

Program and System Information Protocol (PSIP) 해석기 설계

Design of a Program and System Information Protocol (PSIP) Parser

최미란*, 최성중**

*(주)디지털스트림테크놀로지, **서울시립대학교 전자전기공학부

요약

본 논문의 목적은 Electronic Program Guide(EPG)용 응용프로그램을 위한 Application Program Interface(API) 및 이를 위한 Middleware의 구현이다. 이를 위해 Windows 환경 하에서 프로그램 정보 전송 프로토콜인 Program and System Information Protocol(PSIP)을 해석하는 Parser를 구현하고, 해석한 자료를 저장하는 Database를 설계하였다. 구현된 프로그램의 테스트를 위해 디지털 방송 수신 카드를 사용한 테스트베드를 구축하였고, 지상파 디지털 방송에서 수신된 MPEG-2 Transport Stream을 사용하여 구현된 시스템을 검증하였다. 본 논문에서 개발된 Middleware는 EPG를 위한 응용프로그램 및 향후 예약시청, 예약녹화 등의 부가 서비스를 위해 활용될 수 있다.

1. 서론

국내 지상파 디지털 TV 방송은 1999년 6월부터 실시한 실험방송을 마치고, 2000년 9월부터 시험방송에 들어갔다. 2001년 1월부터는 수도권을 대상으로 본 방송이 시작될 예정이다.

디지털 TV 방송으로 인해 시청자들이 처음 느끼는 이점은 화질과 음향의 차이일 것이다. 기존의 아날로그 방송에 비해 선명한 화질과 영화관에서 느낄 수 있는 음질을 가정에서 TV로 경험할 수 있다. 그러나, 디지털방송이 기존 아날로그 방송에 비해 가장 큰 이점은 디지털 데이터 전송이다. 데이터 전송으로 인해 시청자들은 TV를 보면서 주식정보나 날씨 등을 확인할 수 있고, 스포츠를 시청하면서 출전선수의 기록을 확인해 볼 수 있다. 이러한 디지털 전송을 사용하는 서비스는 무한한 가능성을 갖고 있다.

이러한 데이터 방송의 첫 단계는 Electronic Program Guide(EPG)이다. EPG는 신문을 통해 확인할 수 있었던 방송에 대한 정보가 TV신호에 함께 전송됨으로써 시청자들은 어떤 채널에서 어떤 프로그램이 나오는지, 영화의 주인공은

누구인지, 주인공의 프로필은 어떻게 되는지 등의 정보를 TV시청 중에 확인할 수 있다. 이 자료를 시청자에게 보여주기 위해서는 전송되는 TV 신호에서 데이터를 추출하여, 의미 있는 정보를 제공하는 시스템이 필요하다.

이를 위해 본 논문은 국내 디지털 방송의 근간인 ATSC T3(Technology Group on Distribution)에서 채널 정보와 프로그램 정보를 전송하기 위해 제정한 Program and System Information Protocol(PSIP)을 해석 저장하는 Middleware와 응용프로그램을 위한 API를 설계하여 Microsoft Windows 환경 하에서 구현하였다. 구현된 Middleware는 C++로 구현되어 있어 다른 시스템에 이식이 간단하며, 하위 계층인 DEMUX가 PSIP section을 추출하는 기능만 있으면 어떠한 디지털 TV 수신 장치에서도 사용 가능하다.

본 논문의 구성은 다음과 같다. 제 2장에서는 시스템을 구현하는데 필요한 여러 표준들에 대해 간략히 서술하고, 제 3장에서는 시스템모델, API, Database를 구현하는 과정을 구체적으로 설명하였다. 그리고 제 4장에서는 구현된 API를 테스트하기 위한 환경과 테스트 결과를 서술하였다. 마지막으로 제 5장은 결론과 앞으로의 연구방향을 제시하였다.

2. 관련 기술 및 표준

2.1. MPEG-2 systems

MPEG-2 시스템 표준[1]에서는 영상데이터와 음성데이터, 혹은 그 외의 데이터들을 전송에 적합하도록 하나의 스트림으로 다중화하는 방법을 기술한다. Transport Stream(TS)은 이러한 다중화 방법의 하나로 주로 디지털 방송용 스트림 전송에 사용된다. TS를 구성하는 패킷들은 188바이트의 고정 길이를 가지고 있으며 각 패킷은 4바이트의 고정헤더와 가변적인 확장헤더(adaptation field), 실제 전송하고자 하는 데이터(payload)로 구성되어 있다. 영상 및 음성 외의 데이터 전송을 위해 MPEG-2 시스템에서는

Table이 정의되어 있다. 이러한 Table을 TS로 전송하기 위해 MPEG-2 시스템에서 정의된 Section 구조체를 이용한다. 하나의 Section은 최대 1K 또는 4K바이트를 전송할 수 있다. 만약 보내야 할 Table이 최대 전송 크기를 넘을 경우 데이터를 같은 version number를 갖는 여러 Section으로 나누어 전송한다.

TS에 포함되어 있는 여러 프로그램의 자세한 내용을 전달하기 위해 여러 Table이 정의되어 있는데 이를 Program Specific Information(PSI)이라 한다. 이러한 PSI로 Program Association Table (PAT), Program Map Table(PMT), Network Information Table(NIT), Conditional Access Table(CAT)이 정의되어 있다. PAT는 하나의 MPEG-2 TS에 다중화되어 있는 Program에 대해 설명한다. PMT에는 하나의 Program을 구성하는 program element들의 형식(영상, 음성, 사용자 데이터)과 PID를 설명한다.

2.2. Program and System Information Protocol (PSIP)

PSIP는 ATSC(Advanced Television Systems Committee)에서 정의하고 있는 규격이다[2-4]. PSIP를 설명하기 위해 ATSC규격에서 사용하고 있는 용어의 정의가 필요하다. 이는 일반적으로 사용하고 있는 용어와 차이가 있기 때문이다. ATSC에서 정의한 Event는 시작 시간과 길이가 정해져 있는 요소 스트림 (elementary stream)들의 집합이다. 이는 일반적으로 사용하는 TV program과 동일한 개념이다. 하지만 MPEG-2 표준에서 정의된 Program은 서로 연관된 A/V 스트림의 집합을 의미한다. MPEG-2 Program은 ATSC Virtual Channel (VC)과 같은 개념이다. VC는 기존 6MHz bandwidth를 갖는 아날로그 방식의 한 채널에 다수의 디지털 채널을 전송할 수 있기 때문에 새로이 도입된 채널 개념이다. 본 논문에서는 되도록 "program"의 사용을 피하고, ATSC에서 정의된 Event, VC의 개념들을 사용한다.

PSIP를 사용하여 전송되는 정보는 MPEG-2 Table로 표현되며, 프로그램을 구성하는 별도의 영상, 음성 데이터와 함께 TS로 다중화되어 전송된다. 그림 1은 PSIP에서 정의한 Table의 계층적 구조이다. 그림 1에 나타난 Table 중 System Time Table(STT), Master Guide Table(MGT), Virtual Channel Table(VCT), Rating Region Table(RRT)은 고정 PID를 사용하고, Event Information Table(EIT)과 Extended Text Table(ETT)은 MGT에 기록된 PID를 사용하여 TS에 수록된다.

PSIP를 사용하여 전송되어지는 정보는 크게 시스템 정보, 이벤트 정보, 메타 정보의 세가지

로 정의될 수 있다. 시스템 정보는 하나의 TS에 다중화되어진 여러 VC에 대한 내용이다. 이를 위해 VCT와 ETT를 사용한다. VCT에서는 채널 이름과 같은 시청자를 위한 정보와 각 채널을 구성하는 여러 Elementary stream의 타입과 PID와 같이 수신에 필요한 내용이 포함되어 있다. VCT는 MPEG-2 시스템의 PAT를 확장한 개념이다. 각 VC에 대한 자세한 문자 정보는 ETT를 사용하여 전송된다.

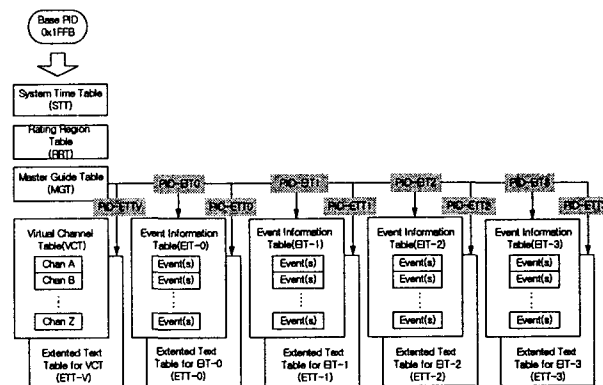


그림 1. PSIP 테이블의 계층도

이벤트 정보는 하나의 VC에 포함된 event들에 대한 정보이다. 즉, 일반적으로 Electronic Program Guide(EPG)라는 서비스를 위한 정보이다. 이를 위해 EIT, ETT, STT, RRT를 사용한다. EIT는 Virtual 채널에서 방송되고 있거나 앞으로 방송될 프로그램에 대한 정보를 시간대별로 안내하기 위해 사용되는 Table이다. 이것은 시간대가 다른 여러 지역에 같은 방송을 전달하기 위해 전 세계적인 기준 시각(Coordinated Universal Time, UTC)을 사용하며, 하나의 EIT는 세 시간 동안의 프로그램 안내를 수록하고 있다. ETT는 EIT에 수록된 프로그램에 대한 자세한 부가 설명을 전송하기 위해서 사용된다. STT는 현재 날짜와 시간을 전송하며, 이 시간 정보는 프로그램 안내에 기록된 시간의 기준이 된다.

메타 정보는 각 PSIP에 정의된 테이블에 대한 내용을 갖고 있다. 이를 위해 MGT를 사용한다. MGT에는 STT를 제외한 테이블들의 메타 정보를 제공한다. 중요한 메타 정보로서 EIT와 ETT의 PID, 각 Table의 버전 및 크기가 있다. MGT도 자신의 버전 정보를 가지고 있는데, 다른 테이블들의 내용이 변경될 때마다 새로운 값을 가지게 된다. 따라서, 수신측에서는 항상 MGT의 버전을 확인하여 새로운 버전일 경우만 MGT 내용을 확인하여 새 정보를 제공 받는다.

3. Software Design

3.1. 시스템 모델 정의

우리나라 지상파 디지털 TV방식은 미국 ATSC 방식에 근간을 두고 있다. ATSC T3에서는 방송에 관련된 시스템 정보와 프로그램 정보를 전송하는 규격으로 PSIP를 제정하였고 이는 표준 문서 A/65, A/66, A/67에 기술되어져 있다[2-4]. 디지털 방송에서 제공되는 시스템 정보와 프로그램 정보를 응용프로그램에 제공하기 위해서 PSIP를 해석하는 Parser가 필요하다.

본 논문에서 사용된 ATSC 지상파 디지털 수신기는 MS Windows기반의 PC와 ATSC 지상파 디지털 방송 수신 카드로 구성되어 있다. 이 카드는 복조, MPEG-2 Demultiplex 기능을 갖추고 있다. 본 논문에서는 ATSC 표준을 준수하고 Windows 시스템에서 동작하는 PSIP 시스템 모델을 그림 2와 같이 정의하였다. PSIP 시스템 모델은 세 개의 계층과 크게 세 개의 API로 구성되어 있다.

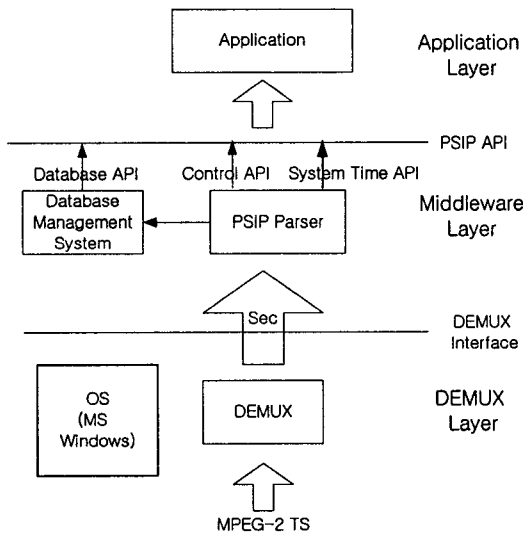


그림 2. PSIP 시스템 모델

DEMUX 계층은 상위 계층이 특정 PID의 section을 요구하면 수신하고 있는 TS에서 요구된 PID의 section만 역다중화하여 정해진 Buffer에 저장한다.

Middleware는 PSIP Parser와 Database Management System(DBMS)으로 구성된다. PSIP Parser는 DEMUX 계층에 section을 요구하고, DEMUX가 저장해 놓은 Buffer에서 section을 가져와 각 Table 별로 통합 해석하고 이 결과를 Database에 저장한다. DBMS는 PSIP Parser에서 보낸 자료를 저장하고, Application이 요구하는 자료를 제공한다.

본 논문에서는 PSIP Parser가 해석한 자료를

저장하는 수단으로 기존의 Database Engine (MS Jet Engine)을 채택하였다[5]. 이는 자료의 저장, 수정, 검색이 파일을 사용하는 것보다 기존의 DBMS에서 제공되는 편리한 API를 사용함으로써 쉽게 구현이 가능하며, 별도의 추가 비용 없이 사용가능하기 때문이다. 특히, PSIP 자료 중 프로그램 안내 정보의 경우, 최소 12시간의 자료가 기본이며 최대 16일 동안의 자료를 저장할 수 있어야 한다. 이 자료를 체계적으로 관리하고, 채널조회, 채널별 프로그램 조회, 프로그램별 상세정보 조회 등 사용자의 다양한 요구에 부응하기 위해서는 DBMS가 필요하다. 또한, DBMS를 사용하면 단순 조회 기능 외에 사용자의 시청 습관을 반영한 특정 프로그램의 예약, favorite 채널/프로그램 관리와 같은 부가 서비스 제공이 용이하게 된다.

Application 계층은 사용자에게 채널/프로그램 정보를 Graphic User Interface(GUI)로 제공하며, 이외의 부가 서비스를 사용자에게 제공한다.

본 논문에서는 DEMUX가 S/W나 H/W로 이미 존재한다고 가정을 한다. 또한, 이 DEMUX에는 PSIP Parser가 요청하는 PSIP section을 제공하는 DEMUX Interface가 이미 정의되어 있다고 가정한다.

3.2. PSIP API 설계

본 논문에서 구현해야 할 PSIP API는 세 종류로 구분된다. Application에서 PSIP parser의 동작을 제어하는 Control API와 Application에서 DBMS에 필요한 자료를 요구하는 Database API, 마지막으로 Application이 System Time을 요청하고 제공하는 System Time API이다. PSIP Table 중 STT의 내용은 Database에 저장하는 것이 의미가 없으므로 해석하는 즉시 Application에 올려주는 경로를 별도로 만들었다. 구현해야 할 대표적인 API를 표1, 표2, 표3에 정리했다.

표1. PSIP API 중Control API

함수명	기능
Open()	PSIP Parser의 Object를 생성
ChannelRun()	PSIP Parser가 채널 정보를 해석하여 Database에 저장함
ChannelStop()	PSIP Parser가 채널 정보 해석하는 작업을 중지 시킴
EPGRun()	PSIP Parser가 프로그램 정보를 해석하여 Database에 저장함
EPGStop()	PSIP Parser가 프로그램 정보 해석하는 작업을 중지 시킴
Close()	PSIP Parser가 사용했던 Object 삭제

표2. PSIP API 중Database API

함수명	기능
EPGOpen()	EPG Component의 Handle을 가져옴
EPGClose()	EPG Component를 Release 시킴
GetProgramData()	프로그램 정보를 가져옴
GetEventData()	프로그램 상세 정보를 가져옴
GetChannelData()	채널 정보를 가져옴
GetCategoryData()	프로그램의 장르를 가져옴
GetChannelPIDData()	Virtual 채널별로 스트림과 PID를 가져옴
DeleteChannelAndPIDData()	Virtual 채널과 해당 채널의 관련 PID를 Database에서 삭제함
DeleteChannelPIDData()	특정 Virtual 채널과 관련된 PID 데이터를 Database에서 삭제함
DeleteChannelData()	특정 Virtual채널 데이터를 Database에서 삭제함
SearchProgramData()	Application에서 입력된 조건의 프로그램을 검색함
CheckLanguage()	특정 Virtual채널에서 방송되는 언어를 가져옴

표3. PSIP API 중System Time API

함수명	기능
STTOpen ()	STT를 해석하는 Component의 Handle을 가져옴
GetSTT()	PSIP Parser가 STT를 해석하여 현재시간을 보내줌
STTClose ()	STT를 해석하는 Component의 Handle을 Release 시킴

3.3. PSIP Parser 구현

PSIP의 parsing을 위해 가장 중요한 테이블은 MGT이다. MGT에는 다른 테이블들의 버전 정보와 EIT, ETT에 대한 PID와 같은 메타 정보가 포함되어 있기 때문이다. 그림 3은 MGT의 State Diagram을 나타낸다. State는 Dead State, Start State, Update State 세 가지로 구분하였다. Dead State는 디지털 TV 수신기가 Power off 되어 있는 State이다. Start State는 디지털 TV 수신기가 Power On 되거나 Physical Channel이 변경되어 PSIP Parser가 MGT section을 DEMUX에 요청하는 State이다. Update State는 받은 MGT가 새로운 version임을 확인하고, VCT, EIT, ETT 등을 해석하여 Database에 저장하는 상태이다. Dead State에서 디지털 TV 수신기를 Power On 하면 Start State로 전이한다. Start State에서 받은 MGT의 버전이 이전 MGT의 버전과 같거나, MGT 외의 테이블을 받으면 Old MGT Event가 발생하여 다시 Start State로 전이한다. 받은

MGT의 버전이 새로운 버전의 MGT이면 New MGT Event가 발생하여 Update State로 전이한다. Update State에서는 MGT에 들어있는 테이블들의 버전과 각 테이블들의 기존 버전을 비교하여 버전이 달라진 테이블의 section을 DEMUX에 요청한다. 요청한 section을 받으면 해석하여 필요한 내용을 Database에 저장하고 End Update Event를 발생하여 Start State로 돌아간다.

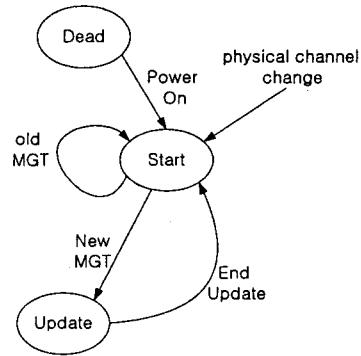


그림 3. MGT의 State Diagram

PSIP Parser에서 STT를 처리하는 루틴은 다른 PSIP 처리 루틴과 독립적으로 동작한다. Application에서 GetSTT()를 호출하면 STT section을 DEMUX에 요청하고 요청한 STT section을 받으면 해석해서 시간 정보를 Database에 저장하지 않고 바로 Application에 Return한다.

3.4. Database 설계

설계한 Database의 주요 Entity들의 구성과 관계를 그림 4에 나타내었다.

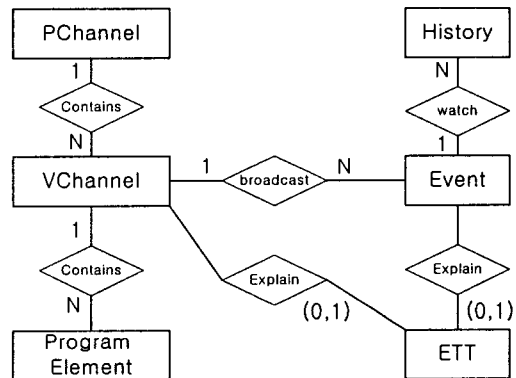


그림 4. Database의 주요 Entity-Relation Diagram

PChannel은 하나의 아날로그 채널 (Physical Channel) 정보를 나타내고 VChannel은 각 가상 채널 (Virtual Channel)의 정보를 나타낸다. 하나의 아날로그 채널은 다수 개의 가상 채널을 포함할 수 있다. Program Element는 채널별로 전송되는 Stream Type과 PID를 저장한다. Event는 가상 채널에서 방송되는 프로그램과 시간을 저장

하고 ETT는 프로그램별 상세정보나 가상 채널의 상세정보를 저장한다. History는 시청자가 시청한 프로그램을 기록한다. History는 EPG의 기능을 단순히 전송된 정보를 보여주는 단계를 넘어 사용자의 시청 습관을 파악하여 자동적으로 Favorite 채널을 등록하거나 하는 부가적인 기능에 이용될 수 있도록 한다.

3.5. Class 설계

모든 PSIP 테이블들은 table_id 부터 protocol_version까지 동일한 구조를 가지는 section header와 각 테이블마다 고유한 부분인 section payload로 구성되어 있다. 클래스를 설계할 때 이 부분을 고려하여, 모든 테이블이 공통적으로 가지는 header 부분을 SectionHeader 클래스로 만들고, 각 테이블에서 고유한 payload부분을 각 테이블 클래스로 만들었다. 테이블 중 VCT나 MGT처럼 for loop로 구성되어 있는 테이블은 for문 안의 내용을 하나의 클래스로 만들고, for문의 개수만큼 리스트로 처리한다. 클래스의 계층구조를 보면 그림 5와 같다

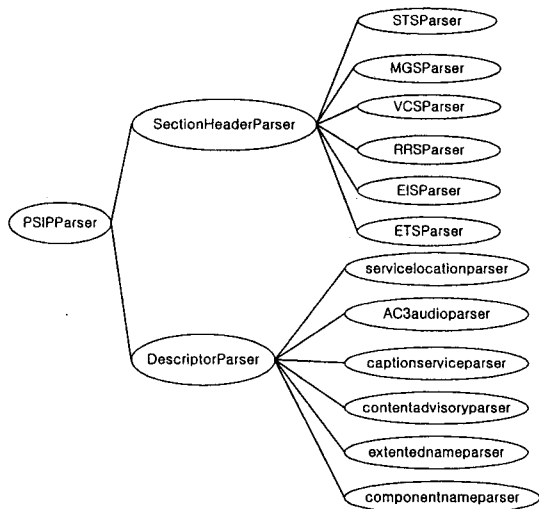


그림 5. 클래스 계층구조

4. 테스트 및 결과

4.1. 테스트 환경

본 논문에서 구현된 시스템의 테스트를 위해 인텔(Intel)의 펜티엄 III 733MHz CPU가 장착된 PC를 사용했으며, TV 시청에 기본적으로 지원이 필요한 하드웨어로 컴퓨터에 장착된 사운드 카드(Sound Card)와 그래픽 카드(Graphic Card)를 사용하였다. 프로그램 동작을 위한 운영체제는 Win98 SE를 사용하였고, 구현에 사용된 소프트웨어는 마이크로소프트(Microsoft)의

Visual C++ 6.0이다.

국내 지상파 디지털 TV는 아직 시험단계에 있다. 현재 PSIP 데이터를 보내는 채널은 KBS, MBC에 국한되며 SBS는 간헐적으로 PSIP 자료를 보내고 있는 실정이다. PSIP 자료를 보내고 있는 KBS와 MBC도 시스템 정보만 정상적으로 전송할 뿐, 아직 시험 방송이므로 이벤트 정보는 정상적으로 전송하지 않는다. 이러한 이유로, 실험 환경으로 두 가지 방법을 사용하였다. 지상파 수신이 가능한 시스템 정보 실험을 위해서는 그림 6의 시스템 정보 실험 모델을 사용하였다. 지상파를 수신할 수 있는 안테나를 설치하고, 디지털 TV 수신 장치를 이용하여 방송을 수신하면서 PSIP section을 받아서 해석하는 방법을 이용하였다. 지상파 수신으로 실험이 불가능한 이벤트 정보 실험을 위해서는 그림 6의 이벤트 정보 실험 모델을 사용하였다. 이를 위해 이벤트 정보가 포함되어 있는 테스트 스트림을 스트림 서버에서 RF출력을 시킨다. 디지털 TV 수신 장치에서는 이 신호를 수신하여 EIT, ETT등을 추출하고 해석하였다.

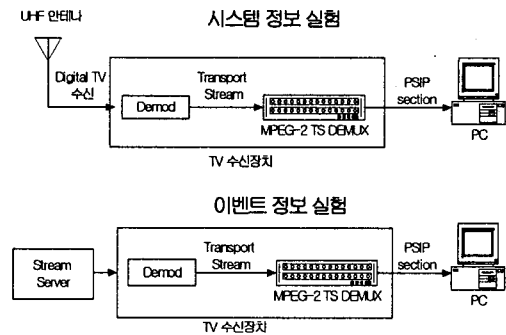


그림 6. 테스트 환경

4.2. 실험결과

본 논문에서는 구현된 API와 PSIP Parser를 검증하기 위해 간단한 Application을 구현하였다. 그림 7은 시스템 정보를 보여주는 화면이다. 채널 항목에서 A는 아날로그 채널을, D는 디지털 TV 채널을 나타내며, 현재 국내에서 방송되는 디지털 채널로 14, 15, 16번이 있음을 보여준다. 14-1로 나타낸 것은 Major Channel이 14번이고, Minor Channel이 1임을 나타낸다. 채널 정보를 나타낼 때, 특이한 점은 VCT의 Major Channel, Minor Channel 외에 RF 채널도 나타내었다. 이름항목은 VCT의 Short name을 나타내고, 설명은 VCT Descriptor 중 extended channel name descriptor의 내용을 나타내었다.

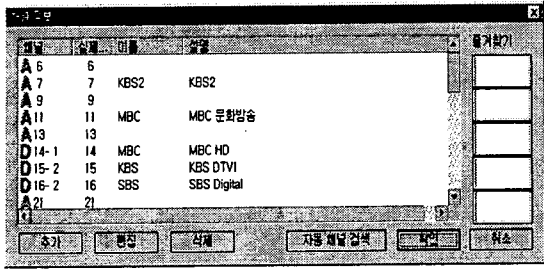


그림 7. 채널 정보

그림 8은 EPG 메인 화면이다. 프로그램 정보는 실제 정보가 전송되지 않아 임의의 프로그램 정보를 DB에 입력한 후, 구현된 API를 테스트하였다. 화면 하단에는 VCT의 short name과 EIT로 전송된 각 채널별 프로그램 정보를 시간별로 나타내었다. 화면 상단에는 ETT로 전송된 프로그램별 상세 정보를 나타내었다. 프로그램 이름에 커서가 위치하면 해당 프로그램의 상세정보가 화면 상단에 나타난다.

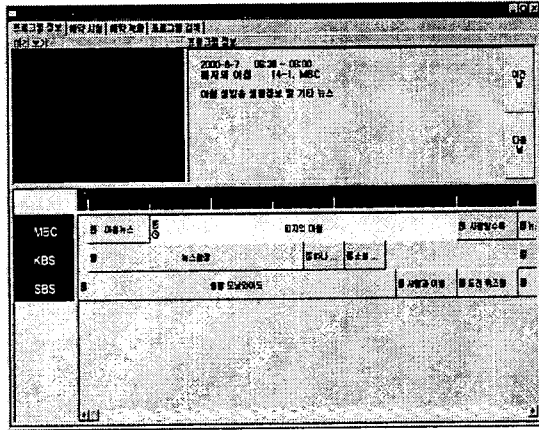


그림 8. EPG 메인 화면

그림 9는 EIT로 전송된 프로그램 정보가 저장되어 있는 DB를 이용해 사용자가 원하는 프로그램을 검색하는 화면이다. 검색 조건으로는 프로그램 제목에 포함되어 있는 특정 검색어를 이용하는 방법과 시간대를 이용하는 방법, 프로그램 분류를 이용하는 방법을 구현하였다.

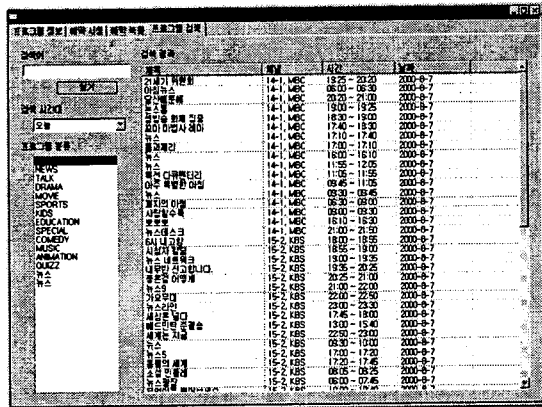


그림 9. 프로그램 검색 화면

5. 결론

본 논문에서는 Windows환경에서 디지털 TV 방송으로 전송되는 채널/프로그램 정보인 PSIP를 해석하여 응용프로그램에 제공하는 API를 구현하였다. 구현된 시스템을 테스트하기 위해 현재 방송 중인 디지털 방송을 수신하는 응용 프로그램을 개발하여 기능을 확인하였다.

아직은 국내 디지털 TV 방송에서 PSIP 전송에 미비한 점이 많지만, 디지털 TV 방송이 안정화되는 단계에서는 단순한 Audio-Visual 전송 기능에 부가된 데이터 전송의 이점을 이용한 서비스에 중점을 둘 것이다. 본 논문은 이때를 대비하여 기본적인 EPG 기능을 시청자에게 편리하고 효율적인 방법으로 제공하기 위한 방법을 구현했다.

앞으로의 연구 방향으로 데이터 방송의 이벤트 안내를 위한 Data Event Table(DET)과 데이터 서비스 정보를 위한 Data Service Table(DST)을 본 시스템에 추가할 예정이다.

참고 문헌

- [1] ISO/IEC 13818-1, "Information Technology - Generic coding of moving pictures and associated audio information: System", International Standard, 1994
- [2] ATSC Document A/65, "Program and System Information Protocol for Terrestrial Broadcast and Cable", ATSC, 1997
- [3] ATSC Document A/66, "Corrigendum No.1 to PSIP for Terrestrial Broadcast and Cable A/65", ATSC, 1999
- [4] ATSC Document A/67, "Amendment No.1 to PSIP for Terrestrial Broadcast and Cable A/65", ATSC, 1999
- [5] <http://www.microsoft.com/sql/techinfo/msdejet.htm>