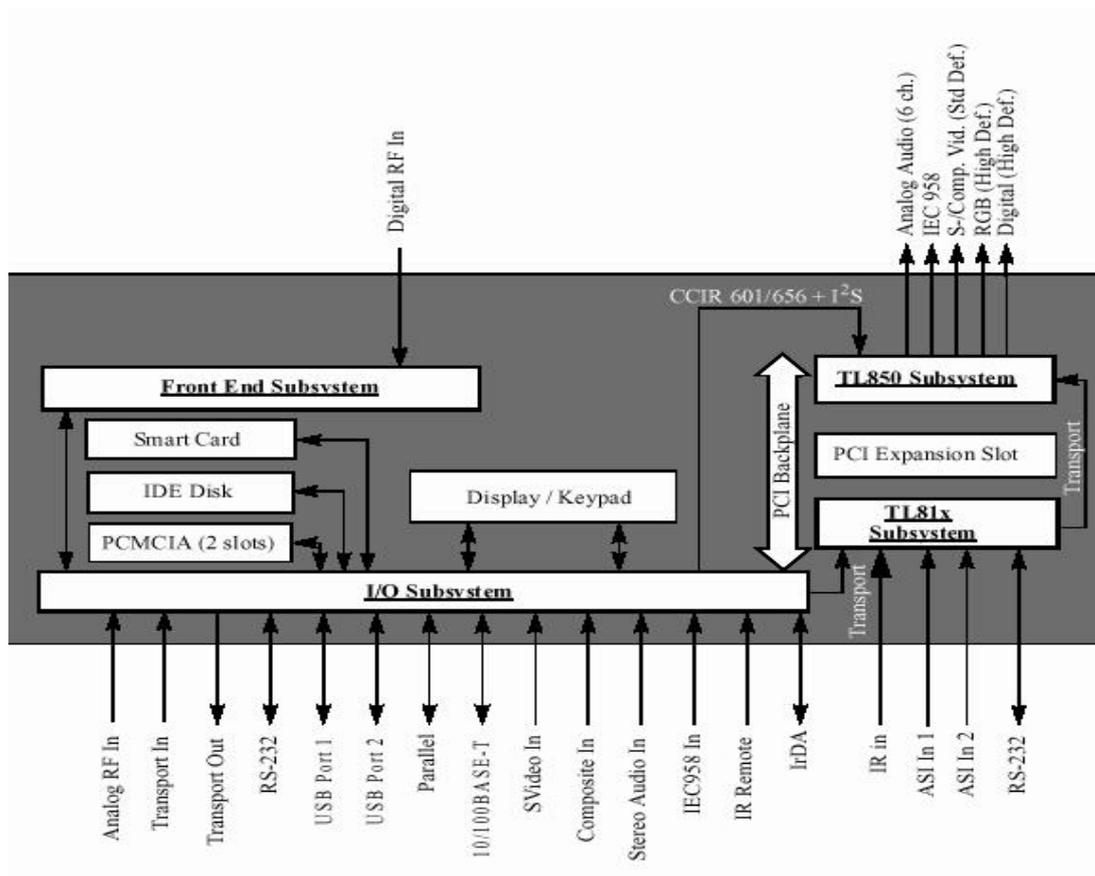
3.5.5 응용프로그램 생명주기의 관리

응용프로그램의 생명주기는 초기화, 상태변화, 소멸 등 일련의 단계들의 집합으로 정의 되고, JavaTV API는 Xlet Manager를 이용하여 수신기에서 동작되는 응용프로그램의 생명주기를 관리하도록 정의한다.

4. 구현 환경

4.1 셋톱박스 하드웨어 환경

셋톱박스 하드웨어는 [그림 4-1]에서와 같이 MIPS core QED 5231 200MHz 프로세서를 사용하며, DTV와 그래픽 프로세서로 TeraLogic사의 TL850, 32MB SDRAM, 1.2GB IDE 하드디스크, 10/100BASE-T 이더넷 인터페이스 등을 기본 구성으로 한다. 본 논문에서 구현하는 디지털 TV 셋톱박스의 하드웨어환경은 한국전자부품연구원(KETI)로부터 지원을 받았다.

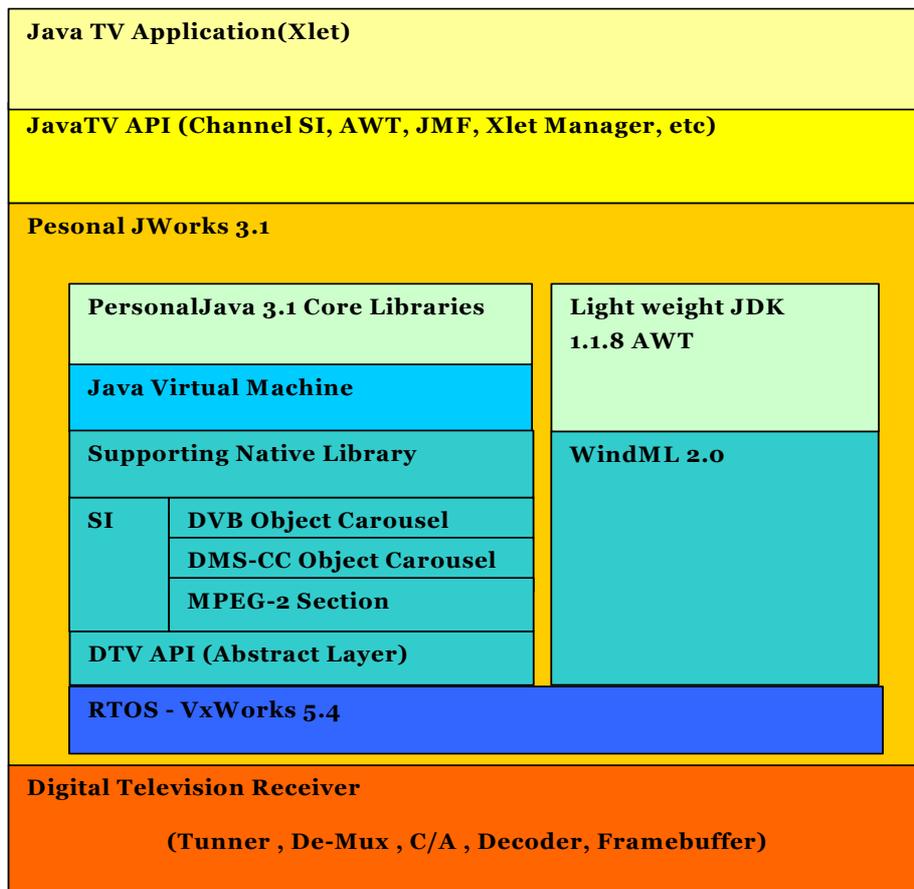


[그림 4-1] 셋톱박스의 하드웨어 구성도



[그림 4-2] 구현 목표 셋톱박스

4.2 셋톱박스 소프트웨어 환경

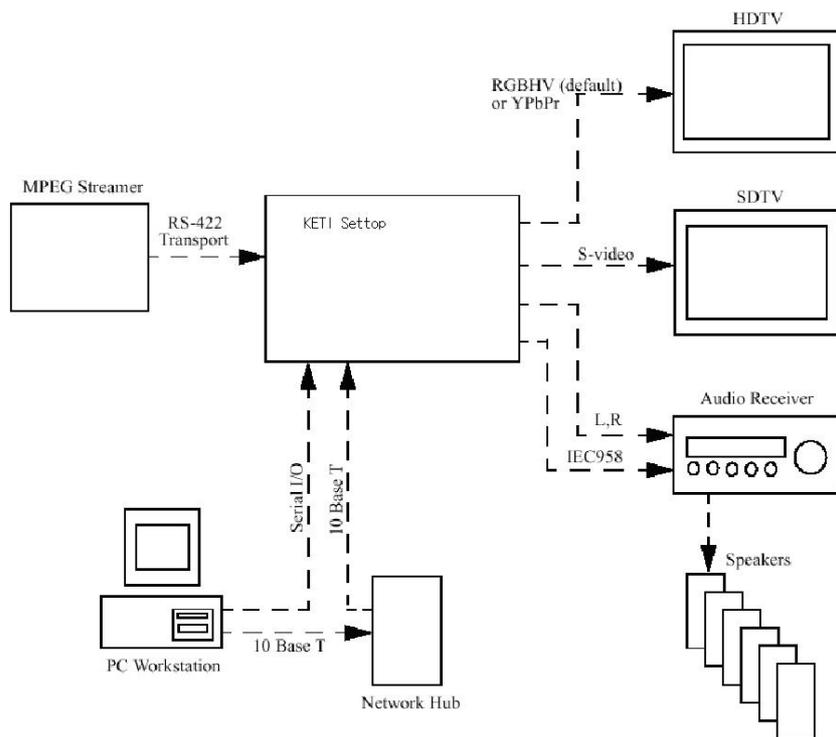


[그림 4-3] 셋톱박스의 소프트웨어 구성

소프트 웨어 환경은 [그림 4-3]에서 처럼 RTOS로는 WindRiver사의 VxWorks 5.4를 사용하며 MPEG2 시스템과 관련된 DTV API는 TeraLogic사의 Cougar DTV API를 사용하였다. Java 플랫폼으로는 JDK1.1.8을 기반으로 하는 자바 실행환경을 제공하는 WindRiver사의 Personal Jworks 3.1을 사용한다. 동작 대상이 되는 JavaTV 응용프로그램은 본 논문에서 구현되어지는 JavaTV API를 인터페이스로 사용하게 된다.

4.3 개발 환경

JavaTV API를 이용한 디지털 TV 구현은 우선, DVB규격을 만족하는 DTV API의 구현을 목표 셋톱박스를 대상으로 진행되어 진다. DTV API의 구현은 세부적으로 DVB-SI부분과 데이터 방송에 해당 되는 데이터 케로셀, 객체 케로셀의 구현이다. JavaTV API의 구현은 앞서 구현된 DTV API와 JNI(Java Native Interface)를 통해 JavaTV API 규격을 만족하도록 재구성하게 된다.

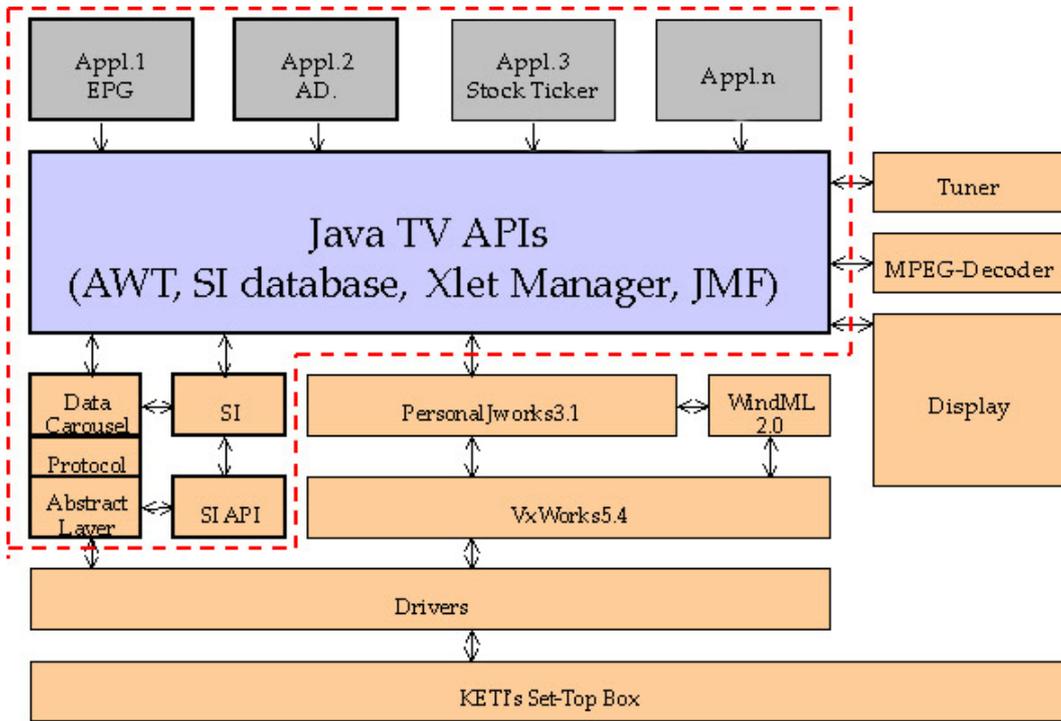


[그림 4-4] 개발 환경 구성도

5. 구현 내용

5.1 시스템 구조

[그림 5-1]은 구현된 디지털 TV 셋톱 박스의 전체 구성도를 보이고 있다.

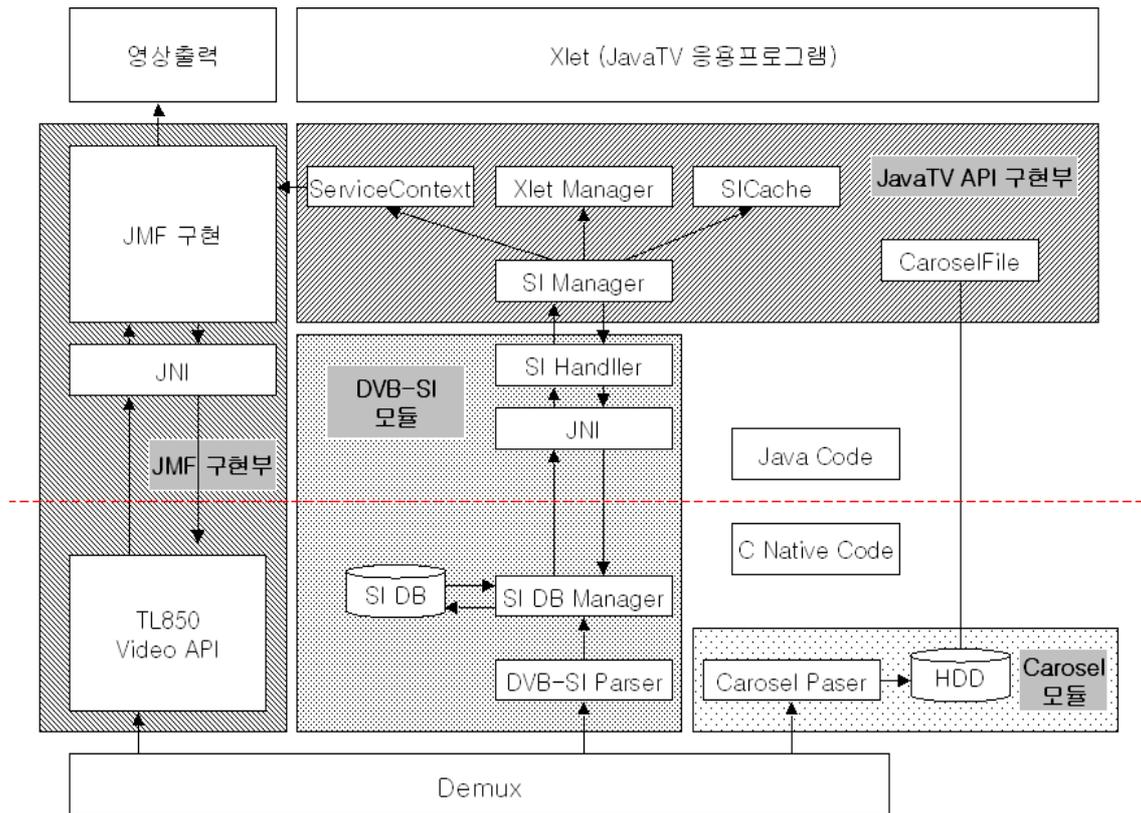


[그림 5-1] 구현된 디지털 TV 셋톱박스 전체 구성도

본 논문에서 구현하는 내용이 목표 셋톱박스에서 어떠한 부분을 차지하고 있는지를 보여주고 있는 [그림 5-1]에서 점선안에 해당하는 부분이 본 논문의 구현내용으로, JavaTV API 구현부, 영상출력을 위한 JMF 구현부, 방송 데이터 전송을 위한 데이터 및 객체 캐로셀 모듈, DVB SI모듈, 그리고 JavaTV API를 만족하는 디지털 TV 응용프로그램인 Xlet이다.

구현내용은 크게 DVB-SI 모듈, 데이터 및 객체 캐로셀 모듈, JavaTV API 구현부, JMF 구현부로 나뉘어 지며, [그림 5-2]는 각 구현부들의 구성을 보여주고 있다. [그림 5-2]의 중앙에 있는 점선을 기준으로 상단부는 Java 언어로 구현되었으며, 하단부는 C 언어로 작성되었다. 목표 셋톱박스의 직접적인 제어와 지원하는 라이브러리의 사용을 위해 DVB-SI, 캐러셀, JMF 구현부의 하단부는 모두 C 언어로 작성되었다. 그러나 각 구현부들이 C언어로 작성되었을 경우 특정 플랫폼에 종속적인 실행만 가능하기 때문에 본 논문에

서는 JavaTV API를 이용하여 각 구현부의 상단부는 Java 언어로 작성되었으며, 이로 인해 동작하는 셋톱박스의 플랫폼에 완전히 독립적으로 디지털 방송 응용프로그램은 작성되어 동작하는 것이 가능하게 된다. 즉 디지털 방송 응용프로그램 개발자는 동작되어질 셋톱박스의 플랫폼을 전혀 의식하지 않고 JavaTV API에 적합하게만 작성하면 정상적인 동작을 보증할 수 있다는 것이다.



[그림 5-2] DTV 셋톱박스 미들웨어 구현 구조

본 논문에서 구현한 디지털 박스 셋톱박스 미들웨어의 동작은 Java 언어로 작성된 응용프로그램을 통해 시작되며, 이 응용프로그램은 SIManager 객체를 생성하며, SIManager객체는 DVB-SI 구현부의 SIHandler객체를 생성하고, SIHandler객체는 객체 생성과 동시에 JNI(Java Native Interface)를 이용하여 C Native 코드로 작성된 구현부를 초기화 시킨다. C Native 코드는 역다중화기의 동작을 실행시키고 나머지 구현부들을 초기화 시켜 동작에 필요한 작업을 준비한다.

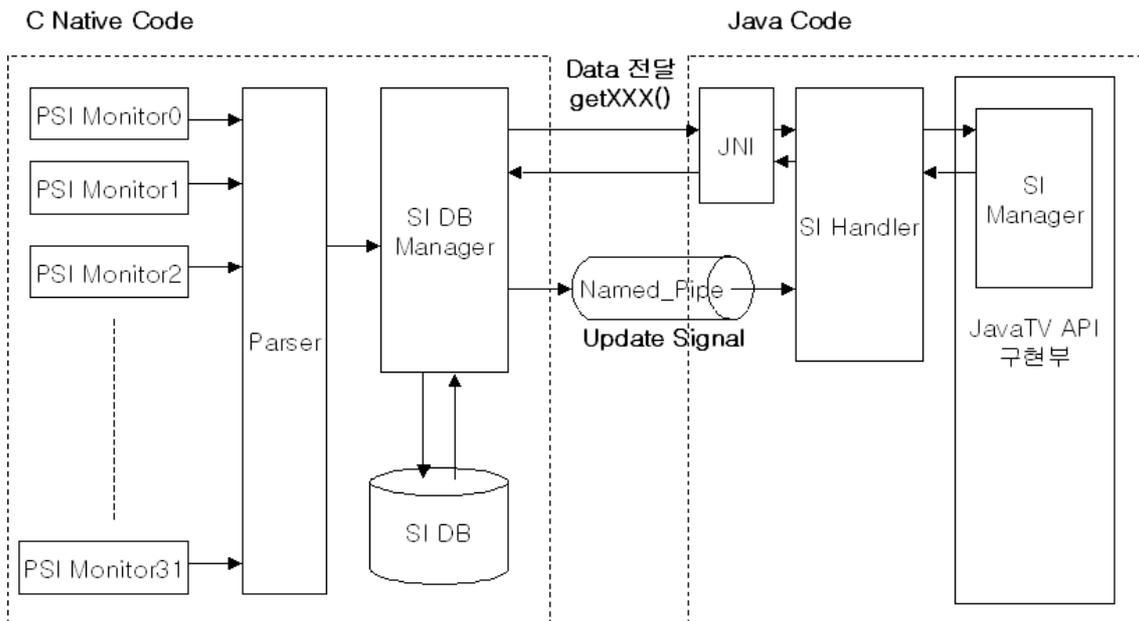
SIManager객체는 SIHanadller객체 생성이 끝나면, JMF구현부와 JavaTV API 구현부의 초기화 작업을 마친다.

본 논문에서 구현 목표로 하고 있는 셋톱박스는 역 다중화기를 TL850 칩셋 하드웨어로 지원하며, 동시에 32개까지 PSI 모니터링을 할 수 있는 기능을 지원하고 있다. 초기화가 시작되면, 동작에 필요한 PSI 및 DVB-SI, DSM-CC 데이터들을 관련 PID를 이용하여 쓰레드로 동작하면서 모니터링을 시작하게 된다. 각 모니터에 의해 발견된 데이터들은 해당 모듈로 전달되어 분석 및 필요한 작업을 진행하게 된다.

5.2 DVB SI 구현부

DVB-SI 모듈은 PID 모니터에 의해 발견된 PSI Data와 DVB SI Data를 관련 규약에 맞게 분석하여 그 정보를 SI Database에 저장하고 JavaTV API 구현부에 전달하는 역할을 한다. [그림 5-3]은 DVB SI 구현부의 구조를 나타내고 있다.

PSI 데이터 분석은 TeraLogic사에서 제공하는 DVT API를 이용하였으며, DVB-SI 데이터 분석은 관련규격에 맞도록 추가적으로 구현하였다. 구현된 SI 테이블 parser는 NIT, BAT, SDT, EIT, TOT, SIT이며, network_name_descriptor, service_listdescriptor, stuffing_descriptor 등 모두 47종류의 Descriptor parser가 구현되었다.



[그림 5-3] DVB SI 구현부 구조

PID에 의해 구분된 각 DVB-SI 데이터들은 구현된 각 Parser에 의해 분석되어진 후 SI DB Manager에 의해 SI DB에 저장된다. SI DB는 SI 데이터를 검색 및 관리가 용이하도록 저장하며, 여러 스레드가 동시에 접근할 경우 동시성 제어를 위해 세마포어(semaphore)를 이용한다.

SI DB에 데이터를 저장하고 관리하는 SI DB Manager는 새로운 SI 데이터가 분석되어 전달되면, 데이터를 SI DB에 저장하고 Named Pipe를 통해 Java 언어로 구현되어진 SI Handler에게 새로 갱신된 내용에 해당하는 신호를 보낸다. SI Handler는 갱신된 데이터의 종류를 확인하고 갱신된 데이터에 적합한 메시지를 호출하여 SI DB Manager에게 데이터의 전달을 요구하게 된다. 이때 Java 언어로 구현된 SI Handler는 C Native 언어로 구현된 SI DB Manager의 함수를 호출하기 위해 JNI(Java Native Interface)를 이용한다.

SI Handler는 SI DB Manager로부터 반환된 새로운 데이터를 JavaTV API에서 정의하고 있는 적절한 SI 객체로 생성하여 JavaTV API구현부의 SIManager객체에게 전달한다.

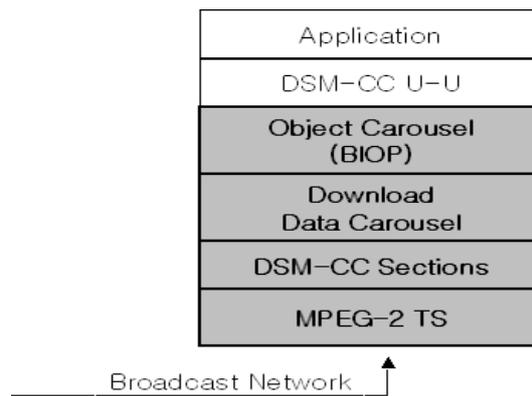
```
SIElement si=null;
switch(received_signal){
    case NIT:
        si= new Network(getNIT());
        break;
    case BAT:
        si = new Bouquet(getBAT());
        break;
    case SIT:
        ;
}
process(si)
```

[표 5-1] SIHandler의 SI 데이터 처리 알고리즘

SIManager 객체는 전달받은 SI 데이터 객체들을 필요로 하는 JavaTV 응용프로그램들에게 전달하며, JavaTV 응용프로그램이 추가적으로 필요로 하는 SI 데이터들은 SI DB로부터 전달 받은 방법과 동일한 방법을 이용하여 SI DB Manager에게 요청하여 제공받게 된다.

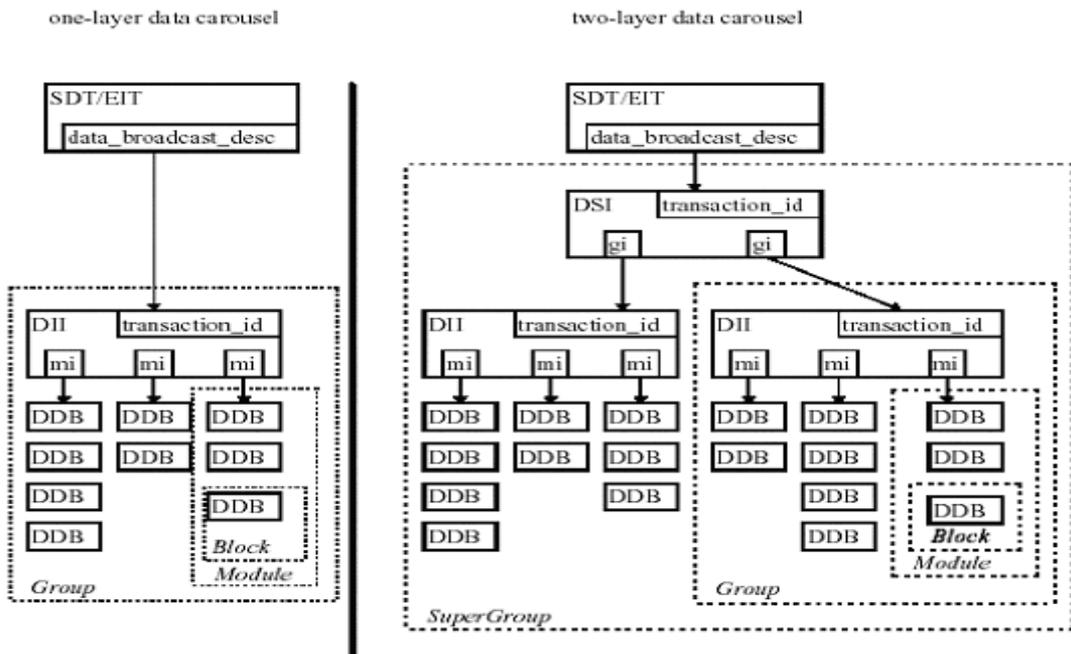
5.3 데이터 방송 구현부

방송환경에서 MPEG2-TS 내에 Audio/Video 소스이외에 부가적으로 데이터를 전송하기 위한 프로토콜로 Data piping, Data Streaming, Multiprotocol encapsulation, Data/Object Carousel의 4가지 규격으로 분류되어 지며, 본 논문에서는 Object Carousel을 구현하였다.



[그림 5-4] 객체 캐러셀 계층도

MPEG-2 TS로부터 PID에 의해 DSM-CC 데이터로 판별된 패킷들은 DSM-CC Section 규격에 맞게 분석되어 Data Carousel 분석단계로 전달된다.

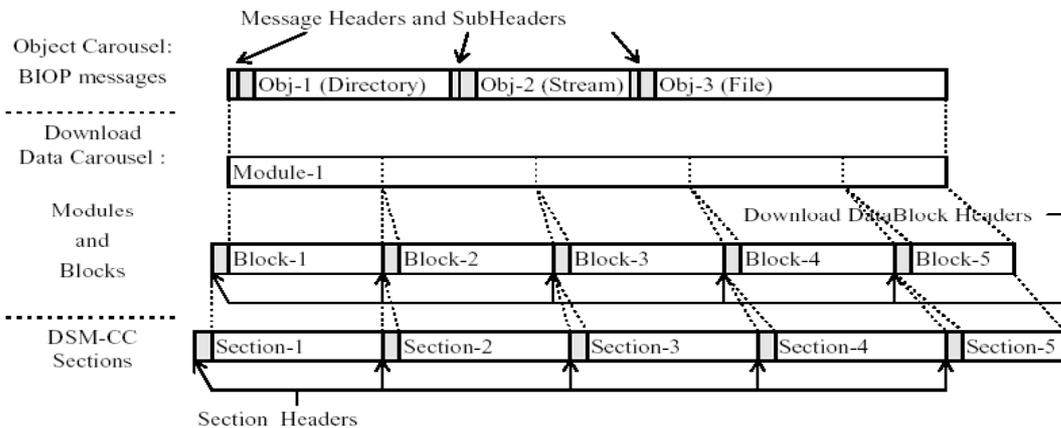


[그림 5-5] 데이터 캐러셀 구조

Data Carousel안에 전송될 데이터는 모듈들로 구성되며, 각 모듈은 데이터 블록으로 다시 나누어지며, 각 모듈내의 마지막 데이터 블록(같거나 작음)을 제외하고는 그룹내의 모든 블록의 크기는 같다.

Data Carousel 규격은 DSM-CC Download 규격의 데이터의 운반을 위해 DownloadDataBlock(DDB)메시지, 제어를 위한 Download InfoIndication(DII), DownloadServerInitiate(DSI), DownloadCancel 메시지를 이용한다. Data carousel 구조는 [그림 5-5]에 나타낸 것처럼 제어 정보에 따라 단일 계층 혹은 이중계층 구조로 구분되어 진다.

여러 Data Carousel중 특정 SI Data와 관련된 것을 식별하기 위해서 DVB SI의 SDT/EIT에 포함되는 data_broadcast_desc의 transaction ID값을 이용하여 단일계층의 경우 DII의 transaction ID값과 일치하는 것을, 이중계층의 경우 DSI의 transaction ID값과 일치하는 것을 찾게 된다. DII는 DDB들로 구성된 모듈을 그룹짓는 정보를 갖으며, DSI는 DII를 그룹 짓는 슈퍼그룹 정보를 갖게 된다. DII 메시지는 ModuleInfoByte(mi)필드를 사용하여 Carousel내의 모듈에 대한 정보를 기술하며, DSI메시지는 GroupInfoByte(gi)필드를 사용하여 Carousel내의 그룹에 대한 정보를 기술한다.



[그림 5-6] 객체 캐러셀 분석과정

각 블록의 헤더정보를 분석하여 하나의 모듈객체를 완성할 수 있게 되고, [그림 5-6]에서 보여지는 것과 같이 하나의 모듈에는 여러 개의 객체 (Directory, Stream, File)로 구성될 수 있다. 결국 모듈의 BIOP 메시지를 분석하여 각각의 Object를 얻어 낼 수 있다.

```

if(table_id == DSMCC_message_header){
    Parse DSMCC_message_header
    If(message_id == DSI){
        Parse DSI_header
        Parse BIOP::Srg(Service_Gateway)
    }
    if(message_id == DII)
        parse DII_header
    if(message_id == DDB){
        parse DDB_header
        if(BIOP::Dir) parse BIOP::Dir //Directory
        if(BIOP::Fil) parse BIOP::Fil //File
    }
}
}

```

[표 5-2] 객체 캐러셀 분석 알고리즘

본 논문에의 구현은 얻어진 각각의 Object들을 디렉토리 구조로 하드디스크에 저장하고, 그 정보를 JavaTV API 구현부에 전달하여 JavaTV API에서 정의하고 있는 javax.tv.carousel.CarouselFile을 이용하여 응용프로그램이 접근할 수 있도록 하고 있다.

5.4 JavaTV API 구현부

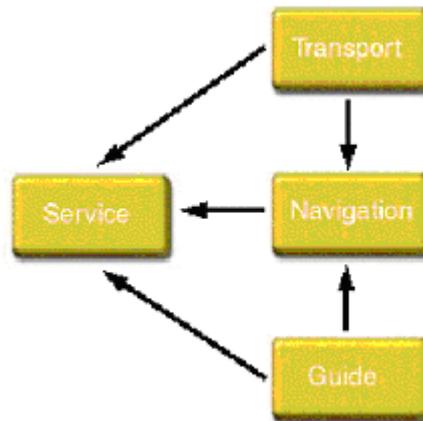
JavaTV API는 javax.tv 패키지로 구성되어 있으며, Java 언어를 이용하여 작성한 응용프로그램이 디지털 TV 수신기에서 동작하는데 필요한 인터페이스를 정의하고 있다. JavaTV API 구현부는 JavaTV API에서 정의하고 있는 인터페이스의 내부 구현으로 C Native 코드로 작성된 DVB SI모듈과 방송 데이터 모듈 등을 이용하여 JavaTV API로 정의된 인터페이스에 맞게 동작하게 하는 역할을 맡는다. JavaTV API 구현의 주요구성은 서비스 및 서비스 정보, 서비스 선택, JMF, 응용프로그램 관리자로 크게 나눌 수 있다.

5.4.1 서비스 및 서비스 정보

DVB SI모듈에 의해 전달되는 서비스 정보를 표현하고 응용프로그램이 서비스와 관련된 동작을 할 수 있게 하는 역할을 하는 부분으로 JavaTV API의

javax.tv.service, javax.tv.service.navigation, javax.tv.service.guide, javax.tv.service.transport 패키지로 구성되어 있다.

각 패키지들은 DVB SI 테이블로 정의 되어진 SI 정보를 표현하는 여러 클래스들과 SI Database를 접근하고 SI 데이터를 관리하는 클래스들로 구성된다.



[그림 5-7] SI 패키지들간의 의존성

javax.tv.service 패키지는 SIManager, Service, SIElement 등과 같은 SI와 관련하여 다른 패키지들이 필요로 하는 공통적인 클래스들로 구성되어있다. SI 정보를 관리하고 제공하는 역할을 하도록 설계된 SIManager 인터페이스는 SIManagerImpl 클래스로 구현되었으며, SIManagerImpl 클래스는 SIHandler클래스가 C Native로 구성된 SI 모듈과 통신하여 얻어온 SI 데이터를 각각의 SI 정보형식에 알맞은 객체로 생성하여 이를 필요로 하는 응용프로그램에게 전달하는 역할을 한다. 결국, SIManagerImpl는 응용프로그램에게 있어서 SI Database를 접근할 수 있는 기본적인 포인트 역할을 한다.

javax.tv.service.nevigation 패키지는 ServiceList, ServiceComponent, SIElementFilter 등과 같은 서비스들을 탐색하는데 사용되는 클래스들로 구성되어 있다.

javax.tv.service.guide 패키지는 ProgramEvent, ProgramSchedule 등과 같은 프로그램의 스케줄, 프로그램 이벤트 등을 표현하여 EPGs를 위해 사용되는 클래스들로 구성되어 있다.

javax.tv.service.transport 패키지는 Transport, TransportStream, Network, Bouquet 등과 같은 MPEG-2 TS의 전송 메카니즘과 관련된 정보를 표현하는 클래스들로 구성되어 있다.

SI 데이터들을 표현하는 각 클래스들은 SIManagerImpl에 의해 관리되며, SIManagerImpl 객체는 SI 모듈로부터 전달 받은 객체들 중 동작과 관련하여 중요한 몇몇 객체들을 SICache에 저장하여 응용프로그램으로부터의 SI 데이터 요청을 동기적으로 처리한다.

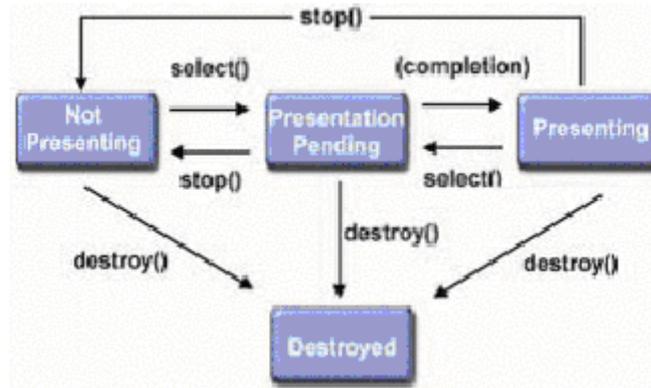
SICache에 저장되어 있지 않은 SI 데이터에 대한 요청은 비동기적으로 이루어 지는데, 요청된 데이터가 저장되어 있지 않은 경우 SI 모듈에 의해 MPEG-2 TS에 접근하는 데에 있어 비교적 긴 시간이 요구되기 때문이다. 동기적인 접근은 일반적으로 Java에서 사용하는 이벤트 통지 메커니즘을 사용하며, 비동기적인 접근은 SIRequestor, SIRequest 인터페이스를 이용하는데, SIRequestor를 등록하여 필요로 하는 정보가 사용 가능한 경우 Callback을 수신하여 요청한 데이터 또는 실패를 수신할 수 있으며, SIRequest는 비동기 요청의 취소를 위해 사용된다.

5.4.2 서비스 선택

응용프로그램이 서비스 정보 패키지를 통해 알려진 여러 서비스 중 특정 서비스를 선택하여 표현하게 하는 기능을 제공하는 부분으로 JavaTV API에는 javax.tv.service.selection 패키지로 구성되어 있다.

서비스가 표현되고 있는 환경을 표현하는 클래스는 ServiceContext이며, 특정 서비스와 연관된 컴포넌트의 표현 제어를 응용프로그램이 하게 제공한다. ServiceContext는 Presenting, NotPresenting, Presentation Pending, Destroyed 네 가지 상태중에 하나에 존재 할 수 있다. [그림 5-8]은 ServiceContext의 상태변이를 보여 주고 있다.

서비스의 표현은 ServiceContext의 select()메서드의 호출로 시작되며, 만약 select()메서드의 호출이 성공적으로 끝났다면, NormalContentEvent 혹은 AlternativeContentEvent가 생성되고, ServiceContext는 Presenting 상태로 이동하며, 만약 서비스 선택이 실패하면, SelectionFailedEvent가 생성된다. 만약 select() 호출의 이전상태가 Not Presenting이면, ServiceContext는 그 상태로 돌아가게 되고, PresentationTerminatedEvent가 생성된다. select() 호출의 이전상태가 Presenting이면, ServiceContext는 NormalContentEvent 또는 AlternativeContentEvent의 결과로 이전상태로 돌아가려고 시도한다. 그것이 불가능하다면, PresentationTerminatedEvent를 리턴한다.



[그림 5-8] ServiceContext의 상태 전이

PresentationTerminatedEvent이 보고되면 서비스 표현의 종단을 의미하며 이것은 Not Presenting 상태로의 전이의 원인이 된다. Service Presentation의 종단은 stop()메서드의 호출에 의하거나, 동작환경에서 더 이상 진행 할 수 없는 변화의 발생에 의해 이루어진다.

Destroyed상태는 destroy()메서드의 호출에 의해 전이되며, 한번 이 상태에 들어오게 되면 ServiceContext는 더 이상 어떤 목적으로도 사용될 수 없으며, destroyed된 ServiceContext는 Garbage Collection의 후보가 된다.

서비스가 표현될 때 getServiceContethandler()메서드가 호출되어 ServicContentHandler객체를 얻을 수 있는데, 이것은 서비스 컴포넌트를 표현하는 플레이어의 참조값을 리턴하며, 결국 JMF에서 제공하는 Player객체를 참조하게 된다.

5.4.3 JMF(Java Media Framework)

JavaTV API는 방송 Media 관리를 위해 JMF(Java Media Framework)1.0 API를 이용한다. JMF API는 전송메카니즘, 전송 프로토콜, Media Content 형식에 독립적인 미디어 디스플레이를 위한 프레임웍의 정의로 JavaTV API 미디어 출력의 기반을 제공한다.

JMF API는 javax.media 패키지로 구성되며, JavaTV API의 구현에서는 JMF API에서 JavaTV API 밀접한 관련을 갖는 부분만을 구현하여 사용하며, JavaTV API는 미디어 재생과 관련하여 디지털 TV에 필요한 몇 가지 클래스들을 javax.tv.media패키지로 구성하여 JMF API에서 지원하지 않는 클래스들을 추가 하고 있다.

본 논문에서의 구현은 WinNT 기반으로 썬 마이크로 시스템즈에서 제공하는 JavaTV RI(Reference Implementation)에 포함되어 있는 JMF lite 구현을 일부 수정하여 목표 셋톱박스에서 동작되도록 재 구현하였다.

JavaTV RI에 포함되어 있는 JMF lite는 Windows에서 동작하기 위해 DirectX를 사용하였으나, 본 논문에서의 구현은 목표 셋톱박스에서 사용하고 있는 Cougar DTV API를 이용하여 재 구현하였다. [그림 5-9]은 JMF 구현부를 표현하고 있으며, 어두운 부분이 본 논문에서 재 구현한 부분으로 수정된 Windows 구현과 대칭적으로 보여주고 있다.

JMF API(javax.media)	
JavaTV Implementation (com.sun.tv.media)	
JMF LITE Win32 (java) (win32/com.sun.media)	JMF LITE Cougar (java) (khu.tv.media)
Jmam.dll (C native)	XXX.so (C native)
DirectX Interface	Cougar Interface

[그림 5-9] JMF Lite 구현

5.4.4 응용프로그램 관리자

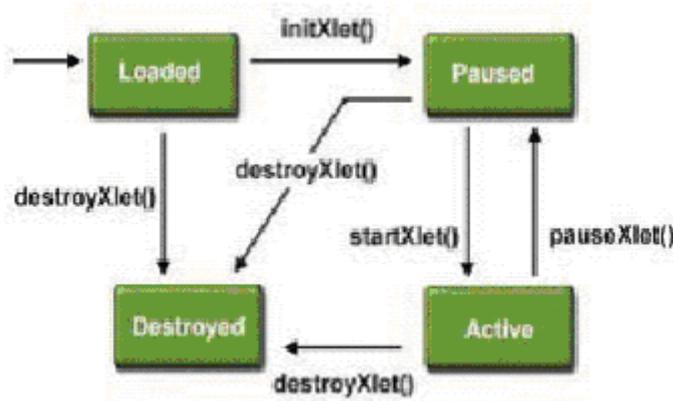
JavaTV API는 응용프로그램과 응용프로그램이 동작할 환경과 관련하여 생명주기 모델을 정의하고 있으며, javax.tv.xlet 패키지로 구성된다.

JavaTV API는 이와 같은 환경에서 동작하는 응용프로그램을 Xlet이라는 이름으로 정의하고 있으며, Xlet은 JavaTV API에서 정의한 생명주기 모델을 반드시 따라야 한다. 응용프로그램 관리자는 이와 같은 Xlet의 생명주기를 제어하는 역할을 한다.

Xlet 생명주기는 Loaded, Paused, Active, Destroyed와 같은 네 가지 상태를 가지며, 각 상태의 변이로 생명주기 모델이 이루어진다. [그림 5-10]은 Xlet 생명주기 모델을 보여 주고 있다.

Loded 상태는 Xlet의 인스턴스가 생성된 직후의 상태로서 응용프로그램 관리자는 아규먼트 없는 기본 생성자의 호출로 Xlet을 생성하고, Exception이 발생하면 Xlet을 Destroyed상태로 전이 시킨다.

Paused상태는 Xlet이 생성되어 초기화되어지고 아직 시작되지 않은 상태이다. 응용프로그램 관리자는 Xlet.initXlet()메서드를 이용하여 Xlet을 초기화 하고 이 메서드에서 Exception의 발생 없이 반환되면 이 상태가 Paused 상태가 될 수 있다. 또는 동작 중인 Xlet의 pauseXlet() 메서드를 호출하여 일시정지 하는 경우도 Paused 상태가 될 수 있다.



[그림 5-10] Xlet 생명주기 모델

Active 상태는 Xlet이 일반적인 기능을 수행 중인 상태로 Paused 상태에서 startXlet() 메서드가 호출되어 정상적으로 반환된 후 이 상태로 전이하게 된다. Destroyed 상태는 destoryXlet() 메서드가 성공적으로 반환되었을 때이며, Xlet은 할당된 모든 자원을 반납하고 종결된다. 종결된 Xlet은 Garbage Collection의 후보가 된다.

모든 Xlet은 자신의 XletContext객체를 하나씩 갖는데, Xlet과 응용프로그램 관리자는 XletContext를 이용하여 통신하고, Xlet의 속성을 접근하는데 사용한다. XletContext는 Xlet이 디스플레이 될 때 필요한 컨테이너 등 Xlet이 동작하는 환경을 정의 한다.

5.5 실험 결과 및 고찰

5.5.1 실험 결과

[그림 5-11]는 구현된 디지털 TV 셋톱박스를 동작시킨 기본 화면이다. 실험에서 사용된 TS는 실험을 위해 별도 제작한 후 셋톱박스의 하드디스크에 저장하여 실험을 실시 하였다.



[그림 5-11] JavaTV 기본 화면

셋톱박스가 동작을 시작하면, MPEG-2 영상과 음성으로 구성된 주 화면이 나타나게 되고, 무선 키보드를 이용하여 메뉴키를 누르면 [그림 5-12]과 같이 현재 동작 중인 서비스에서 실행될 수 있는 응용프로그램의 메뉴를 보여 준다.



[그림 5-12] 응용프로그램 메뉴

시청자는 서비스 가능한 메뉴 중에서 관심있는 메뉴를 선택하여 응용프로그램을 동작 시킬 수 있다. [그림 5-13]에서와 같이 선수 명단을 선택하여 경기 중인 선수들의 명단을 볼 수 있다.



[그림 5-13] 선수 명단 응용프로그램

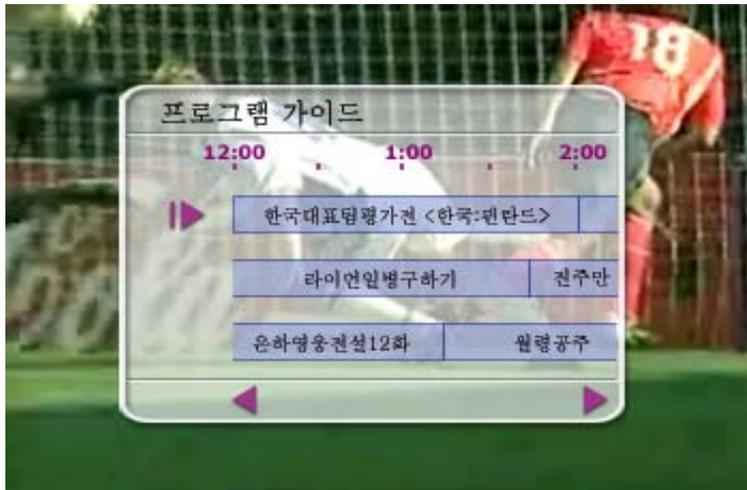
선수 명단 응용프로그램은 보여지는 선수 명단 중 원하는 선수의 이름을 선택하면 선택된 선수의 신상명세를 제공하고 있다. [그림 5-14]은 선택된 선수의 신상명세를 보여 주고 있다.



[그림 5-14] 신상명세 제공 응용프로그램

현재 방송 중이거나 앞으로 방송될 프로그램들에 대한 정보를 알려주는 프로그램 안내 또한 메뉴에서 선택하여 동작 시킬 수 있으며, [그림 5-15]에서 EPG 실행 예시를 보여 주고 있다. 현재 선택되어져 있는 프로그램은 해당 프로그램의 앞에 화살표로 선택되어져 있음을 표시하고, 다른 프로그램

을 선택하면, 채널의 변경도 가능하다.



[그림 5-15] 프로그램 가이드 응용프로그램

화면에 나타나지 않은 앞으로 방송될 프로그램의 표시는 화면 하단 좌우에 있는 화살표를 선택하여 현재 또는 이후에 방송될 프로그램의 안내를 볼 수 있다.

선수 신상명세를 제공하는 응용프로그램은 Xlet으로 작성되었으며, 현재 동작될 수 있는 Xlet의 명세는 SI 데이터에 의해 알 수 있으며, 선택되어 실행되는 Xlet의 코드와 Xlet이 사용하는 선수명세 자료는 객체 카로셀을 이용해 셋톱박스에 전달된다.

프로그램 가이드를 제공하는 응용프로그램 또한 Xlet으로 작성되었으며, 화면에 나타날 프로그램 스케줄 정보는 DVB-SI의 EIT에 의해 전달되어 진다.

5.5.2 결과 고찰

본 논문에서 구현한 디지털 TV 셋톱박스의 동작결과는 실험결과에서 알아본 바와 같다. 직관적으로 볼 때 본 논문에서의 구현은 JavaTV API를 이용한 미들웨어구조를 갖지 않는 셋톱박스에 비해 동작에 있어 한 계층을 더 거쳐야 하므로 성능 저하의 원인이 될 수 있다. 그러나 텔레비전의 특성 상 시청자가 체감하지 못하는 성능의 차이는 전혀 고려의 대상이 될 필요성이 없으며, 시청자에게 다양성을 제공하는 것이 오히려 텔레비전 본래의 역할에 충실하다 볼 수 있다.

JavaTV API를 이용한 디지털 TV 셋톱박스 미들웨어는 응용프로그램에게 동작하는 플랫폼에 제한을 두지 않을 수 있게 하며, 구현에 따라 기반으로 하는 방송기술(프로토콜)의 차이에도 제한을 두지 않게 할 수 있다.

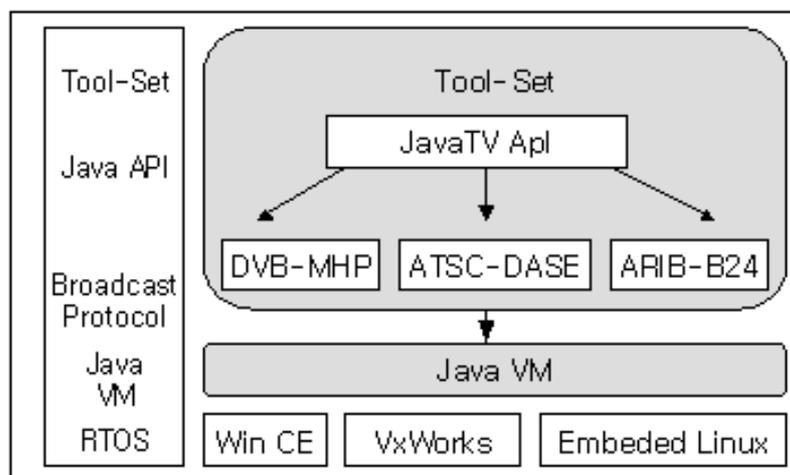
결국, 응용프로그램 개발자는 JavaTV API에만 충실하게 응용프로그램을 개발하면 동작은 미들웨어가 보증하며, 이것은 응용프로그램 개발자로 하여금 응용프로그램에서 필요한 부분만을 전념하게 할 수 있게 한다

6. 결론 및 향후 과제

본 논문에서는 JavaTV API를 이용하여 디지털 TV 셋톱박스에 미들웨어를 구현하였다. 다양한 플랫폼과 방송규격을 갖는 디지털 TV 셋톱박스는 만족하고자 하는 방송규격을 특정 플랫폼에 종속적으로 구현하고, 이와 같은 디지털 TV 셋톱박스는 특정 플랫폼에 종속적인 디지털 TV 용 응용프로그램만 동작될 수 있다. 그러나 본 논문에서는 플랫폼 독립성을 보장하는 Java를 이용하는 JavaTV API를 디지털 TV 셋톱박스에 구현함으로써, 디지털 TV 응용프로그램이 여러 상이한 플랫폼에서 독립적으로 동작될 수 있는 미들웨어를 제공하였다.

JavaTV API는 플랫폼의 독립성 뿐만 아니라, 방송규격에 있어서도 디지털 TV 응용프로그램이 방송규격에 종속적인 코드를 갖지 않게 추상화 하여, 디지털 TV 셋톱박스 미들웨어의 의미를 지닐 수 있다.

그러나, JavaTV API는 특정 플랫폼과 방송규격에 만족하는 실행환경을 제공하기 위해서 구현 및 포팅에 전문적인 지식이 요구되고, 구현기간에 많은 시간과 노력이 소요된다는 단점이 있다. 따라서 일반적으로 흔히 사용되는 실시간 운영체제와 방송규격에 맞는 JavaTV API의 포팅을 위한 툴셋이나 프레임 워크의 필요성이 요구된다.



[그림 6-1] JavaTV 포팅 툴셋의 개념도

본 논문에서는 실시간 운영체제로 VxWorks를, 방송규격으로 DVB규격을 만족하는 미들웨어를 구현하였으나, [그림 6-1]과 같이 WinCE, Embedded Linux 등의 흔히 선호되는 실시간 운영체제와 ATSC, ARIB 등과 같은 유력한 방송

규격에 맞는 JavaTV API의 구현이 컴포넌트화 되어 사전에 구현되어져 있는 틀셋 및 프레임 워크를 가진다면, 디지털 TV 셋톱박스 개발자는 JavaTV API 포팅 틀셋을 이용하여 필요에 따라 원하는 플랫폼과 방송규격에 맞게 손쉽게 개발할 수 있을 것이다.

하지만 단순히 몇몇 목표로 하는 플랫폼과 방송규격에 대한 사전 구현이 그 해결이 될 수 있는 것만은 아니며, 디지털 TV에 필요한 각 모듈이 완벽히 컴포넌트화 되어 저야 하며, 각 컴포넌트들 간의 인터페이스는 넓은 시야를 가지고 설계되어야 할 것이다.

참고 문헌

- [1] ISO/IEC 13818-1 : “Information technology - Generic coding of moving pictures and associated audio information - Part 1: Systems”
- [2] ISO/IEC 13818-6: “Information technology - Generic coding of moving pictures and associated audio information - Part 6: Extensions for DSM-CC”
- [2] ITU-T Recommendation H.222.0 : “Generic coding of moving pictures and associated information : System”
- [3] ETSI TS 102 812 : “Digital Video Broadcasting(DVB); Multimedia Home Platform(MHP) Specification 1.1”
- [4] ETSI EN 300 468 : “Digital Video Broadcasting(DVB); Specification for Service Information(SI) in DVB Systems”
- [5] ETSI TR 101 211 : “Digital Video Broadcasting(DVB); Guidelines on implementation and usage Of Service Information(SI)”
- [6] ETSI EN 301 192 : “Digital Video Broadcasting(DVB); DVB specification for data broadcasting”
- [7] ETSI TR 101 202 : “Digital Video Broadcasting(DVB); Implementation guidelines for Data Broadcasting”
- [8] JavaTV API Technical Overview v1.0 - Technical White Paper : Sun Microsystems. 2000
- [9] JavaTV API Reference Implementation Porting Guide : Sun Microsystems. 2000
- [10] Java Technologies for Interactive Television - Technical White Paper : Sun Microsystems. 2000
- [11] JavaTV, <http://java.sun.com/products/javatv/>
- [12] Personal Jworks,
http://www.windriver.com/products/html/persjwks31_ds.html
- [13] OpenTV, <http://www.opentv.com>
- [14] MicrosfotTV, <http://www.microsoft.com/tv/default.asp>
- [15] DASE, <http://www.dase.org>
- [16] ATVEF, <http://www.atvef.com>
- [17] MHEG, <http://www.mheg.org>

- [18] 전자부품 연구원 HDTV 사업단, 서울대학교 뉴미디어 통신공동 연구소.
디지털 TV 용어 사전. 영한출판사
- [19] 유시룡, 장규환, 이병욱, 김종일, 정해묵. MPEG 시스템. 대영사

Abstract

Implementing Digital TV Set-top Box Middleware Using JavaTV API

Sewoo Lee

Department of Electronic Engineering
Graduate School of Industry and Information Science
Of Kyunghee University

Digital TV not only provides high-definition and multi-channels, but also provides application programs. The TV viewer can be supplied one way video, audio and further more various type of program specific applications, stand alone application, program guide application, advertisement application and so forth.

This kind of additional service can be provided by broadcast network including video and audio which can be downloaded within set-top-box. Depending on the application program and digital-TV set-top-box, it also can provide interactive service using return channels.

But, applications that transported by MPEG-2 TS is not suitable for all digital TV set-top-box for there exists various products and platforms. And, there being so many broadcasting protocols, it is supposed to receive same protocol that provider uses.

To solve those problems, in this paper, JavaTV API which provides broadcast protocol abstraction for application program and guaranteeing platform independency, is used to implement digital TV middle-ware.

In this paper, DVB standard, as known as Europe's form, is used to implement digital TV set-top-box middle ware. Latterly, ATSC standard and ARIB standard can also be used to implemented. Toolsets and frameworks which supports implement easily on there own standards and platform, can be developed.

Keywords : MPEG2-TS, Digital TV Set-top-box middleware, JavaTV API, DVB

감사의 글

그다지 길지 않은 연구실 생활을 돌이켜 보면 좀 더 성실하고 부지런하게 연구실 생활을 했으면 하는 아쉬움이 남습니다. 미약하나마 한 편의 논문을 완성할 수 있도록 많은 분들의 도움이 있었기에, 여기에 그 분들께 감사의 마음을 전하고자 합니다.

부족한 저에게 많은 지도와 가르침을 주신 이승룡 교수님께 깊은 감사 드립니다. 그리고 연구실 생활에 많은 도움을 주신 김형일님과 무에서 유를 창조하기 위해 추운 겨울부터 무더운 여름까지 밤낮없이 연구실에서 함께한 곽병돈, 조인수, 전용주, 김현경, 이석진, 조두원에게 감사 드리며, 이들의 앞날에 행복이 함께 하길 바랍니다.

또한, 길지 않은 시간동안이나마 많은 도움을 아끼지 않은 실시간 & 멀티미디어 연구실 선, 후배님들께 감사 드립니다.

아울러, 저의 부족함에도 항상 격려를 아끼지 않으신 전자부품연구원의 고속네트웍연구센터 연구원님들 모두에게 감사드립니다.

끝으로, 못난 자식이 미약 하나마 석사과정을 마칠 수 있도록 낱아 길러 주신 부모님께 고마운 마음을 전할 길이 없습니다. 항상 제 이야기에 귀 기울여준 사랑하는 누나와 형수님, 귀여운 내 조카들, 그리고 저에게 항상 등불이 되어주고, 제 인생에 가장 큰 스승이 되어준 나의 사랑하는 민우형에게 감사의 눈물을 흘립니다.

항상 저를 위해 많은 도움을 주신 여러분을 생각하며, 이제 작은 마침표를 찍고 더 나은 미래를 위해 노력하고 정진하겠습니다.

모두 항상 건강하시고 행복하시길 바랍니다.

2002년 여름 이 세 우