

논문-02-07-3-08

DASE 데이터 방송 처리가 가능한 DTV 셋톱 박스의 구현

박 용 규*, 장 규 환**

Implementation of DASE Middleware for the Data-ready DTV Settop Box

YongGyu Park* and KyuHwan Chang**

요 약

최근 디지털 TV를 이용한 데이터 방송의 관심과 중요도가 날로 증대되고 있다. 본 논문에서는 데이터 방송을 처리할 수 있는 DTV 셋톱박스를 구현하는데 필요한 미들웨어의 코어 부분을 다루며, 이를 실제 셋톱에 탑재하여 동작 테스트를 완료하기까지의 과정을 기술하였다.

Abstract

In recent years, data broadcasting has been drawing much attention with glowing importance. In this paper, we implement a DASE middleware which is crucial for the data-ready DTV settop box. The software has been ported to a prototype system and tested with real time broadcasting streams.

I. 서 론

향후 TV의 추세는 HD급의 고화질과 AV이외에 다양한 부가 정보를 받아볼 수 있는 데이터 방송이 큰 주류를 이룰 것이다. 데이터 방송에는 ATVEF, DASE, MHP등의 여러 규격들이 있다. DASE(DTV Application Software Environment)^[1]는 디지털 TV 상에서 다양한 애플리케이션을 지원하기 위해 정의한 국제 규격이다. DASE 셋톱 박스는 XXML 응용 프로그램을 처리하기 위한 DAE(Declarative Application Environment)^[2]와 Xlet 응용프로그램을 처리할 수 있는 PAE(Procedural Application Environment)^[3]를 지원해야 한다. 본 논문에서는 DASE 규격을 만족하는 셋톱 박스의 구성 요소와 주요 모듈의 구현 방법을 기술한다.

본 논문의 2절에서는 시스템의 하드웨어 및 소프트웨어 구성 요소를 소개한다. 3절에서는 DASE 지원의 근간이 되는 데이터 방송^[4] 및 PSIP(Program and System Information Protocol)^[5] 관리 모듈을 기술하고, 4절과 5절에서는 DAE, PAE를 구성하는 주요 모듈을 기술한다. 6절은 구현한 실제 하드웨어와 여기에서 동작하는 화면을 도시하였다. 7절에서는 본논문의 주요 결론을 정리하여 기술하였다.

II. 시스템 개요

1. DASE 규격

DASE 시스템은 그림 1에서처럼 DAE, PAE, Common Content Decoders, Security Framework으로 이루어져 있다. 각 구성 요소가 만족시켜야 할 기능은 다음과 같다.

Declarative Application Environment는 XXML (eXtensible

* (주) 마르시스 대표이사
C.E.O MARUSYS Co.,Ltd

** (주) 대우전자 대표이사
C.E.O DAEWOO Elec. Co.,Ltd

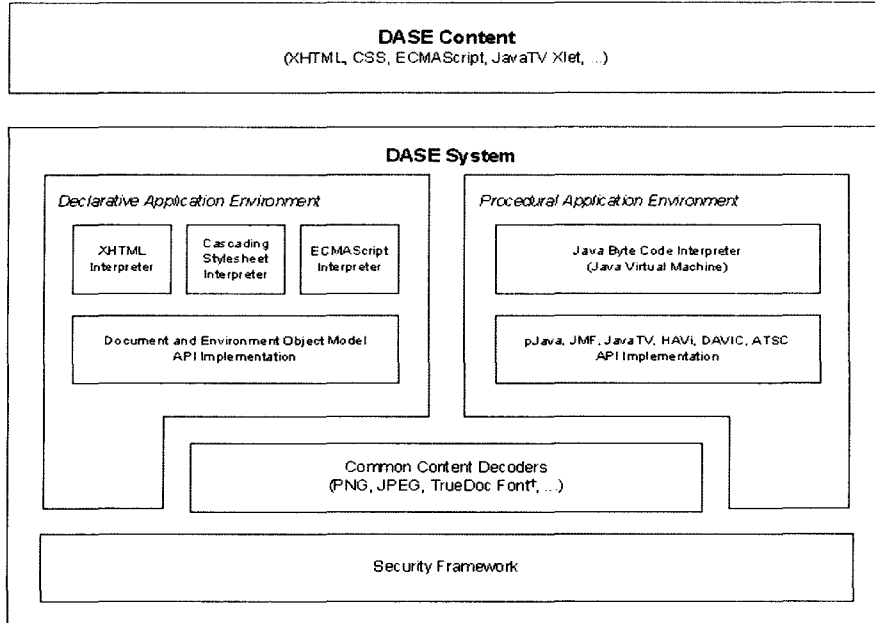


그림 1. DASE의 구조
Fig. 1. DASE structure

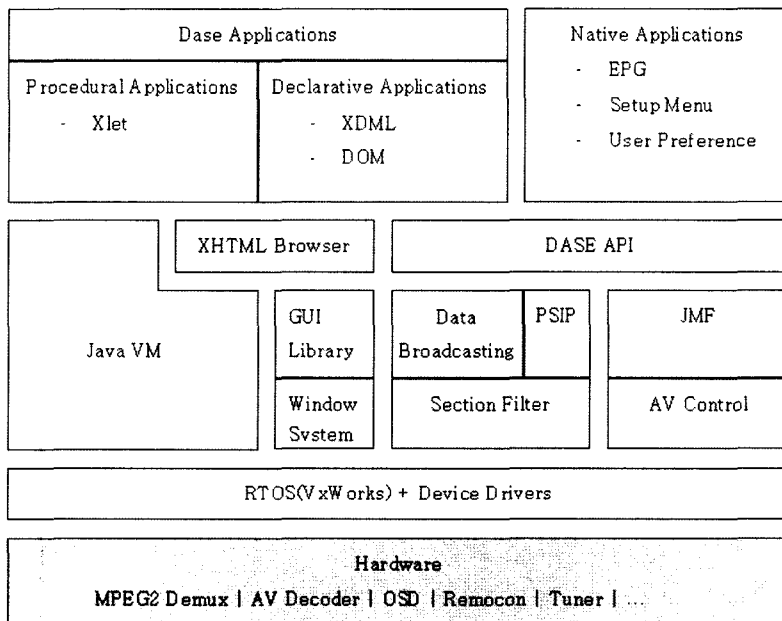


그림 2. 시스템 구성
Fig. 2. System configuration

DTV Markup Language) 콘텐츠를 화면에 표시할 수 있어야 하며, ECMA 스크립트, DOM, CSS를 지원해야 한다. Procedural Application Environment는 Xlet 애플리케이션을

실행할 수 있어야 하며, 이를 위해 Personal Java, JMF, JavaTV, HAVI, DAVIC 등의 Java API를 지원해야 한다. Common Content Decoders는 DASE 셋톱 박스는 image/

jpeg, image/png, video/mng, video/mpeg, video/mpv, audio/ac3, audio/basic 데이터를 처리할 수 있어야 한다.

2. 시스템 구성

DASE 셋톱 박스는 그림 2에서와 같이 DTV 수신을 위한 하드웨어, DASE 지원을 위한 소프트웨어, 애플리케이션의 세 부분으로 이루어져 있다.

2.1. 하드웨어

하드웨어는 크게 CPU, TL850, I/O 모듈로 나누어 진다. CPU는 QED R5231A를 사용하였으며 350MHz로 동작한다. TL850은 MPEG-2 비디오 디코더, 비디오 프로세서, 2D 그래픽 가속기를 포함하고 있으며 튜너로부터 수신한 트랜스포트 스트림을 디코딩하여 AV로 출력하는 기능을 담당한다.

2.2. 소프트웨어

소프트웨어 모듈은 데이터 방송 및 PSIP 데이터를 관리하는 모듈, Java TV를 비롯한 DASE API를 구현하기 위한 모듈, DASE 애플리케이션을 수행하기 위한 JVM과 브라우저로 구성된다. 데이터 방송 및 PSIP 모듈은 MPEG2 트랜스포트 스트림으로부터 해당 테이블을 추출하고 이들 데이터를 DASE 구현 클래스가 사용할 수 있는 형태로 변환한다. DASE API는 AV 모듈과 데이터 모듈의 기능을 이용하여 JavaTV, JMF, HAVi 인터페이스를 구현한다. JVM은 Xlet 애플리케이션을 구동하기 위해서 필요하다. 본 논문에서는 DASE 규격에 따라 Personal Java를 사용하였다. XDMML 브라우저는 Declarative Application을 지원하기 위해 Ice 브라우저를 커스터마이징하여 사용하였다.

III. 데이터 방송 및 PSIP 모듈

ATSC PSIP 표준에서는 트랜스포트 스트림 내의 서비스 및 이벤트에 대한 정보를 담고 있는 MGT(Master Guide Table), STT(System Time Table), VCT(Virtual Channel Table), EIT(Event Information Table), ETT(Extended Text Table)을 정의하고 있다. Data Carousel, MPE(Multiprotocol Encapsulation) 등의 방식으로 데이터를 전송하기 위해서는 이들 테이블 외에 DST(Data Service Table),

DET(Data Event Table)를 처리할 수 있어야 한다.

데이터 방송 및 PSIP 모듈은 트랜스포트 스트림으로부터 해당 테이블을 구성하는 섹션을 추출하고, 이를 DB로 구축하여 관리하게 된다. 그림 3은 트랜스포트 스트림으로부터 DB를 관리하는 PSIPManager, DatabroadcastManager 사이의 데이터 흐름을 보여 준다.

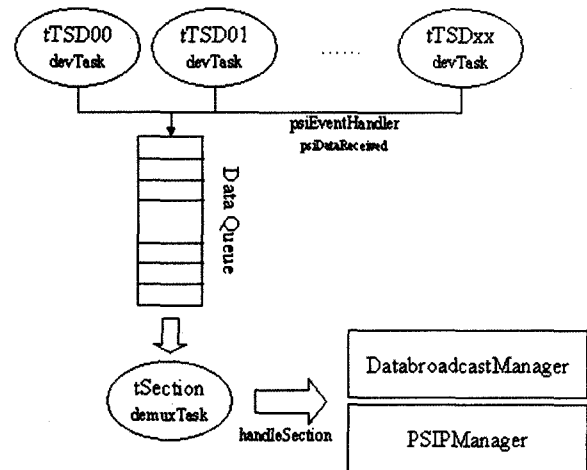


그림 3. 데이터 흐름
Fig. 3. Data flow

1. PSIP Manager

PSIP Manager는 PSIP 표준에서 정의하고 있는 MGT, STT, RRT, VCT, EIT, DET, ETT를 관리한다. PSIP 정보를 전달하는 대부분의 테이블들은 PID와 버전을 MGT에 기록하도록 되어있다. 따라서 수신부에서는 MGT만 관찰하고 있다가 새로운 버전의 MGT가 도착하면 변경된 내용을 반영한다. VCT, RRT같은 경우는 PID가 미리 알려졌기 때문에 버전만을 관찰한다. 즉, 새로운 MGT를 분석한 결과 VCT의 버전이 바뀌면 이를 반영하는 식이다. EIT, DET 등은 PID를 알려면 반드시 MGT를 참고해야 한다.

2. DatabroadcastManager

DatabroadcastManager의 역할은 데이터 방송의 핵심 역할을 하는 DST를 수신하고, DST의 내용에 따라 Data Carousel 및 MPE로 전송되는 데이터를 해당 DASE 애플리케이션에 전달하는 것이다. 다음 그림은 DatabroadcastManager의 멤버인 DSTManager의 구조를 보여 준다.

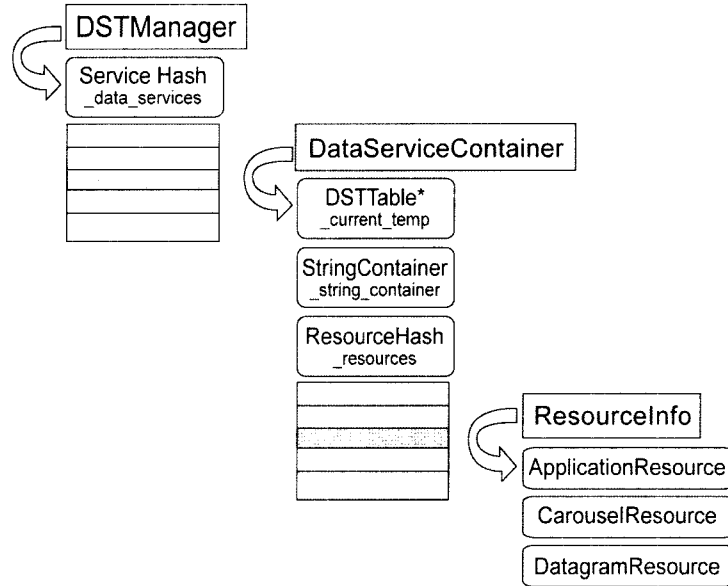


그림 4. DST 관리자
Fig. 4. DST manager

표 1. DataCarousel 관리자
Table 1. Data carousel manager

클래스	기능
DataCarouselManager	최상위 클래스, 여러 개의 다운로드 시나리오를 관리한다. 각 다운로드 시나리오를 Download Id/PID로 찾을 수 있도록 둘 사이의 관계를 해시 테이블로 관리한다.
DownloadScenario	하나의 PID에 대응하는 다운로드를 가리킨다. 여러 개의 그룹을 해시 테이블로 관리한다. DSI 섹션이 여기에 저장된다.
GroupManager	자기 그룹 속에 속한 모듈들을 해시 테이블로 관리한다. module id로 각 모듈을 찾을 수 있다. DII 섹션을 저장한다.
ModuleManager	모듈은 여러 개의 블록으로 구성되는데, 모듈이 갱신되는 경우 기존 버전에 대한 요구를 만족시키면서 새로운 버전을 만들어야 한다. 이를 고려하여 current, temp 버전의 모듈 컨테이너를 관리한다.
ModuleContainer	section/table의 관계와 같다. 모듈을 구성하는 블록은 하나의 DDB 섹션에 들어있다. 몇 개의 섹션이 모여 하나의 모듈을 구성하는데 DDB 섹션을 모으는 것이 이 클래스의 일이다.

DST는 채널당 하나씩 있을 수 있으므로 여러 개의 DST를 동시에 관리하기 위해 채널 번호를 이용한 해시 테이블을 사용한다. DSTManager의 해시 테이블에 저장되는 데이터 타입은 DataServiceContainer 클래스이다. 여기서 중요한 것은 ResourceHash인데 DST에 있는 모듈 정보를 locator를 이용하여 검색할 수 있도록 해준다. ResourceHash에 들어갈 수 있는 데이터는 ResourceInfo 클래스 타입을 갖는데 이것은 pure virtual class이다. 따라서 실제로는 이를 계승 받은 ApplicationResource, CarouselResource, DatagramResource 클래스가 된다. 이는 각각 애플리케이션, Carousel File, Datagram 정보를 들고 있다.

Data Carousel로 전송되는 데이터는 DataCarouselManager

를 통해 관리된다. 표 1.은 Data Carousel을 지원하기 위해 사용되는 클래스를 정리한 것이다.

IV. Declarative Application Environment

Declarative Application을 지원하기 위해서는 XHTML을 지원하는 브라우저가 있어야 한다. 본 논문의 DAE는 IceSoft사의 IceStorm Browser v5.01을 사용하여 구현하였다. IceStorm Browser는 자바 기반의 웹 브라우저로 HTML 4.01, DOM 2.0, CSS 2.0등의 W3C 표준을 지원한다. 본 절에서는 브라우저의 기본 기능 이외에 부가로 추가된 기능을 기술한다.

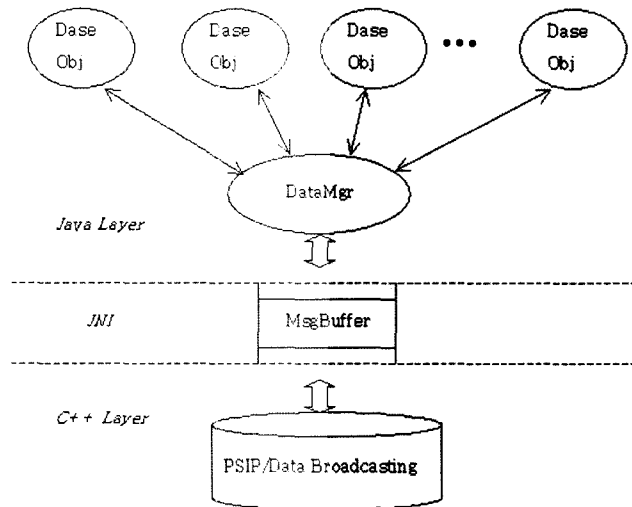


그림 5. DataMgr 클래스
Fig. 5. DataMgr class

1. TV Scheme 지원

TV Scheme은 URI를 통해 방송 중인 MPEG 스트림을 표현하는 방법이다. DA에서 백그라운드나 스크린의 일부 영역에 MPEG 비디오 화면을 디스플레이하기 위해 TV scheme을 사용한 URI를 사용할 수 있다. XDMML에서 TV Scheme을 사용할 수 있는 부분은 다음과 같다. CSS에서 background-image property, <BODY>, <TABLE>등의 HTML tag에서 background attribute, <OBJECT> tag에서 data attribute등이 있다.

2. Hybrid DA

DASE DAE에서는 DA내에 JavaTV Xlet 형태의 애플리케이션을 사용할 수 있도록 Active Content Object Element를 정의하고 있다. <OBJECT> Element의 content-type이 application/javatv-xlet인 경우가 이에 해당한다. 본 구현에서는 이러한 Hybrid DA를 구현하기 위해 IceStrom 브라우저에서 제공하는 Pilot 인터페이스를 확장하였다.

V. Procedural Application Environment

DASE PAE를 구현하기 위해서는 데이터 방송 및 PSIP 모듈에서 수신한 데이터를 해당 DASE API를 통해 접근할

수 있어야 한다. 본 절에서는 DASE API를 구현하는 데 핵심적인 역할을 하는 클래스들을 기술한다.

1. DataMgr 클래스

PSIP/Data Broadcasting에 대한 모든 접근은 그림 5에서와 같이 DataMgr 클래스를 통해서 이루어진다.

DataMgr 클래스의 기능은 모든 데이터 접근을 serialize 함으로써 여러 개의 쓰레드가 동시에 PSIP/Data Broadcasting 데이터에 접근하는 경우 데이터의 일관성을 보장한다. Java로 구현된 DASE API의 query를 JNI로 구현된 MsgBuffer를 통해 C++ query로 변환한다. 마찬가지로 C++ 데이터로 저장되어 있는 PSIP/Data Broadcasting 데이터를 Java 타입으로 변환한다. 또한 데이터 접근을 일원화함으로써, 향후 시스템 확장이나 수정이 있을시, 코드의 수정을 용이하게 한다.

2. Xlet 관련 클래스

Xlet 애플리케이션을 처리하기 위해서는 다음과 같은 클래스를 사용한다.

XletMgr는 Xlet의 라이프 싸이클, 즉, loading, pause/resume, destroy 인터페이스를 제공한다. XletLoader는 Xlet은 트랜스포트 스트림을 통해 전송되므로, 시스템 초기화시에는 로드될 수 없다. 따라서, Xlet을 JavaVM에 로드하기 위해서는 Personal Java에서 기본적으로 제공하는 ClassLoader의 기능을 확장해야 한다. 사용자가 특정 애플리케이션을 실행시

키려 할 경우, XletMgr는 해당 Xlet의 Root Resource를 찾아 내어, 이를 XletLoader를 통해 로딩한다. 이때, XletLoader는 Root Resource에서 참조하는 모든 관련 클래스들도 같이 로드한다. XletThread는 모든 Xlet은 각각의 XletThread에 의해 수행된다. XletLoader를 통해 Xlet이 성공적으로 로드되면, XletMgr는 XletThread를 생성하여 Xlet을 실행시킨다.

3. Data Carousel 관련 클래스

Xlet을 구성하는 대부분의 리소스들은 Data Carousel을 통해 전송되며, Xlet은 CarouselFile 인터페이스를 통해 이들 리소스에 접근할 수 있다. Xlet이 CarouselFile 객체를 생성하면 PSIP/Data Broadcasting 모듈은 해당 데이터 모듈을 파일 시스템에 마운트 한 후, 해당 java.io.File 객체를 CarouselFile에 넘겨 준다. 이후로 CarouselFile에 대한 모든 접근은 java.io.File 객체를 통해 이루어지게 된다.

특정 데이터 모듈이 변경되었을 경우, PSIP/Data Broadcasting 모듈은 MsgQueue에 변경 사항을 기록한다. DataNotifier 쓰레드는 MsgQueue를 검사하여 특정 CarouselFile이 변경되었을 경우, 이를 해당 CarouselListener에 알려 주게 된다.

VI. 구현 사례

그림 6과 그림 7은 구현된 DASE 셋톱박스의 정면과 후면을 보여 준다. 그림 8의 내부의 모습이다.

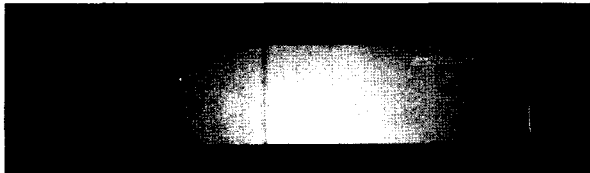


그림 6. 구현된 DASE 셋톱의 외관 (정면)
Fig. 6. DASE settopbox (front)



그림 7. 구현된 DASE 셋톱의 외관 (후면)
Fig. 7. DASE settopbox (back)

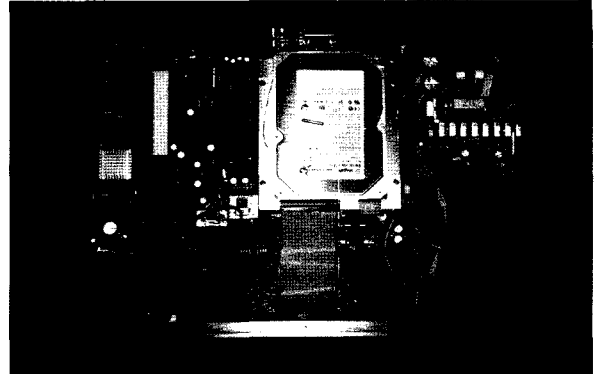


그림 8. 구현된 DASE 셋톱의 내부
Fig. 8. DASE settopbox hardware

그림 9는 브라우저 화면으로 XXML로 구성된 콘텐츠를 디코딩하여 화면에 표시하였다. 그림10에 이 콘텐츠와 관련된 DASE 응용프로그램의 실행 과정을 나타내었다.

브라우저 부분과 그래픽 루틴 등도 속도 개선을 위해 다음과 같은 작업을 수행하였다.

- (1) 일반 JavaVM이 아니라 임베디드 시스템에 최적화된 Personal JWorks 사용
 - 바이트 코드 인터프리터를 어셈블리어로 구현
 - 자바 쓰레드가 VxWorks의 태스크로 직접 매핑되도록 함
- (2) AWT drawing 루틴 최적화
 - Java AWT의 repaint() 함수를 수정하여 recursive function call을 최소화
 - 이미지를 표시할 때 최대한 native 함수를 사용하여 성능 향상 도모



그림 9. 브라우저상에 구현된 XXML 콘텐츠
Fig. 9. XML contents on browser

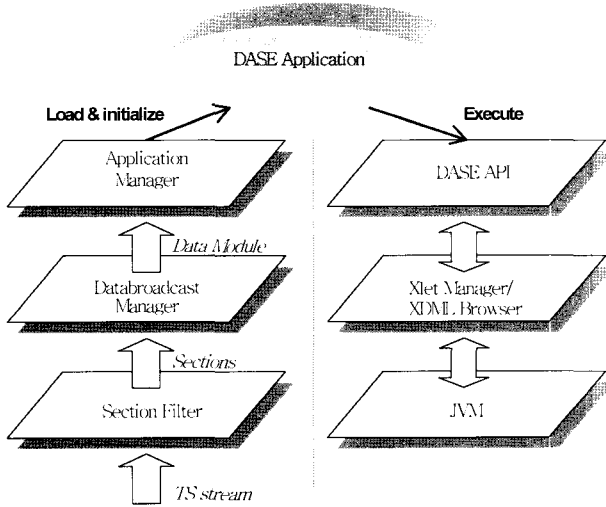


그림 10. DASE 애플리케이션의 실행 과정
Fig. 10. Flow of DASE application

다음 화면은 JAVA VM 상에서 Xlet이 동작하는 실제 화면이다.

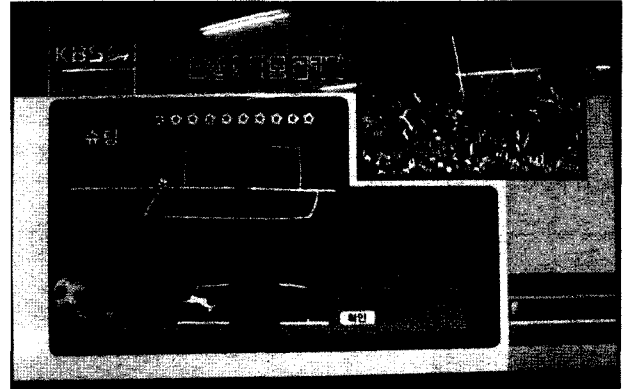
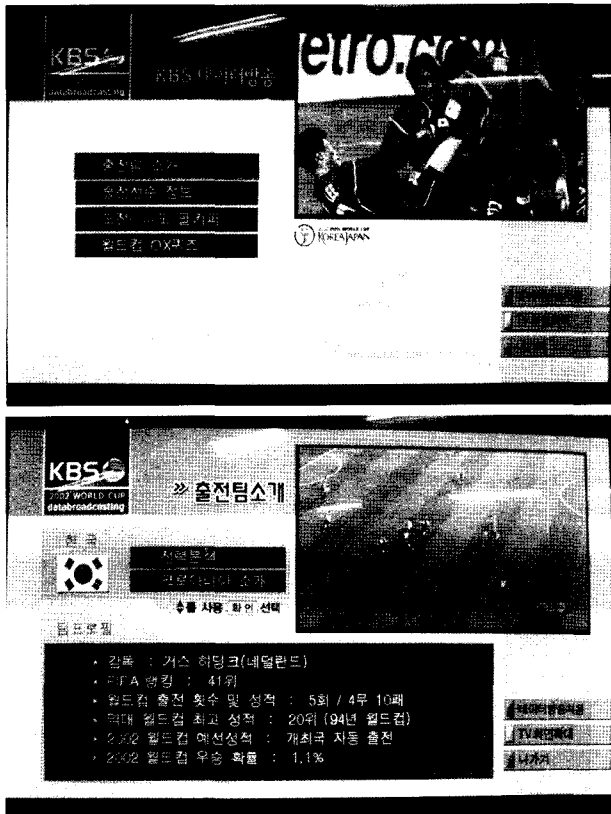


그림 11. Xlet 동작 화면
Fig. 11. Xlet application

Ⅶ. 결 론

본 논문에서는 DASE 규격을 만족하는 셋톱 박스를 개발하였다. DASE 셋톱 박스는 ATSC의 PSIP 및 데이터 방송 표준을 지원하며, 이를 기반으로 XDML, Xlet 애플리케이션을 처리한다. ATSC DASE 규격에 준하는 DTV 셋톱 박스를 개발함으로써 HDTV 방송을 수신할 수 있을 뿐 아니라, 향후 제공될 다양한 데이터 서비스를 지원할 수 있다. 또한 콘텐츠 제작자에게 레퍼런스 플랫폼을 제공함으로써 데이터 서비스의 상용화에도 일조할 수 있으리라 기대된다. 본 논문에서 가장 문제가 되었던 부분은 Xlet을 처리하기 위한 JavaVM의 성능 및 속도 문제였다. 임베디드 환경에서의 느린 속도와 기능 제한 등의 근본 문제를 해결하기 위해 하드웨어와 미들웨어의 개선은 물론 Authoring Tool에서의 셋탑환경에 적합한 콘텐츠의 제작등이 필수적이며 이는 향후 방송사 및 콘텐츠 제작사와의 긴밀한 협조가 필요한 부분이라고 본다.

참 고 문 헌

- [1] "DTV Application Software Environment Standard Part 1: Introduction, Architecture, and Common Facilities," *Draft ATSC Standard T3-528*, Revision 1, 9 Feb, 2001.
- [2] "DTV Application Software Environment Standard Part 2: Declarative Applications and Environment," *Draft ATSC Standard T3-529*, Revision 1, 9 Feb, 2001.
- [3] "DTV Application Software Environment Standard Part 3: Procedural Applications and Environment," *Draft ATSC Standard*

T3-530, Revision 1, 9 Feb. 2001.

[4] "Data Broadcast Standard," ATSC Standard A/90, 26 Jul. 2000.

[5] "Program and System Information Protocol for Terrestrial

Broadcast and Cable," ATSC Standard A/65, Revision A and Amendment No. 1, 31 May. 2000.

저 자 소 개



박 용 규

- 1987년 : 서울대학교 공과대학 전기공학과 졸업
- 1989년 : 서울대학교 대학원 전기공학과 졸업 (석사)
- 1993년 : 서울대학교 대학원 전기공학과 졸업 (박사)
- 1990년~1999년 : 대우전자 DTV 사업부 책임연구원
- 1994년~1995년 : 스펀포드 대학교 컴퓨터공학과 객원연구원
- 1999년~현재 : (주)마르시스 대표이사



장 규 환

- 1977년 : 서울대학교 공과대학 전자공학과 졸업
- 1993년~1996년 : 대우전자 영상연구소장
- 1997년~2001년 : 대우전자 디지털TV 사업부장 (이사)
- 2001년~2002년 : 대우전자 디지털 사업부장 (상무)
- 2002년 : 대우전자 대표이사
- 1996년 : 미시간대 최고 경영자 과정 수료
- 2002년 현재 : 생산기술연구원 국책과제 심의위원, HDTV ASIC 과제 운영위원, 전자공학회 편집위원