

실시간 OS용 ITV 개발¹

김상욱*, 신규상**, 마평수**, 신용경*, 이윤주*, 김원국*, 조창식**

*경북대학교 컴퓨터과학과

**한국전자통신연구원

e-mail : swkim@woorisol.knu.ac.kr

ITV Development on Realtime OS

S Kim*, G. Shin**, P. Mah**, Y. Shin*, Y. Lee*, W. Kim*, C. Cho**

*Dept of Computer Science, Kyungpook University

**ETRI

요약

이 논문에서는 조립형 실시간 OS에서 운용되는 지능형 텔레비전(Intelligent Television, ITV)을 개발한다. 인터넷에서의 웹기반 TV에 대하여 여러 연구가 있었지만, 실시간 운영체제가 아니어서 여러 문제를 가지고 있다. 이 논문은 실시간 OS에서의 ITV 개발에서 MPEG-4 프리젠테이션을 중심으로 설명한다. 즉, 미디어 서버에서 분배 관리기, MPEG-4 압축, 미디어 재생기까지의 일련의 미디어 전송 과정을 거쳐 프리젠테이션 과정을 기술한다. 이 논문의 ITV는 한국전자통신연구원에서 개발 중인 실시간 운영체제에서 운용되며, MPEG-1은 물론 MPEG-4 스트림을 지원함으로써 사용자 상호작용이 효율적이다. 또한, MPEG-7 스트림으로 확장될 수 있다. 이 연구의 결과는 QoS는 물론 전송과 재생 속도의 향상을 가져오므로 디지털 TV(DTV) 등에 활용될 수 있다.

1. 서론

본 논문의 목적은 조립형 실시간 OS에서 구동하고, 사용자 요구사항을 만족시키는 ITV를 개발하는 것이다.

웹이나 인터넷에서의 TV에 대한 연구는 여러 연구가 있다[1, 2]. 그러나 이 논문에서 제시하는 ITV는 기존의 Web TV나 인터넷 TV와는 다르다.

이 논문의 ITV는 실시간 OS에서 운용되며 MPEG-1은 물론 MPEG-4 스트림을 지원함으로써 사용자의 상호작용을 효율적으로 지원한다. 그러므로 국제 표준의 MPEG-7을 활용하는 TV로 확정될 수 있다.

본 논문은 실시간 OS에서의 ITV개발에서 MPEG-4 프리젠테이션을 중심으로 설명한다[7]. 즉, 미디어 서버에서 분배 관리기, MPEG-4 압축, 미디어 플레이어까지의 일련의 미디어 전송 과정을 거쳐 프리젠테이션 되는 과정을 기술한다.

본 연구는 실시간 OS를 사용함으로써 QoS는 물론 전송과 재생 속도의 향상을 가져오며, 기존의 MPEG-1은 물론 MPEG-4의 재생을 지원함으로써 사용자의 상호작용을 월등히 지원한다. 그러므로 디지털 TV(DTV)[4] 등에 활용될 수 있는 중요한 연구이다.

이 논문은 제 2 절에서는 실시간 OS용 ITV 시스템[3]의 전반적인 개요를 설명하고, 제 3 절에서는

1. 본 연구는 한국전자통신연구원 실시간 컴퓨팅 연구부 `99 국책과제의 결과임

본 논문의 중심인 미디어 프리젠테이션에 대하여 기술한다. 제 4 절은 개발된 결과를 설명하고, 제 5 절에 결론이 있다.

2. 실시간 OS 용 ITV 시스템

ITV용 응용 API 서브시스템은 커널 서브시스템과 라이브러리 서브시스템의 상위에 위치하고 있으며, VBI 뷰어, 웹 브라우저, 전자우편, 미디어 재생기 등의 모듈로 구성된다. 이러한 모듈은 ITV API, Media API, Content Delivery API 등과 같은 라이브러리를 사용하여 구현된다.

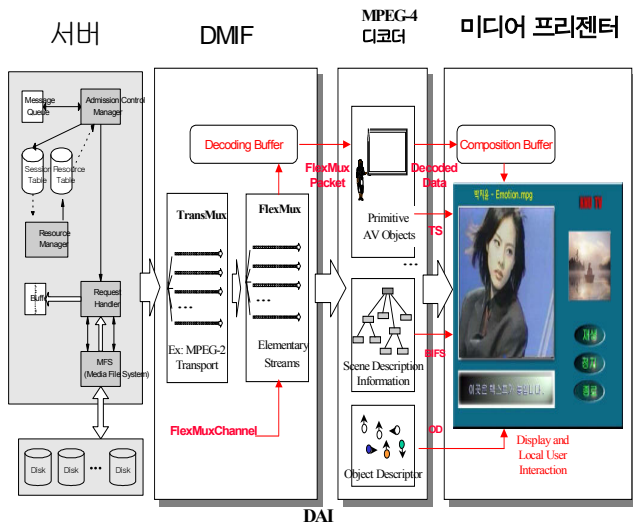


그림 1. 실시간 OS용 ITV 환경

운영체제

실시간 OS는 한국전자통신연구소에서 개발하는 조립형 실시간 OS이다. 이 서브시스템에서는 커널 서브시스템의 Task Management API, Semaphore API, Event API, Network I/O API 등을 사용한다. 또한 마우스, 리모콘, TV Tuner 등의 Device를 제어하기 위해 커널 서브시스템에서 제공하는 인터페이스를 사용한다. 뿐만 아니라, 라이브러리 서브시스템의 C 라이브러리와 그래픽/윈도우 라이브러리, 플래시 파일 시스템 등을 사용한다.

미디어 저장과 전송

미디어 서버는 ITV 클라이언트에서 요구하는 멀

티미디어 파일 서비스를 담당한다. 본 연구의 VOD 서버는 연속 매체를 서비스하기 위해서 기존의 텍스트용 Linux Ext2 파일시스템의 디스크 배열이다. 이 파일 시스템은 고속의 대용량 저장 장치를 요구하는 멀티미디어 데이터들을 관리하고, 접근하는 기능을 가진다.

ITV 클라이언트와 DMIF DAI 인터페이스를 사용하여 통신하므로, 네트워크에 관련된 모든 사항이 투명하다. 서버가 지원하는 미디어 스트림은 MPEG-1, MP3, MPEG-4 이다.

서버의 주요 기능은 권한 제어, 세션 관리, 자원(네트워크, 디스크, 버퍼) 관리, VCR 연산 지원이다. 서버가 지원하는 VCR 기능은 재생, 일시 정지, 정지, 점프(임의 위치 재생), 종료가 있다.

미디어 플레이어

미디어 플레이어는 미디어 서버에 저장되어 있는 비디오/오디오와 같은 멀티미디어 파일을 네트워크를 통하여 전송하고, 실시간으로 디코딩 한 후 재생하는 기능을 제공한다. 물론 오디오, 비디오, 텍스트의 동기화 기능도 지원한다. 또한 MPEG-4의 경우에는 각 미디어 별로 위치 이동, 크기 조절, On/Off 등의 사용자 인터랙션 기능도 제공한다.

미디어 플레이어는 미디어 프리젠테이션, MPEG-1/, MPEG-4 디코더, Delivery Manager로 구분된다.

MPEG-4 디코더는 내부에 비디오, 오디오, 텍스트 등을 개별적으로 복호할 수 있는 개별 디코더를 내장한다. 네트워크를 통하여 MPEG-4 디코더, 즉 ITV 클라이언트로 전송되어온 TransMux 스트림속의 다양한 형태의 멀티미디어 자료를 개별 Elementary stream(ES)으로 구분한다. 분리된 각각의 ES를 적합한 디코더로 분배하여 원래의 비디오, 오디오, 텍스트로 복원한다.

분배 관리자(Delivery Manager)는 미디어 프리젠테이션과 스토리지 서버간의 통신을 담당한다. 미디어 프리젠테이션과 스토리지 서버 사이의 제어 전송과 데이터 전송은 서로 다른 네트워크 채널로 분리된다. 분배 관리기는 미디어 프리젠테이션의 이벤트 관리기로부터 받은 제어를 제어 전송용 네트워크 채널을 이용하여 스토리지 서버의 메시지 큐에 저장한다[6].

그 후, 데이터 전송용 네트워크 채널을 통해 스토

리지 서버의 버퍼로부터 미디어 프리젠티어의 디코딩 버퍼에 데이터를 전달한다. 제어 전송은 신뢰성을 보장하여야 하므로 TCP를 통해 이루어지며 데이터 전송은 UDP를 통해 이루어진다. 클라이언트 측 분배 관리기의 FlexDemux는 MPEG-4 Flexmux 스트림에만 적용된다. 서로 mux된 Flexmux 스트림을 받아 각각의 스트림으로 분리하여 디코딩 버퍼에 쓴다.

3. 미디어 프리젠티이션

미디어 프리젠티어는 MPEG-1 프리젠티어, MPEG-4 프리젠티어, MP3 프리젠티어로 나뉘어진다. 각각은 파일의 확장자를 분석해서 분리된다. 각각의 미디어 프리젠티어는 동일한 인터페이스를 가지지만 독립적인 어플리케이션으로 동작한다. MPEG-1, MP3, MPEG-4 프리젠티어는 재생, 일시 정지, 정지 버튼을 메뉴로 가지고 플레이 되는 스트림의 진행 상태를 보여주는 상태바를 가진다. MPEG-1, MP3는 각각의 디코더를 통해 디바이스로 내보내는 형태로 구성된다. MPEG-4는 서버로부터 장면 구성에 관한 정보를 받아 장면을 구성하고 사용자 상호작용을 제공해 준다. MPEG-4에서 지원되는 미디어 타입은 2D 객체, 이미지, 텍스트, 비디오, 오디오이다.

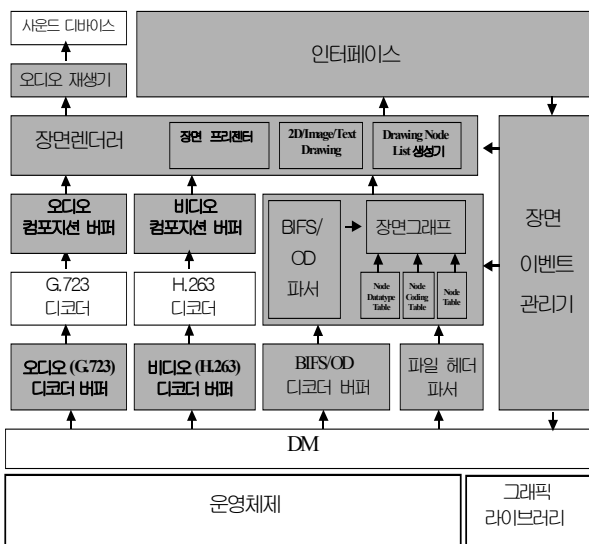


그림 2. ITV용 MPEG-4 프리젠티어 구조

MPEG-1, MP3의 각 프리젠티어는 비디오/오디오/텍스트 컴포지션 버퍼, 동기화 관리기, 이벤트 관리기, MPEG-1 오디오 프리젠티어, 인터페이스로 구성

된다. 분배 관리기를 통해 MPEG-1, MP3 미디어 스트림을 전송 받아 각 디코더 버퍼에 저장한다. 각 디코더는 디코더 버퍼의 내용을 읽어 디코딩 한 다음 비디오와 오디오를 각각 비디오, 오디오 버퍼에 적는다. MPEG-1 비디오/오디오 프리젠티어는 각 버퍼의 내용을 확인하여 프리젠티이션 시간이 되면 인터페이스와 오디오 디바이스를 통해 내보낸다. 인터페이스를 통해 정지, 재생 버튼을 누르면 이벤트 관리기는 디코딩을 중단시키고 분배 관리기를 통해 서버에 전송을 중단할 것을 요청한다.

MPEG-4 프리젠티어는 비디오/오디오 컴포지션 버퍼, 파서관리기, 장면렌더러, 오디오 프리젠티어, 공유 메모리 관리기, 이벤트 관리기, 파일 헤더 파서, 인터페이스로 구성된다. MPEG-4 프리젠티어는 분배 관리기로부터 받은 BIFS/OD 스트림 및 미디어 스트림을 채널 버퍼에 저장한다. 각각의 디코더 버퍼에 대해서 하나의 독립적인 작업이 수행된다. BIFS/OD 스트림에 대해서는 파서관리기가 동작하고 비디오/오디오 디코더 버퍼에 대해서는 비디오/오디오 디코더가 동작한다. BIFS/OD 파서를 통해서 장면 그래프가 만들어 지고 비디오/오디오 디코더를 통해서 비디오/오디오 컴포지션 버퍼에 저장된다. 장면 렌더러는 장면그래프를 통해 Drawable Node List를 생성하고 이를 2D/Image/Text Drawing 모듈과 비디오/오디오 컴포지션 버퍼에 있는 미디어로 장면을 구성한다. 이를 오디오 프리젠티어와 인터페이스를 통해 내보낸다.

채널 버퍼

채널 버퍼는 채널로부터 오는 미디어 데이터를 디코딩하기 위해 읽어 들이는 버퍼이다. 각각의 채널 버퍼는 비디오/오디오/BIFS/OD 의 디코더 버퍼로 사용된다. 각 버퍼는 외부변수로 선언되어 있고 이는 공유메모리 관리기에 의해서 생성/삭제된다.

파서

파서는 분배 관리기를 통해 전송 받은 미디어 스트림을 프리젠티이션 할 때, 시청각 장면을 렌더링 할 수 있도록 SD를 파싱하고, 장면을 구성한다. 즉, BIFS/OD 파서는 기본 스트림 형태의 멀티미디어 정보를 파서 관리기에서 읽어 들여 파싱 및 해석하

고, 그 결과를 프리젠티가 시청각 장면을 디스플레이하기 위한 정보로 형상화한다. 또한, 파서 관리기는 형상화한 정보, 파싱에 필요한 노드 데이터 테이블, 노드 코딩 테이블, 그리고 노드 테이블에 대한 정보를 관리한다.

파서는 분배 관리기로부터 전달된 BIFS/OD 스트림을 파싱한 후 그 결과를 프리젠티 인터페이스로 넘겨주는 역할을 한다. 이 정보는 장면 렌더러가 장면을 렌더링 하는데 사용된다.

파서 관리기는 노드 정보, BIFS/OD 파서, 명령어 관리기로 구성된다.

노드 정보

노드 정보는 파서가 스트림을 파싱할 때 참조하는 노드와 필드들에 관한 정보이다. 노드 정보의 종류는 다음과 같다.

- NT(Node Table)
- NDT(Node Data Table)
- NCT(Node Coding Table)

BIFS/OD 파서

BIFS/OD 파서는 파서 관리기의 핵심이 되는 모듈이다. 분배 관리기로부터 전달된 초기 객체 디스크립터 스트림을 파싱 및 해석해서 장면 디스크립터 스트림과 객체 디스크립터 스트림을 전송 받는다. 이를 받기 위해 분배 관리기에게 채널을 요청하고, 분배 관리기로부터 추가적으로 생성된 채널로부터 장면 디스크립터 스트림과 객체 디스크립터 스트림을 전송 받는다.

BIFS/OD 파서는 정보 관리기가 관리하는 노드와 필드들에 관한 정보를 참조하여 장면 디스크립터 스트림을 파싱 및 해석하여 장면 그래프를 구성하고, 프리젠티가 프리젠티이션 할 수 있도록 해석된 정보를 프리젠티 인터페이스로 넘겨준다. 한편, BIFS/OD 파서는 전송 받은 객체 디스크립터 스트림을 파싱 및 해석한다. 이렇게 해석된 정보를 이용하여 스트림 관리기는 장면을 구성하는 미디어 객체가 프리젠티이션 될 때 필요한 이미지, 비디오, 오디오

오등의 미디어 스트림을 전송 받기 위해 채널을 분배 관리기에게 채널 생성을 요청하고, 생성된 채널을 통해 전송되는 미디어 스트림과 이 스트림을 필요로 하는 미디어 객체를 연결시켜 프리젠티이션이 이루어지도록 한다.

BIFS/OD 파서는 OD파서와 BIFS파서의 2가지 모듈로 구성된다.

(1) OD 파서

OD파서는 초기 객체 디스크립터 스트림과 객체 디스크립터 스트림을 파싱 및 해석한다. 원격 서버에 대한 서비스 요청이 성공적으로 수행되었을 때 결과값으로 넘어오는 초기 객체 디스크립터 스트림을 파싱 및 해석한다. 해석한 정보의 스트림 타입에 따라 각각 장면 디스크립터 스트림, 객체 디스크립터 스트림을 전송 받기 위한 채널들의 생성을 채널 관리기에게 요청한다. 채널 관리기로부터 객체 디스크립터 스트림을 받았을 경우에 이를 파싱, 해석 및 관리하며, 해석한 정보의 스트림 타입과 스트림 타입에 대한 명세정보를 바탕으로 미디어 스트림에 대한 채널 생성을 채널 관리기에게 요청한다. 스트림 타입(StreamType)은 다음과 같다.

- InitialObjectDescriptorStream
- ObjectDescriptorStream
- SceneDescriptionStream
- VisualStream
- AudioStream

또한 스트림 타입에 대한 명세정보(SpecificInfo)는 다음과 같다.

- JPEG
- GIF
- H.263
- G.723
- MPEG-1
- MP3

(2) BIFS 파서

BIFS 파서는 채널 관리기로부터 넘겨 받은 장면

디스크립터 스트림을 정보 관리기의 노드 데이터 테이블, 노드 테이블, 노드 코딩 테이블을 참조하여 파싱 및 해석하여 장면을 구성하고 관리한다. 또한, 파싱 및 해석된 결과로 생성된 장면을 구성하는 각 노드들에 관한 정보를 프리젠타가 프리젠테이션할 수 있도록 장면 렌더러의 Drawable Node List에 넘긴다. Drawable Node List는 프리젠테이션을 위해 준비된 노드들의 리스트를 나타낸다.

명령 관리기

채널 관리기로부터 넘겨받은 장면 디스크립터 스트림과 객체 디스크립터 스트림을 해석하여 OD명령이나 BIFS명령을 수행한다. OD명령과 BIFS 명령을 통하여 초기 장면을 구성하고, 삽입, 삭제, 갱신의 명령을 통해 장면을 변화시킨다. 또한, OD명령을 통해 각 미디어 객체가 필요로 하는 미디어 스트림을 삽입, 삭제, 갱신함으로써 장면을 변화시킨다.

다음은 BIFS 명령이다.

- 장면 명령 (초기 장면을 구성)
- 삽입 명령 (노드, 필드, 라우터)
- 삭제 명령 (노드, 필드, 라우터)
- 갱신 명령 (노드, 필드, 라우터)

BIFS 명령은 하나의 AU(Access Unit)에 여러 명령어를 보내기 위해 CommandFrames로 그룹화 시킨다. 첫번째 명령은 BIFS scene의 초기 장면을 구성하여 BIFS 파싱을 시작하고, 나머지 3개의 명령들은 노드, 필드, 라우터의 구조로 갱신된다.

다음은 OD 명령이다.

- 갱신 명령 (미디어 스트림의 채널 연결)
- 삭제 명령 (미디어 스트림의 채널 삭제)
- 기본 스트림 갱신 명령
- 기본 스트림 삭제 명령 (미디어 스트림의 채널 삭제)

각 명령은 한 개 이상의 객체 디스크립터나 기본 스트림에 영향을 준다. OD 명령 중 갱신 명령은 새로운 OD이거나 갱신된 OD를 전달한다. 기존의 OD 컴포넌트는 이 명령에 의해 변경되지 않고 컴포

넌트 형태로 존재하는 ES_Descriptor만을 제거하거나 삽입한다. OD내에 있는 ES_Descriptor의 제거는 OD에 첨가되는 ES_Descriptor의 배열에서 제거되고, 삽입은 그 배열에 첨가되면서 명령이 처리된다. 그 배열은 0에서 255의 element를 가질 수 있다. OD 명령 중 삭제 명령은 일련의 OD를 불활성화시킨다. BIFS 노드들은 제거된 OD는 참조하지 않고 무시된다.

렌더러는 스트림을 포함하는 미디어 객체를 렌더링한다. 미디어 렌더러가 장면을 구성하고 있는 미디어 객체의 필드들 중 url 값과 OD 파서로부터 해석된 정보인 OD_ID를 참조한다. 또한 스트림을 받을 채널을 결정하고 이 채널을 통하여 미디어 스트림을 전송받아 미디어 객체 정보를 프리젠테이션한다[5].

4. 개발

본 연구의 ITV는 RISC 칩을 사용하고 있고, 메모리가 매우 적다. 또한 압축된 비디오/오디오의 복원도 소프트웨어로 구현되었다. 개발 언어는 C 언어만 사용하였다. 국내 가전 3사에서 작성한 ITV 표준안을 지원하며, 향후 DTV에서도 활용될 수 있도록 MPEG 등 국제 표준을 지원한다.

그림 3은 ITV에서 재생되고 있는 미디어 프리젠테터이다.



그림 3. 실시간 OS용 ITV의 MPEG-4 프리젠테터

5. 결론

본 논문은 실시간 OS에서의 ITV 개발을 MPEG-4 프리젠테이션을 중심으로 설명하였다.

미디어 서버의 멀티미디어 파일을 압축하고 미디어 플레이어에서 각 미디어 별로 프리젠테이션 한다. 이러한 연구는 기존의 MPEG-1은 물론 MPEG-4 의 재생을 지원함으로써 사용자의 상호작용을 월등히 지원한다. 뿐만 아니라, 실시간 OS를 사용함으로써 QoS는 물론 전송과 재생 속도의 향상을 가져온다. 이 연구는 디지털 TV(DTV) 등에 활용될 수 있는 중요한 연구이다.

앞으로는 MPEG-4 압축 기법을 더욱 발전시켜 재생 속도를 증가시키는 것이 필요하다. 또한, MPEG-7으로 확장하기 위하여 사용자의 상호작용과 검색 기능을 더 확장 발전시켜야 한다.

참고문헌

- [1] PowerTV, Inc. PowerTV Application Development API Reference, PowerTV, Inc., 1998.
- [2] Joseph T. Sinclair, Developing Web Pages with TV-HTML, 1998.
- [3] A. Beu, MUSIST (Multimedia User Interfaces For Interactive Systems and TV ACTS Project AC010, Final Online Style Guide, 1998.
- [4] John Washburn, DAVID 22, A Superior Foundation for Digital Television Solutions, Microware Systems Corporation, 1998.
- [5] Digital Audio Visual Council, DAVIC 1.0 Specifications, Digital Audio Visual Council, 1995-1996.
- [6] ISO/IEC JTC 1/SC 29N 1906, Information technology Generic coding of moving picture and associated audio information Part 6 : Delivery Multimedia Integration Framework, 97-10-31.
- [7] 김상욱, 멀티미디어 프리젠테이션 기술 연구, 한국전자통신연구원, 최종연구보고서, 1999.12.