디지털방송으로 전송되는 애플리케이션을 위한 PC 시뮬레이터의 구현

류일권[1],정문열[2]
서강대학교 영상대학원 미디어 공학[12]
ginatryu@hanmail.net[1],moon@mail.sogang.ac.kr

A PC Simulator for Applications carried in Digital Broadcasting

Yll Kwon Ryu[1], Moon Ryul Jung[2]
Media Technology, Graduate School of Media Communication, Sogang University[1]

요약

디지털 방송은 비디오뿐 아니라, 각종 데이터와 Xlet 이라고 부른 실행 프로그램 (Java 애플리케이션)을 TV 수신기로 전송할 수 있다. 본 논문은 이 애플리케이션을 방송 전송 스트림 (TS)으로부터 직접 읽어 이를 실행하는 PC 시뮬레이터의 구현방법을 기술한다. 기존에 나와 있는 Xlet 시뮬레이터들은Xlet 코드를 전송스트림에서 읽는 것이 아니라 로컬 파일 시스템에서 읽는 방식을 사용한다. 따라서 실제 방송 상황과 많이 다르고 특히 전송스트림을 통해서 전송되는 시간의 영향을 받는 스트림 이벤트 같은 것을 처리하기 힘들다. 뿐만 아니라, 본 시뮬레이터를 이용하면 PC에서도 인터랙티브 방송이 포함되어 있는 디지털 방송을 시청할 수 있게 해 준다. 본 논문에서 구현하는 에뮬레이터는 방송 스트림인 TS를 실시간으 읽으면서 Xlet을 실행하기 때문에 기존 에뮬레이터의 한계를 극복한다. 현대인은 PC를 이용하여 많은 작업을 하므로 PC에서 인터랙티브 방송을 이용할 수 있게 하는 것은 방송과 통신의 융합이라는 21세기 미디어 개념에 잘 부합된다.

1.서론

디지털 방송매체의 핵심 서비스라고도 불리 우는 데이터 방송은 영상과 음성 이외에 텍스트, 그래픽, 문서, 소프트웨어 등의 멀티미디어 데이터를 방송매체를 이용하여 전송하고, 전용 셋탑 박스를 통하여 시청자가 그 정보를 이용하게 하는 서비스이다. 이는 방송과 통신의 융합이라는 시대적 흐름에 가장 잘 부합되는 서비스로서, 방송에 커뮤니케이션 속성을 부여하여 기존의 일방향적인 방송 서비스의 한계를 벗어나 시청자와 메시지를 주고 받는 양방향성을 제공한다. 데이터 방송에서 자바로 개발된 Xlet이 이와 같은 양방향성을 제공한다.

본 논문은 이 Xlet을 방송 전송 스트림 (TS)으로부터 직접 읽어 이를 실행하는 PC 시뮬레이터의 구현방법을 기술한다.

현재 Xlet을 PC에서 구동시키는 시뮬레이터들은 Transport Stream을 처리하는 것이 아니라 PC에 자바 클래스 파일 형태로 존재하는 Xlet을 구동시키고 있다. 이처럼 Transport Stream을 처리하지 않은 시뮬레이터는 PC에서 Xlet을 구동시키는데 있어서 실제 방송 상황과 많이 다르고 특히 전송스트림을 통해서 전송되는 시간의 영향을 받는 스트림 이벤트 같은 것을 처리하지 못하는 한계가 있다.

본 논문에서 구현하는 시뮬레이터는 방송 스트림인

Transport Stream을 처리하여 Xlet을 구동시킴으로써 기존 시뮬레이터의 한계를 극복한다. PC에서 실제 방송 스트림을 처리하여 Xlet을 구동시키기 때문에 궁극적으로 본 논문의 시뮬레이터는 PC를 통해서도 일반 사용자가 데이터 방송을 이용할 수 있는 시스템을 구축해준다. 현대인은 PC를 이용하여 많은 작업을 하므로 PC에서 데이터 방송을 이용할 수 있게 하는 것은 방송과 통신의 융합이라는 21세기 미디어 개념에 잘 부합된다.

국내 디지털 위성 방송의 데이터 방송은 DVB-MHP 방식으로 서비스하고 있다. DVB-MHP 규약에서 Xlet은 Object Carousel 프로토콜로 캡슐화되어 Transport Stream에 담겨 전송된다. 본 논문에서 구현하는 시뮬레이터는 Transport Stream에 Object Carousel 방식으로 인코딩되어 비트 스트림으로 존재하는 Xlet을 Transport Stream으로부터 읽어 들여 PC에서 Xlet을 구동시킨다. 이를 위해 본 시뮬레이터는 Demultiplexer, Object Carousel Parser, 애프리케이션 매니저, 미들웨어 시스템으로 구성된다.

2.본론

2.1데이터방송전송프로토콜

디지털 방송에서 비디오/오디오 스트림과 데이터 스트림은 MPEG-2 System 규약에서 정의하고 있는 MPEG-2 Transport Stream에 다중화되어 전송된다. MPEG-2 Transport Stream은 188 바이트의 크기를 가진Transport Stream Packet이 연속적으로 모여 이루어진 비트 스트림이다.[1][14] Transport Stream Packet은 4 바이트의 Packet 헤더와 184 바이트의 Packet 페이로드 부분으로 구성된다.[1] Packet 페이로드에 비디오/오디오 또는 데이터가 해당 스트림 타입에 맞는 방법에 따라 인코딩되어 채워진다.

데이터는 MPEG-2 System에서 정의하고 있는Section 구조에 따라 캡슐화되어 하나 이상의 Packet 페이로드에 나누어져 담긴다.[1][14] MPEG-2 Section은 데이터를 Packet 페이로드에 담아 전송하기 위해 정의된 형식으로써 가장 간단한 구조로 데이터를 Transport Stream에 담을 수 있는 방법이다. MPEG-2 Section은 MPEG-2 PSI와 DVB SI 등의 Table이나 사용자가 정의한 임의의 데이터를 담을 수 있다.

국내 위성 디지털 데이터 방송 규약인 DVB-MHP에서는 애플리케이션의 전송을 위해 DVB Object Carouel 프로토콜을 사용한다. Object Carousel 프로토콜도 MPEG-2 Section으로 캡슐화되어 Transport Stream Packet으로 전송된다. DVB Object Carouel은 DSMCC User to User Data Carousel을 기본으로 해서 정의되었다. 다음 그림(그림2.1)은 애플리케이션의 전송을 위해 사용하는 프로토콜의 관계를 나타내고 있다.

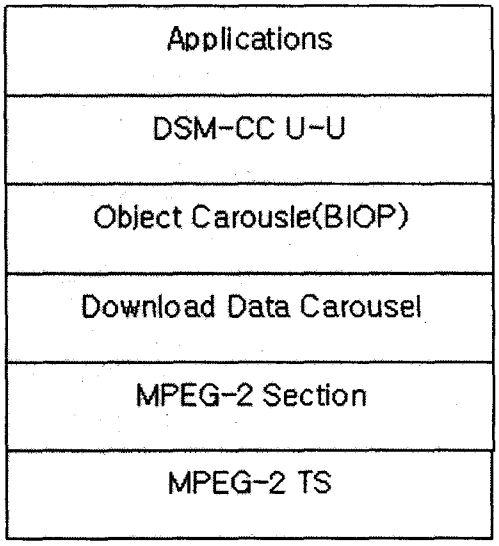

그림 2.1 애플리케이션 전송 프로토콜

애플리케이션을 구성하는 파일과 자원들은 Object Carousel 프로토콜에 따라 BIOP 메시지로 인코딩된다. 각 BIOP 메시지들은 Data Carousel 프로토콜에 따라 모듈을 형성하게 되고 모듈은 하나 이상의 DownloadDataBlock 메시지로 다시 인코딩된다.[2][11] Data Carousel 프로토콜로 인코딩된 후에 MPEG-2 Section 구조로 캡슐화되고 최종적으로 Transport Stream Packet으로 패킷화되어 애플리케이션을 전송하는 MPEG-2 Transport Stream이 형성된다.

Data Carousel은 Download Data Messages와 Download Control Messages로 구성된다. Download Control Messages는 각 모듈과 모듈들의 집합에 대한 정보를 제공하며 DownloadServerInitiate 메시지와 DownloadInfoIndication 메시지로 구성된다.[11] 전송될 모듈의 이름, 크기, 전송 주기, Block 크기와 같은 정보가 Download Control Messages를 통해 제공되어 다음에 전송되는 모듈에 접근할 수 있게 된다. DownloadServerInitiate 메시지는 최상위의 제어 메시지이며 Data Carousel에서 수신기가 데이터를 가져올 시작지점이 된다. DownloadInfoIndication 메시지는 Data Carousel안의 모듈들에 대한 정보를 제공하고 있다. Download Data Messages는 실제 데이터를 담고 있으며 DownloadDataBlock 메시지가 사용된다. 전송될 데이터

모듈은 여러 block으로 나누어지고 각 block은 최대 4066 바이트의 크기를 가지며 block당 하나의 DownloadDataBlock 메시지로 표현된다.[1][2][11] Download Control Messages에block의 크기를 정하고 있기 때문에 모듈을 구성하는 각각의 block은 마지막 block을 제외하고는 크기가 일정해야 한다.

Object Carousel은 directory 오브젝트, file 오브젝트, stream 오브젝트 등을 사용하여 그룹화된 파일 구조를 서버로부터 수신기에 전송하는 프로토콜이다.[11] Object Carousel은 디렉토리(Director), 파일(File), 스트림(Stream), 스트림 이벤트(Stream Event), 서비스 게이트웨이(Service Gateway) 메시지를 전송한다. 디렉토리 메시지는 그 안에 포함된 디렉토리, 파일, 스트림 이벤트에 대한 정보를 담고 있다. 서비스 게이트웨이는 Object Carousel에서 최상위 디렉토리로써 메시지의 구조는 디렉토리 메시지와 같다. 파일 메시지는 애플리케이션, 이미지, 텍스트와 같은 실제 데이터를 담고 있다. 스트림 이벤트 메시지는 특정 스트림상에서 전송되고 있는 스트림 이벤트의 위치 정보를 가지고 있다. 전송되는 디렉토리와 파일 메시지는 각각 디렉토리와 파일의 내용을 담고 있는데 반해 스트림과 스트림 이벤트 메시지는 방송되는 특정 다른 스트림의 참조값을 담고 있다.

Object Carousel은 그룹화된 디렉토리, 파일, 스트림 이벤트, 스트림 메시지들이 DVB 서비스 안에서 반복적으로 전송된다.[11]

2.2Demultiplexer

본 시뮬레이터의 Demultiplexer 은 Transport Stream이 최초로 거치는 프로그램 요소로써 Transport Stream을 구성하고 있는 Transport Stream Packet이 비디오/오디오를 담고 있는 Packet인지 Section 데이터를 담고 있는 Packet인지 구분하여 데이터를 처리하는Object Carousel Protocol Parser 또는 미들웨어 시스템으로 Transport Stream Packet을 넘기는 역할을 한다. Demultiplexer 는 우선 파일 형태로 존재하는 Transport Stream을 FileInputStream으로 한 번에 188 바이트씩 Transport Stream Packet의 길이 단위로 Transport Stream을 읽어들인다.

Demultiplexer 는 읽어 들인Transport Stream Packet이 어떤 종류의 데이터를 가지고 있는지 구분하기 위해 PSI/SI 정보를 우선 획득한다. Demultiplexer 는 PAT를 우선적으로 획득하여 PAT 정보를 바탕으로 PMT, AIT, EIT와 같은 PSI/SI 정보를 획득한다. PAT는 PID가 0인 Transport Stream Packet에서 얻어온다. Demultiplexer 는 PAT 정보를 통해PMT가 전송되는 Transport Stream Packet의 PID를 알아내어 이 PID의 Transport Stream Packet으로부터 PMT정보를 얻어 낸다. PMT는 비디오/오디오와 데이터가 전송되는 Transport Stream Packet의 PID값을 제공한다. Demultiplexer 는 PMT에서 제공하고 있는 Elementary Stream의 PID값을바탕으로 Transport Stream Packet을 구분하여 Object Carousel Parser와 미들웨어 시스템에 Transport Stream Packet을 넘겨준다.

Demultiplexer 는 PSI/SI 정보를 바탕으로 Transport Stream Packet을 구분하여 Object Carousel Parser와 미들웨어 시스템에 Transport Stream Packet을 넘겨주는 것이 주된 기능이다. Transport Stream은 여러 Elementary Stream이 다중화되어 있으므로 각Elementary Stream을 실시간으로 동시에 처리하려면 이들 스트림을 처리하는 프로그램 요소(Object Carousel Parser와 미들웨어 시스템)와 Demultiplexer 는 독립적으로 동작되어야 한다. Demultiplexer 는 Transport Stream Packet을 구분하여 Elementary Stream을 처리하는 프로그램에 넘겨주고 Transport Stream Packet은 이들 프로그램 요소에서 처리하게 되는 것이다. 이들 각각은 자바 Thread로 구현한다.

PSI/SI 정보는 PAT, PMT, EIT, AIT 등 해당 Table 이름에 부합되는 클래스에 정보를 담아 관리한다. 이 Table들은 MPEG-2 Private Section에 담겨 전송되므로 MPEG2PrivateSection이라는 클래스를 정의하여 PAT, PMT, EIT, AIT 등의 PSI/SI 정보를 담고 있는 클래스가 상속하도록 구현하였다.

PSI/SI Table들의 정보를 추출하기 위해서는 이 Table을 담고 있는 Section 구조 전체를 얻어야 한다. 하나의 Section을 전송하고 있는 같은 PID의 Transport Stream Packet을 SectionBuffer 클래스로 구현한Buffer에 임시 저장한다. 이 SectionBuffer는 하나의 Section을 전송하고 있는 같은 PID의 Transport Stream Packet을 담을 수 있는 Vector를 가지고 있다. Section을 전송하고 있는 Transport Stream Packet을 모두 얻은 후에는 각 Transport Stream Packet을 Packet Header의 continuity_counter 값에 따라 재배열하고 Packet header를 제거하여 하나의 Section 구조를 얻어낸다. 시뮬레이터는 얻어낸 Section 를 비트 연산 처리하여 PSI/SI에서 제공하고 있는 프로그램 정보를 얻어낸다.

2.3ObjectCarouselProtocolParser

Xlet 애플리케이션은 Transport Stream에 Object Carousel Protoco로 인코딩 되어있다. 본 시뮬레이터의

Object Carousel Parser는Object Carousel 프로토콜을 해석할 수 있도록 구현되었다. 따라서 Object Carousel Parser는 Transport Stream으로부터 Xlet 클래스와 Xlet 클래스가 사용하는 파일 및 데이터를 추출할 수 있다. Xlet애플리케이션이 전송되는 Transport Stream Packet의 PID는 PMT에Stream type이 0x0B인 Elementary Stream Loop에서 제공된다. 따라서 Demultiplexer 는 PMT를 통해 애프리케이션을 담고 있는Transport Stream Packet을 Object Carousel Protocol Parser에 넘겨준다. Object Carousel Parser이 Xlet 클래스와 Xlet 클래스가 사용하는 파일 및 데이터를 추출하는 과정은 다음과 같다 거친다.

첫번째, Download Control 메시지인 DSI와 DII 메시지를 우선적으로 확보한다. 이들 메시지에 모듈에서 오브젝트를 인식할 수 있는 참조값이 담겨 있기 때문이다. DSI, DII 메시지는 Section Header의 tableId와 dsmccMessageHeader의 messageId을 통해 이들 메시지를 구별한다.[11] DSI, DII, DDB 메시지의 tableId와 messageId는 다음 표 1에나타내었다.

메시지	tableId	messageId
DSI	0x3B	0x1006
DII	0x3B	0x1002
DDB	0x3C	0x1003

표1 DSI, DII , DDB 메시지의 tableId와 messageId

DSI, DII 메시지는 각각 다른Section에 담겨 전송된다. Object Carousel Protocol Parser는 이런 DSMCC Message를 담고 있는 Section을 얻기 위해 Demultiplexer로부터 넘겨오는 Transport Stream Packet을 SectionBuffer 클래스의 Vector에 넣는다. Object Carousel Protocol Parser가 DSMCC Message를 담고 있는 Section을 얻기 위한 과정은 Demultiplexer 가 PSI/SI Table을 담고 있는 Section을 얻기 위한 과정과 동일 하다.

두번째, 하나 이상의 DDB 메시지로 나누어져 있는 모듈을 완전히 복원하는 것이다. Object Carousel에 전송되어 오는 파일, 디렉토리, 스트림 이벤트, 서비스 게이트웨이, 스트림 오브젝트가 BIOP 메시지 형태로 모듈에 담겨져 있으므로 각 오브젝트를 얻으려면 모듈 전체를 복원해야 한다.[11] DSMCC Message를 담고 있는 Section을 Transport Stream으로부터 복원하여 표1의 나와 있는 Section Header의 tableId와dsmccMessageHeader의 messageId을 통해 DDB 메시지를 Section으로부터 얻는다. 하나의 DDB 메시지는 MPEG-2 Section과 일대일 대응이 되므로 Section Header의 Section Length 값을 Section Header를 비트 연산 처리하여 얻어와 개별 DDB 메시지를 획득한다. 전체 모듈을 구성하는데 필요한 DDB 메시지의 개수는 Section Header의 last section number 값을 참조하여 알아낸다. 하나의 모듈을 구성하고 있는 DDB 메시지의 moduleId 값은 동일하므로 동일한 moduleId 값을 갖는 DDB 메시지를 하나의 Module Buffer를 생성하여 여기에 담아 보관한다. DDB 메시지를 담고 있는 Section을 Section Header의 section number값에 따라 배열하고 DDB 메시지의 페이로드에 들어있는 데이터들을 서로 연결하면 전체 모듈이 복원된다.

세번째, 복원된 모듈로부터 각 오브젝트를 얻어내어 이를 자료구조에 담아 제공한다. 이전 과정에서 얻어온 모듈은 BIOP 메시지들로 이루어진 비트 스트림이다. 이런 비트 스트림인 모듈에서 각각의 BIOP메시지를 추출하기 위해서는 BIOP 메시지 헤더 정보를 이용한다. BIOP 메시지 헤더의 message size 값은 BIOP 메시지의 길이를 나타내므로 message size 값만큼 모듈로부터 비트스트림을 얻어와 모듈로부터 각각의 BIOP 메시지를 분리한다. BIOP 메시지 헤더의 objectKind_data 값을 통해 분리된 BIOP 메시지가 담고 있는 오브젝트의 종류를 알아낸다. 다음 표는 objectKind_data 값에 따른 오브젝트의 종류를 나타낸다.

objectKind_data	오브젝트 종류
0x64697200	Directory 메시지
0x66696C00	File 메시지
0x73747200	Stream 메시지
0x73726700	ServiceGateway 메시지
0x73746500	StreamEvent 메시지

표 2 오브젝트의 objectKind_datavalue

디렉토리와 서비스 게이트웨이가 참조하고 있는 파일들과 스트림 이벤트는 object_key값으로 식별한다. 모듈로부터 얻어낸 오브젝트들은 자료 구조에 담아 그 주소 값을 애플리케이션 매니져에게 넘겨 준다. Xlet Class들도 모듈에서 얻어낸 오브젝트들에 포함되어 있다. DVB-MHP 규약

에서 애플리케이션 매니저는 셋탑박스의 미들웨어에 내장되어 Xlet의 Life Cycle을 관장하는 프로그램이다.

2.4미들웨어 시스템

본 논문에서 구현한 미들웨어 시스템은 크게MHP API와 비동기적 처리를 위한 쓰레드로 구성된다. 본 논문의 미들웨어 시스템에서 구현한 MHP API는 JavaTV API와 DVB API에서 스트림 이벤트를 처리하고, MPEG-2 Section의 각종 데이터를 얻어오는데 필요한 API들 위주로 구현한다. 즉Transport Stream의 Section에서 발생하는 이벤트를 처리하기 위한 것이 본 시뮬레이터의 미들웨어 시스템의 주된 구현 부분이다.

Xlet이 MPEG-2 Section으로 전송되어 오는 Section 데이터와 PSI, SI 정보를 요청하는 경우 미들웨어 시스템은 이런 요청을 비동기적으로 처리한다. 즉 Xlet이 PSI, SI 정보 또는 어떤 Section 데이터를 요청하면 미들웨어 시스템은 요청된 데이터를 스트림에서 획득하였을 때 이벤트를 발생시켜 Xlet에게 이 사실을 통보함으로써 이 과정을 비동기적으로 처리하게 된다. Xlet은 PSI, SI 정보 또는 어떤 Section 데이터를 요청할 때 Event Listener를 미들웨어 시스템에 등록해야 한다. Xlet에서 등록한 Event Listener가 미들웨어 시스템이 Xlet에서 요청한 데이터를 획득하였을 때 발생시키는 이벤트를 처리하기 때문이다. 미들웨어 시스템은 Xlet이 등록한 Event Listener의 객체 주소값을 저장한다. 저장하고 있는 Event Listener의 객체 주소값으로 미들웨어 시스템이 이벤트를 Xlet에 통보한다. Xlet이 스트림 이벤트를 이용하는 경우, 우선 Xlet은 Event Listener를 등록하고 미들웨어 시스템은 Object Carousel로 전송되어 오는 스트림 이벤트 오브젝트를 Object Carousel Parser로부터 얻어 온다. 미들웨어 시스템은 스트림 이벤트 오브젝트로부터 스트림 이벤트의 위치 정보를 얻어와 특정 스트림을 감시하기 시작한다. 미들웨어 시스템이 스트림에서 스트림 이벤트를 감지하면 스트림 이벤트를 발생시킨다. 스트림 이벤트가 발생하면 등록된 Event Listener가 이벤트를 처리한다.

3.결론

본 논문의 에뮬레이터는 Transport Stream에 Object Carousel의 BIOP 메시지 형태로 존재하는 Xlet을 인식하고 이를 PC에서 동작하도록 Object Carousel Parser와 미들웨어를 구현하였다. 현재Xlet을 PC에서 구동시키는 에뮬레이터들은 Transport Stream을 처리하는 것이 아니라 PC에 자바 클래스 파일형태로 존재하는 Xlet을 구동시키고 있다. 본 논문에서 구현하는 에뮬레이터는 실제 방송 스트림인 Transport Stream을 처리하여 Xlet을 PC에서 구동시키기 때문에 기존Xlet 에뮬레이터들이 애플리케이션 개발자들을 위해 제공하는 Xlet 개발 도구 기능 이외에 일반 사용자가 데이터 방송을 PC에서도 이용할 수 있는 시스템을 제공한다. 또한 Transport Stream을 처리하지 않는 기존 에뮬레이터는 방송 스트림에 담겨오는 각종 정보를 Xlet에게 제공하지 못하는 한계를 극복한다.

4.Reference

1.Richard S. Chernock, Regis J. Crinon, Michael A. Dolan, John R. Mick JR. "Data Broadcasting/Understanding The ATSC Data Broadcast Standard." McGraw Hill 2001

2.ISO,"Information Technology Generic Coding of Moving Pictures and Associated Audio: Digital Storage Media Command and Control - ISO/IEC 13818-6 International Standard".ISO/IEC JTC1/SC29/WG11 MPEG96/N1300p1

3. ETSI EN 300 468 V1.4.1 (2000-11) : Digital Video Broadcasting (DVB); Specification for Service Information (SI) in DVB systems

4.ETSI TS 102 812 : DVB Multimedia Home Platform (MHP) 1.1, July 2001

5.Sun Microsystems, "Java TV API Specification, Version 1.0 " January 2000

6. P.A. Sarginson, B.Sc. "MPEG-2 : Overview of the systems layer" Research & Development Department Policy & Planning Directorate THE BRITISH BROADCASTING CORPORATION

7. The MHP Tutorial : http://www.mhp-interactive.org

8.Gerard O. "The Essential Guide to Digital Set-top Boxes and Interactive TV." Prentice Hall PTR, 2000.

9.Jos C. Lpez-Ardao,Cndido Lpez,Alberto Gil, Rebeca Daz,Ana Fernndez,Manuel Fernndez, Andrs Surez and Javier Muoz1 "EXPERIENCES FROM IMPLEMENTING A MHP RECEIVER". ETSE Telecomunicacin, University of Vigo (Spain) 2001

10.C. Peng and P. Vuorimaa. "DIGITAL TELEVISION APPLICATION MANAGER" Telecommunications Software and Multimedia Laboratory,Helsinki University of Technology. 2001 IEEE. Reprinted, with permission, from Proceedings of the IEEE International Conference on Multimedia and Expo 2001, Tokyo, Japan, August 22-25, 2001

11.ETSI TR 101 202 V 1.2.1(2003-1) : Digital Video Broadcasting (DVB); Implementation guidelines for Data Broadcasting
12.Chengyuan Peng,Pablo Cesar,and Petri Vuorimaa "INTEGRATION OF APPLICATIONS INTO DIGITAL TELEVISION ENVIRONMENT " Telecommunications Software and Multimedia Laboratory,Department of Computer Science and Engineering,Helsinki University of Technology. the 7th International Conference on Distributed Multimedia Systems, September 26-28, 2001
13.Ganesh Sivaraman, Pablo Cesar, and Petri Vuorimaa "SYSTEM SOFTWARE FOR DIGITAL TELEVISION APPLICATIONS" Telecommunications Software and Multimedia Laboratory Helsinki University of Technology. 2001
14.ISO,"Information Technology Generic Coding of Moving Pictures and Associated Audio: System - ISO/IEC 13818-1 International Standard". ISO/IEC JTC1/SC29/WG11 MPEG96/N1300p1
15.J.-P. Evain, "The Multimedia Home Platform an overview," EBU Technical Review, Spring 1998
16.Bart C., Jon C., Bill F., Linda K., David R., James V., and TaoY. "Java TV API Technical Overview". The Java TV API Whitepaper Version 1.0, Release Candidate D, July 11, 2000.