특집

디지털 위성방송 응용 개발을 위한 MHP API

정 문 열* 류 일 권**

목 차

1. 서 론

디지털 방송은 다채널과고화질 서비스이외에도 데이터 방송(data broadcasting)을 가능하게 한다. 데이터 방송은 데이터를 비디오와 함께 방송하여 시청자에게 전달하는 것을 의미하는데, 여기서 데이터라함은 실행 가능한(executable) 애플리케이션 프로그램과 애플리케이션이 사용하는 텍스트나 이미지 파일과 같은 데이터를 총칭한다. 데이터 방송 표준으로는 북미의 ATSC-DASE[1], DASE를 대체할 것으로 예상되는 ATSC-ACAP[2], 케이블 방송의 데이터 방송 표준인 OpenCable-OCAP[3], 유럽식 방송의 데이터 방송 표준인 DVB-MHP[4]가 있다. 데이터 방송은 방송국에서 데이터 방송 애플리케이션을 방송 스트림에 실어 보내고, TV 셋톱박스에서 이를 실행함으로써 구현된다.

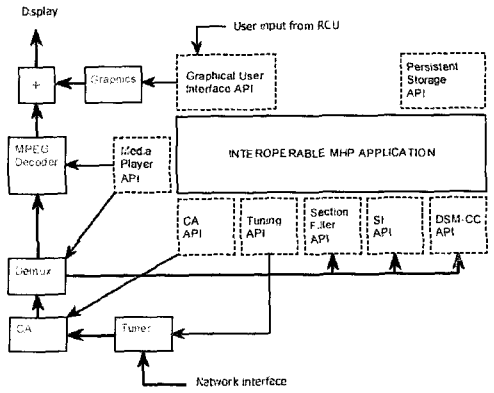
본 논문은 DVB-MHP, 특히 데이터 방송 애플리케이션을 구현하는데 사용되는 API에 대해서 기술한다.ACAP 과 OCAP은 MHP를 기반으로 하여 DVB 방송표준에 국한된 기능은 삭제하고 ATSC와 OpenCable 방송표준에서 꼭 필요로 하는 기능을 추가하여 만들었다. 따라서 ACAP과 OCAP는 핵심기능에서 MHP와 공유하는 부분이 매우 많다. 이것은 한 방송 플랫폼에서 수행되는 애플리케이션을 다른 방송 플랫폼에서 수행되도록 수정하기가 용이하다는 것을 의미한다. MHP 애플리케이션은 MHP-J와 MHP-HTML이 있는데, 본 논문에서는 전자만을 고려한다. MHP-J 애플리케이션은 Xlet이라고도 불리는데 Java 언어로 구현하므로 애플리케이션 프로그래머는 Java에 대한 지식이 있어야 한다. 뿐만 아니라, Xlet은 일반 프로그래머가 흔히 가지고 있지 않은 디지털 방송 규약에 대한 지식에 필요하므로, 제대로 구현하기가 쉽지 않다. 이에 본 논문은 MHP의 API 전체의 구조와 기능을 전체적으로 설명하여, 프로그래머들이 MHP API를 공부하고자 할 때 숲속에서 길을 잃지 않도록 도움을 주는 것을 목적으로 한다.

MHP와 MHP API는 Steven Morris가 구축한 웹사이트의 MHP 튜토리엘에 잘 설명되어 있다[10]. 본 논문에서는 프로그래머의 관점에서 MHP API를 조망하고자 한다. 특히 애플리케이션이 주로 시스템 내부와 외부에서 발생하는 이벤트를 처리하는 방식으로 구성된다는 것을 고려하여 API를

* 서강대학교 영상대학원 미디어공학과 교수
** 서강대학교 미디어공학과 재학

관련된 이벤트를 중심으로 분석하고자 한다. MHP API라는 것이 결국 이벤트들을 처리하기 위한 것이라고 볼 수 있으므로 이벤트를 알면 API 전체를 꿰뚫어 보는 효과가 있다.

2. 디지털/데이터 방송 개요, Java 이벤트 처리 방식, 애플리케이션



(그림 1) MHP 애플리케이션의 구동환경. 애플리케이션은 Network Interface를 통해 들어오는 전송스트림(TS)을 선택하고, 읽고, 처리하는 여러 가지 작동(action)을 할 수 있으며, 이런 작동은 주어진 표준 API(Java class들의 집합)들을 통해서 수행한다. 회색으로 표시된 박스는 하드웨어다.

2.1 애플리케이션 구동환경

(그림 1)은 TV 셋톱박스에서 애플리케이션이 구동되는 환경을 보여주고 있다. DVB에서는 방송의 기본단위를 서비스(Service)라고 한다. 하나의 서비스는 보통 하나의 채널에서 방송하는 방송물 전체를 가리키므로 채널과 서비스를 서로 교환하여 사용한다. 하나의 서비스는 보통 여러 개의 성분 스트림(Elementary Stream, 비디오/오디오/데이터 스트림), 그리고 애플리케이션과 이것에 사용되는 데이터스트림을 포함한다. 서비스들을 MPEG2 System[6]에서 정의된 MPEG-2 전송스트림(TS, Transport Stream) 안에 포함되어 전송된다. TV 수신기에는 다수의 TS가 도착하는데, 시청자가 특정 채널을 선택하면 그 채널이 포함된 TS가 선택되어 네트워크 인터페이스를 통해 TV 수신기로 들어온다(그림 1참조). 우리가 흔히 말하는 방송 프로그램(9시 뉴스, 한일전의 전반전)을 이벤트(Event)라고 한다. 하나의 이벤트는 시작시각과 끝나는 시각이 있다. 이벤트를 서비스를 시간대별로 구분한 것이므로 한 이벤트는 다수의 성분 스트림을 포함할 수 있다. "이벤트"는 "방송 프로그램"이라는 의미 외에 보다 일반적으로 "사건" 이라는 의미로 쓰인다. 디지털 방송 표준에서도 이벤트가 두 가지 의미로 사용되는데, 문맥에서 구분할 수 있다.

애플리케이션이 MHP API를 통해 Digital TV 수신기의 하드웨어를 제어하고 전송스트림(TS)으로부터 필요한 정보를 얻고자 할 때 주로 비동기적인 방식 또는 이벤트 드리븐 방식으로 처리한다[8,9]. 즉, 애플리케이션이 특정 정보를 요청하고, 그 정보가 도착하기까지 기다리지 않고 다른 일을 수행하고, 그 정보가 가용하다는 이벤트가 발생되면 이에 반응한다. 예를 들어 애플리케이션이 전송스트림으로부터 SI(Service Information), DSM CC Object, MPEG- Section의 Private Data 등을 MHP API를 통해 얻어 올 때도 이벤트를 통해 얻는다.

(그림 1)에 표시된 여러 가지 API들은 전송스트림안에 포함되어 전송되는 스트림과 데이터를 처리하는데 사용하기 위한 정의된 Java class들의 집합이다. 프로그래머가 애플리케이션을 구현하기 위해서는 이들 API를 사용해야 하므로, 이들에 대한 이해가 필수적이다. 더구나 PC와는 달리 애플리케이션에 오류가 생기면 이것은 바로 방송사고로 이어지므로 PC 프로그램보다 더 엄밀하게 프로그래밍해야 하므로, 이들 API에 대한 올바른 이해와 지식이 중요하다[11]. 본 논문은 이들 API를 전

체적으로 조망할 수 있도록 기술하는 것을 목적으로 한다.

2.2 객체 카루셀

DVB-MHP 표준에서는 서비스의 구성요소 중의 하나인 애플리케이션과 애플리케이션이 사용할 데이터를 전송하기 위해 DSMCC Object Carousel[7]을 기본으로 해서 정의된 DVB Object Carousel을 사용한다. 일반 컴퓨터 프로그래밍에서 프로그래머가 파일에 대한 이해 없이 응용 프로그램을 구현하는 것을 생각할 수 없듯이 TV 프로그래머가 파일 시스템에 해당하는 객체 카루셀(Object Carousel)에 대한 이해없이 TV 프로그래밍을 잘하기를 기대할 수 없다. 객체 카루셀은 객체의 전송을 위한 프로토콜인데 MHP 수신기 입장에서 보면 쓰기가 불가능하고 읽기만 가능한 파일 시스템이라고 볼 수 있다. 객체 카루셀은 데이터 카루셀(Data Carousel)로 인코딩되고 다시MPEG-2 섹션으로 인코딩된 후에 MPEG-2 TS로 패킷화 되어 전송된다. (그림 1)의DSMCC API는 객체카루셀에 포함된 객체들을 읽는 데 사용한다.

2.3 Java의 이벤트 처리 모델 : 소스객체, 이벤트 객체, 리스너 객체

Java 언어를 이용한 프로그래밍은 이벤트 드리븐 프로그래밍이다. 따라서 이벤트에 대한 이해는 Java 프로그래밍에서 필수적이며 MHP API의 전체적인 구조를 이해하는데도 필수적이다.이벤트란 무슨 일이 일어났다는 것을 알리는 일종의 신호다. 이벤트는 마우스의 클릭이나 움직임과 같은 외부의 사용자 액션에 의해 발생된다. 그리고 이벤트는 통신망을 통해 어떤 데이터가 도착하는 경우와 같이 시스템 내부의 액션에 의해서도 발생된다. Xlet의 경우에는 방송스트림으로부터 발생되는 다양한 종류의 이벤트가 있다. Xlet이 이들 이벤트를 처리하기 위해서는 Java의 이벤트 처리모델인 위임 이벤트 모델(Delegation Event Model)[8,9]을 따라 프로그래밍되어야 한다.

위임 이벤트 모델은 이벤트, 소스객체, 리스너 객체로 구성되어 있다. 시스템에서 처리할 수 있는 모든 이벤트 타입들은 java.util.EventObject의 서브 클래스들이다. 소스 객체는 관련 이벤트들에대해서 이것이 발생하는지를 감지하고, 이벤트가 발생하면 이를 입력인자로 하여 관련 이벤트 핸들러 메소드를 호출한다. 이벤트 핸들러 메소드들을 호출하려면 이들을 구현하고 있는 이벤트 리스너 객체를 생성해야 한다. 이벤트 리스너 클래스는 관련 이벤트 핸들러 메소드를 포함하고 있는 인터페이스를 "구현" 하고 있는 객체를 말한다. 소스 객체가 관련 리스너 객체를 알고 있도록 하기 위해 프로그래머는 관련 리스너 객체들을 소스 객체에게 미리 등록시킨다. 이를 위해 소스 객체는 리스너 객체를 등록시키는 데 사용되는 메소드를 가지고 있다.

예를 들어, Java에서 가장 흔한 소스객체인 버튼이나 스크롤바 같은 가시적인 컴포넌트 클래스(visual component class)를 생각해 보자. 컴포넌트 소스 클래스는 Focus나 Key 이벤트를 감지하여 관련 이벤트 리스너 객체 내의 이벤트 핸들러를 호출하도록 되어 있다. 소스객체인 컴포넌트 클래스가 관련 이벤트 리스너를 알고 있어야 하는데, 이를 위해 FocusListener, KeyListener 같은 관련 리스너 객체를 컴포넌트 클래스에 등록한다. 등록에 사용되는 메소드는 addFocusListener (FocusListener fl), addKeyListener (KeyListener kl) 이다. 예를 들어, KeyListener 객체는 세 개의 키 이벤트핸들러 메소드 즉, keyPressed (KeyEvent e), KeyReleased(KeyEvent e), KeyTyped(KeyEvent e)를 구현한다. 소스 객체인 컴포넌트 객체는 실제로 발생하는 키이벤트의 종류에 따라 세 개 중의 한 핸들러 메소드를 호출한다.

본 글에서 MHP API를 분석할 때 이벤트 클래스를 중심으로, 관련 소스 클래스와 리스너 인터페이스를 나열하면서 설명한다.

2.4 애플리케이션 (Xlet)

DVB-J 애플리케이션은 DVB-MHP 수신기에 실행되는 자바 애플리케이션으로 인터넷의 자바 애플릿 (Applet)과 유사하여 Xlet이라 불린다[4]. Xlet은 비디오와 연동되어 실행되거나 비디오와 무관하게 독립적으로 실행될 수 있다. Xlet은 객체 카루셀에 포함되어 TS를 통해 전송된다. 객체 카루셀은 특정 서비스의 한 요소로서 존재하므로, Xlet도 특정 서비스 안에 포함된다. 따라서 사용자가 Xlet을 선택하려면 그것이 포함되어있는 서비스를 선택해야 한다. 이것은 보통 셋톱박스에 내장되어 있는 EPG를 통해서한다. Xlet이 포함된 서비스가 선택되면 Xlet 중에서 "AutoStart" 라는 특성을 가진 것은 애플리케이션 매니저에 의해서 자동으로 로드된다. AutoStart로 명시되지 않은 Xlet은 사용자가 직접 그 실행을 요구해야 된다. 이 요청은 애플리케이션 매니저에게 전달되어 처리된다. 사용자가 새로운 서비스를 선택하게 되면, 현재 서비스 안에 포함되어 있던 Xlet는 소멸된다.

3. MHP API의 구조

MHP API는 기본적인 자바 API에 Sun JavaTV API, DAVIC API, HAVI API, DVB API 등의 다른 API를 추가한 형태로 구성되어 있다[4]. API들은 여러 가지 기준에 의해 분류될 수 있다. 본 논문에서는 (그림 1)에서 보듯이 MHP API는 API가 처리하는 시스템 컴포넌트 또는 데이터의 종류에 따라 다음과 같이 분류한다.

1. Section Filter API
2. SI API
3. DSM-CC API
4. Tuning API
5. CA API
6. Media Player API
7. Graphical User Interface
8. Persistent Storage API

3.1 Section Filter API

MPEG-2 Section은 오디오, 비데오를 제외한 다양한 데이터를 엔코딩하여 전송할 수 있는 전송메카니즘이다. Section Filter API는 MPEG-2 Section으로 전송되는 데이터를 필터링하기 위해 제공되는 API이다. Section Filter API는 주로 org.davic.mpeg.sections 패키지에 포함되어 있다. 애플리케이션이 필터링하고자 하는 Section이 방송 스트림으로 언제 TV 수신기에 도착될지 모르기 때문에 이를 그냥 기다릴 수는 없다. 따라서 Section Filtering은 비동기적인 이벤트 드리븐 방식으로 처리된다.

이벤트를 이해하려면 이벤트를 정의하는 클래스, 그리고 이 이벤트를 발생시키는 객체를 정의하는 소스 클래스, 그리고 이벤트를 처리하는 핸들러를 정의하는 리스너 인터페이스를 이해해야 한다. Section Filter API에서 정의하는 소스클래스, 이벤트 클래스, 리스너 인터페이스는 〈표 1〉과 같다.

〈표 1〉 Section Filter API

Source Class	Event Class	Listener Interface
SectionFilter	SectionFilterEvent	SectionFilterListener
	SectionAvailableEvent	
	VersionChangeDetectedEvent	
	EndOfFilteringEvent	
	TimeOutEvent	
	IncompleteFilteringEvent	
SectionFilterGroup	ForcedDisconnectedEvent	(org.davic.resources 패키지) ResourceStatusListener
	FilterResourcesAvailableEvent	

SectionFilter 소스 클래스에서 발생시키는 이벤트들은 전송스트림의 Section Filtering이 성공적으로 수행되어 Section의 데이터를 이용할 수 있을 때 발생하는 이벤트와 요청한 Section Filtering이 실패했을 때 발생하는 이벤트를 정의하고 있다. 한편 Section Filtering할 때 셋탑 박스의 하드웨어의 자원이 충분히 지원되지 못하는 경우에 발생하는 이벤트는 ResourceStatusListener 인터페이스에 의해 처리된다.

3.2 SI API

〈표 2〉 SI API 패키지 리스트

SI API 패키지	비 고
javax.tv.service	SI database에 접근하는 기본적인 포인트를 제공하고, Service, SIElement interface와 같은 다른 SI package에서 사용하고 있는 공통 클래스들을 포함하고 있다.
javax.tv.service.guide	스케줄, 개별 프로그램 이벤트, 프로그램 등급 등, EPG의 기능을 지원하기 위한 API를 제공한다.
javax.tv.service.navigation	DVB에서 Service, ATSC에서Vitual Channel이고 불리우는 서비스들을 탐색하는데 사용되는 클래스들을 제공한다.
javax.tv.service.transport	SI Data에 의해 설명되고 있는 현재 Transport Stream의 전송 메카니즘에 관한 추가적인 정보를 제공한다.
javax.tv.service.selection	Service를 선택하는 메커니즘을 제공한다.
org.dvb.si	DVB service information에 대한 접근을 제공한다.
org.dvb.application	Service내 애플리케이션의 리스트에 접근하고 애플리케이션에 대한 정보를 제공한다

전송스트림(TS)은 다수의 서비스로 구성되어 있다. 하나의 서비스는 복수의 성분 스트림(Elementary Stream)으로 구성된다. 따라서 TV 수신기에서 다중화된 전송스트림에서 특정 서비스를 선택하고 선택된 Service의 각 성분 스트림을 찾을 필요가 있다. 이를 위해 필요한 정보는 MPEG-2 규약의 PSI 테이블과 DVB에서 정의한 SI (Service Information) 테이블을 통해 제공된다. SI Table은 현재 방송되고 있는 TS의 구조를 제공할 뿐만 아니라 Service들의 스케쥴 정보까지 제공함으로써 사용자에게 모든 Service의 유익한 정보를 제공한다. SI API는 방송 스트림에 전송되어 오는 SI Table에 대한 접근을 제공한다. 〈표 2〉는 SI API를 구성하고 있는 API 패키지들이다.

SI API를 이용하여 실제 SI Data를 얻어오는 과정은 MPEG-2 Section Filtering에서와 마찬가지로 비동기적인 방식으로 처리된다. 애플리케이션이 요구하는 SI Data가 방송 스트림으로 언제 Digital TV 수신기에 도착될지 모르기 때문이다.

〈표 3〉 SI API

Source Class	Event Class	Listener Interface
	SIChangeEvent	SIChangeListener
ProgramSchedule	ProgramScheduleEvent	ProgramScheduleListener
ServiceDetails	ServiceDetailsSIChangeEvent	ServiceComponent ChangeListener
	ServiceComponentChangeListener	
BouquetCollection	BouquetChangeEvent	BouquetChangeListener
NetworkCollection	NetworkChangeEvent	NetworkChangeListener
Transport	ServiceDetailsChangeEvent	ServiceDetailsChangeListener
TransportStream Collection	TransportStreamChangeEvent	TransportStream ChangeListener
ServiceContext	ServiceContextEvent	ServiceContextListener
	PresentationTerminatedEvent	
	ServiceContextDestroyedEvent	
	NormalContentEvent	
	PresentationChangedEvent	
	AlternativeContentEvent	
	SelectionFailedEvent	
AppsDatabase	AppsDatabaseEvent	AppsDatabaseEventListener
AppProxy	AppStateChangeEvent	AppStateChangeEventListener
SIDatabase	SIRetrievalEvent	SIRetrievalListener
	SISuccessfulRetrieveEvent	
	SITableUpdatedEvent	
	SINotInCacheEvent	
	SIObjectNotInTableEvent	
	SIRequestCancelledEvent	
	SITableNotFoundEvent	
	SILackOfResourcesEvent	
	SIMonitoringEvent	SIMonitoringListener

SI API 라고 하면 협의의 의미로 DBV SI 정보에 접근하는데 필요한 org.dvb.si 패키지를 가리키는 경우가 있다. 그러나 본 절에서 SI 라고 한 것은

Service Information을 일반적인 의미로 사용했다. DVB 에 국한된 SI 정보는DVB SI 라고 구체적으로 기술한다.

SI 관련 이벤트 클래스, 소스 클래스, 리스너 인터페이스 클래스는 〈표 3〉과 같다.

SI는 Table 포맷으로 포장되어 MPEG-2 Section에 담겨 전송이 된다. SI Data를 담고 있는 Table들은 MPEG-2 규약에서 정의한 PAT, PMT, NIT, CAT의 PSI(Program Specific Information)와 이것 이외에 DVB에서 추가적으로 정의한 SDT, EIT, BAT, TDT, TOT 등이 있다. SI Table 이벤트들은 Table에 담겨 있는 SI Data를 애플케이션에 비동기적으로 제공하고 SI Data가 추후에 변동, 갱신되었을 경우에도 애플리케이션에게 이를 비동기적으로 통보하는 매카니즘을 제공한다.

3.3 DSM-CC API

DVB-MHP는 broadcast file system protocol로써 DSM-CC Object Carousel을 사용한다. Object Carousel로 전송이 되는 데이터들은 그 데이터 종류에 따라 Object Carousel에서 정의한 BIOP Message들로 캡슐화 되어 전송된다. BIOP Message는 Object Carousel에서 Object라고 표현되기도 하는데 그 종류는 Directory, Service Gateway, File, Stream, Stream Event 등 5 종류가 정의되어 있다. DSM-CC API는 애플리케이션이 이런 Object들을 접근하도록 지원하는 API이다. DSM-CC API도 앞선 MPEG-2 Section Filtering API와 SI API에서와 마찬가지로 애플리케이션의 이런 Object들에 대한 접근을 비동기적으로 처리를 한다. MHP API에서 DSM-CC API에 속하는 패키지들은 javax.tv.carousel 패키지와 org.dvb.dsmcc 패키지가 있다. 〈표 4〉는 DSM-CC API에서 제공되는 이벤트 클래스, 소스 클래스, 리스너 인터페이스를 정리한 표다.

〈표 4〉 DSM-CC API

Source Class	Event Class	Listener Interface
CarouselFile	CarouselFileChangeEvent	CarouselFileListener
DSMCCObject	AsynchronousLoadingEvent	AsynchronousLoadingEventListener
	SuccessEvent	
	InvalidFormatEvent	
	InvalidPathnameEvent	
	LoadingAbortedEvent	
	MPEGDeliveryErrorEvent	
	ServiceXFRErrorEvent	
	NotEntitledEvent	
DSMCCStream	NPTPresentEvent	NPTListener
	NPTStatusEvent	
	NPTDiscontinuityEvent	
	NPTRemovedEvent	
	NPTRateChangeEvent	
DSMCCStreamEvent	StreamEvent	StreamEventListener
DSMCCObject	ObjectChangeEvent	ObjectChangeEventListener

DSM-CC API의 DSM-CC Section 이벤트들은 Object Carousel로 전송되는 Object들을 비동기적으로 Loading하는 작업과 관련있는 이벤트들이다. 또한 Object가 갱신되어 변화가 생겼을 경우에 발생되는 이벤트도 포함된다. 이외 애플리케이션이 A/V Stream과 동기화를 이루기 위해 쓰이는 StreamEvent와 NPT 관련 이벤트들도 DSM-CC Section 이벤트에 포함시켰다.

3.4 Tuning API

〈표 5〉 Tuning API

Source Class	Event Class	Listener Interface
NetworkInterface	NetworkInterfaceEvent	NetworkInterface Listener
	NetworkInterfaceTuningEvent	
	NetworkInterfaceTuningOverEvent	
NetworkInterface Manager	NetworkInterfaceReleasedEvent	org.davic.resources ResourceStatus Listener
	NetworkInterfaceReversedEvent	

각 전송스트림(TS)은 서로 다른 주파수 대역으로 전송이 된다. 특정 주파수 대역으로 전송되는

TS 를 선택하고자 할 때 (그림 1)에서 보듯이, 수신기가 Tuning API 를 통해 tuner를 제어한다. MHP API에서 Tuning API에 속하는 패키지는 org.davic.net.tuning 패키지가 있다. 〈표 5〉는 Tuning API에서 제공하는 이벤트 클래스, 소스클래스, 리스너 인터페이스를 기술하고 있다.

3.5 CA API

〈표 6〉 CA API

Source Class	Event Class	Listener Interface
CAModuleManager	CAEvent	CAListener
	MMIActiveEvent	
	MMIInactiveEvent	
	ModuleRemovedEvent	
	NewModuleEvent	
	PIDChangeEvent	
	TuneREquestEvent	
DescramblerProxy	DescramblerEvent	DescramblerListener
CAModule	MessageEvent	MessageListener
	ModuleResponseEvent	
	ModuleStateChangeEvent	
	SessionClosedEvent	
	SessionOpenedEvent	
CAModuleManager	MMIEvent	MMIListener
	CloseMMIEvent	
	StartMMIEvent	
CAModuleManager	DescramblingStoppedEvent	org.davic.resources ResourceStatusListener
	DescramblingStartedEvent	

디지털 TV 방송에서 유료 방송 서비스를 위해 방송 송출 당시 신호를 암호화하여 전송하게 된다. 따라서 암호화된 신호를 해독할 수 있는 물리적인 시스템이 없다면 방송 시청이 전혀 불가능 하다. 이처럼 방송의 유료 수신 기능을 담당하는 시스템을 CAS(Conditional Access System)라 한다. 스마트카드가 수신기에 삽입되어 CAS를 구동시켜 수신기에 들어오는 암호화된 신호를 복호화시키는 일을 한다. CAS는 허락 받은 가입자만 암호화된 방송신호를 시청할 수 있도록 한다. CAS는 무료, 기본, 유료 채널은 물론 PPV(Pay Per View)까지 동시에 지원하며 스마트 카드에 의한 다양한 서비스를 구현하게 된다. CA API는 애플리케이션이 셋톱박스의 유료수신기능을 담당하는 장치인 CAS에 접근할 수 있도록 설계되었다. MHP API에서 CA API에 속하는 패키지는 org.dvb.net.ca 패키지와 org.davic.net.ca 패키지가 있다. 〈표 6〉은 CA API에서 제공하는 이벤트 클래스, 소스 클래스, 리스너 인터페이스를 기술하고 있다.

CA API를 구현하려면 에뮬레이터에 암호화된 TS를 해독할 수 있는 물리적인 CAS가 있어야만 한다. TS를 암호화하는 CAS 제조회사는 암호화된 TS를 복호화하는 알고리즘을 공개하지 않는다. 그것은 TS를 암호화하는 목적이 유료 방송 서비스를 하기 위한 것인데, 복호화 알고리즘이 공개되면 허가 받지 않은 가입자의 불법적인 시청을 막을 수 없기 때문이다.

3.6 Media Player API

〈표 7〉 Media Player API

Source Class	Event Class	Listener Interface
VideoFormatControl	VideoFormatEvent	VideoFormatListener
	ActiveFormatDescriptionChangedEvent	
	AspectRatioChangedEvent	
	DFCChangedEvent	
javax.media.Player	CAStopEvent	javax.media. ControllerListener
	NoComponentSelectedEvent	
	ServiceRemovedEvent	
	StopByResourceLossEvent	
	PresentationChangedEvent	
SubtitlingEvent Control	SubtitleAvailableEvent	SubtitleListener
	SubtitleNotAvailableEvent	
	SubtitleNotSelectedEvent	
	SubtitleSelectedEvent	
javax.media.Player	MediaPresentedEvent	javax.media. ControllerListener
	MediaTimePositionChangedEvent	
	ResourceReturnedEvent	
	ResourceWithdrawnEvent	
MediaSelectControl	MediaSelectEvent	MediaSelectListener
	MediaSelectFailedEvent	
	MediaSelectCARefusedEvent	
	MediaSelectSucceededEvent	

Xlet 애플리케이션은 전송스트림에 포함되어 전송되는 스트림을 제어하는데 사용하기 위해 사용

하는 미디어 컨트롤 패키지로는 org.dvb.media, org.davic.media, javax.tv.media 패키지가 있다. 〈표 7〉은 Media Player API에서 제공하는 이벤트 클래스, 소스클래스, 리스너 인터페이스를 기술하고 있다.

3.7 Graphical User Interface

애플리케이션의 GUI를 위해 제공된 API는 MHP 규약에서 AWT와 HAVI API를 정의하고 있다. HAVI는 필립스와 일본의 샤프 등에서 소형 가전에 필요한 API를 정의한 집합들이다. HAVI API 중 사용자 인터페이스 부분만 채용하여 MHP API에 포함시켰다. GUI와 관련된 MHP API 패키지는 org.havi.ui, org.havi.ui.event, org.dvb.event 패키지가 있다. 〈표 8〉은 이들 패키지에서 제공하고 있는 이벤트 클래스, 소스 클래스, 리스너 인터페이스를 기술하고 있다.

〈표 8〉 GUI API

Source Class	Event Class	Listener Interface
org.havi.ui.HTextButton	HActionEvent	HActionListener
org.havi.ui.HAdjustment Value	HAdjustmentEvent	HAdjustmentListener
org.havi.ui.HBackground Image	HBackgroundImageEvent	HBackgroundImage Listener
org.havi.ui.HComponent	HFocusEvent	HFocusListener
org.havi.ui.HItemValue	HItemEvent	HItemListener
org.havi.ui.HTextValue	HKeyEvent	HKeyListener
	HRcEvent	
org.havi.ui.HScreen Device	HScreenConfigurationEvent	HScreenConfiguration Listener
org.havi.ui.HScreen Device	HScreenDeviceReleasedEvent	org.davic.resources. ResourceStatusListener
	HScreenDeviceReservedEvent	
org.havi.ui.HVideo Component	HScreenLocationModifiedEvent	HScreenLocation ModifiedListener
org.havi.ui.Event Manager	UserEvent	UserEventListener
	UserEventAvailableEvent	org.davic.resources. ResourceStatusListener
	UserEventUnavailableEvent	

MHP의 GUI API에 HAVI API와 AWT가 대부분을 차지하고 있다. HAVI API의 이벤트는 대부분 AWT 이벤트를 상속받고 있다. 즉, HAVI API의 이벤트는 AWT 이벤트 기능과 상당수 겹친다.

org.dvb.event 패키지에 정의된 이벤트는 AWT 이벤트 처리 방식과는 달리 User Input Focus가 없어도 사용자의 Input Event를 처리할 수 있는 메커니즘을 제공하는 이벤트다. AWT 이벤트는 특정 component가 User Input Focus를 가지고 있어야만 그 component에서 이벤트가 발생하여 등록된 이벤트 리스너가 이벤트를 처리한다. 그러나 UserEvent는 User Input Focus를 가지고 있지 않은 사용자의Input Event도 처리할 수 있는 메커니즘을 제공한다. 현재 MHP에서 정의된 사용자의 Input Event는 KeyEvent만 정의하고 있는데 AWT의 KeyEvent와 기능이 동일하다.

3.8 Persistent Storage API

셋탑의 persistent storage(하드 디스크)에 상주하고 있는 파일을 읽고, 또 거기에 파일을 쓰는데 사용되는 API로서 java.io package를 확장한 것이다. 또한 Persistent Storage API에 대한 다른 패키지는 org.dvb.io.persistent가 있다.

4. 결 론

본 글에서는 유럽식 디지털 방송 표준에서 사용하는 데이터 방송 표준인 MHP의 API에 대해서 분석하였다. API 전체를 분류하고각 API 그룹에서 정의한 이벤트 클래스와 이와 관련된 소스 클래스와 리스너 인터페이스를 연결시켜 기술하였다. 이벤트를 중심으로 MHP API를 분석한 것은 방대한 API를 심도있게 이해하기 위한 한 방법이라고 생각된다.

참고문헌

[1] ATSC Standard A/100: DTV Application Software Environment - Level 1 (DASE-1), 09 March 2003, http://www.atsc.org

[2] ATSC Candidate Standard CS/101A: Advanced Common Application Platform (ACAP) , 19 February 2004, http://www.atsc.org

[3] OpenCable Application Platform Specification (OCAP) 1.0 - I11 OCAP, 2003. http://www.opencable.com

[4] ETSI TS 101 812 v.1.21. (330-06). Digital Video Broadcasting(DVB): Multimedia Home Platform (MHP) Specification 1.0.2.

[5] Richard Chernock. "Data Broadcasting", McGrawhill, 2001.

[6] ISO/IEC 13818-1 Generic Coding of Moving Picture and Associated Audio : Systems

[7] ISO/IEC 13818-6 Generic Coding of Moving Picture and Associated Audio: Digital Storage Media Command and Control

[8] Sun Microsystems, Java AWT: Delegation Event Model. http://java.sun.com/j2se/1.3/docs/guide/awt/designspec/events.html, 2002.

[9] Y.D. Liang, "Introduction to Java Programming", Que E&T 1999

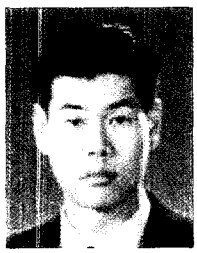
[10] Steven Morris. The Interactive TV Web. http://www.mhp- interactive.org/tutorial 2002.

[11] 정문열, 백두원. 연동형 데이터 방송 애플리케이션의 구조. 2003 년 3월. 한국방송공학회 논문지 9권 1호., 74-82.

저자약력

정 문 열

1980년 서울대학교 자연대학, 공학사
1982년 한국과학기술원, 전산학과 석사
1992년 Univ. of Pennsylvania, 전산학과 Ph.D.
2000년 현재 서강대학교 영상대학원 미디어공학과 교수
연구분야 : 컴퓨터 그래픽스, 캐릭터 애니메이션, 디지털방송



류 일 권

2000년 서강대학교 화학과 학사
2002년 현재 서강대학교 미디어공학과 재학
전공 : CG/애니메이션1. 정문열