

MHEG 엔진에 의한 VOD 시스템의 구현

이 준 형, 최 윤 식
연세대학교 전기공학과

An implementation of VOD system using MHEG engine

Joon-Hyoung Lee, Yoon-Sik Choe
Dept. of Electrical Engineering, Yonsei University

요약

본 논문에서는 멀티미디어·하이퍼미디어 표준 MHEG을 이용하여 차세대 영상 서비스 기술인 VOD(Video-on-Demand) 시스템을 구현하였다. 기존의 VOD 시스템과는 달리 다른 여러 가지 멀티미디어 서비스로의 확장성을 고려하여 MHEG의 데이터 구조를 이용하여 VOD 시스템의 데이터를 표현하고 이를 처리하기 위한 MHEG 엔진을 구성하였다. MHEG 객체를 이용하여 데이터를 처리하면 확장성이 보장되므로 다른 서비스를 추가할 경우 기존의 객체에 다른 서비스를 위한 객체만을 추가하면 된다. 이렇게 구성된 객체를 MHEG 엔진을 이용하여 처리함으로써 시스템에 독립적인 시스템의 구현이 용이하였다. 구체적인 MHEG 엔진의 설계를 위하여서는 기존의 LAN 상에서 구현한 VOD 시스템이 사용되었다.

1. 서론

여러 가지 다양한 압축기술의 등장과 대용량의 데이터를 저장할 수 있는 저장매체의 발달, 이러한 데이터를 빠른 속도로 전송할 수 있는 통신매체의 발달로 대화형 미디어 서비스의 구현이 실현되게 되었다. 이 중 가장 먼저 구체화되고 있는 서비스가 주문자 요구형 비디오, 즉 Video On Demand(VOD)이다. 최근 사용자의 요구가 다양화 되어감에 따라 VOD 뿐만이 아니라 다른 여러 가지 멀티미디어 서비스를 동시에 구현할 필요가 있게 되었으나 기존의 VOD 시스템으로는 이러한 요구를 만족할 수 없다. 따라서 VOD 서비스를 포함하여 여러 가지 멀티미디어·하이퍼미디어 서비스들을 동시에 처리

할 수 있는 시스템의 구현이 필요하게 되었다.

하이퍼미디어란 정보와 그 이용수준이 미리 정해진 관계를 가지고 있고 이들 정보 및 이용수준의 세트에 대해 사용자가 대화형으로 이용할 수 있도록 구성된 정보 미디어를 말한다. 이러한 멀티미디어·하이퍼미디어 정보를 조합한 하나의 객체로서 부호화하기 위한 시나리오 기술방법에 관한 국제표준이 MHEG(Multimedia Hypermedia Experts Group)인데, 본 논문에서는 이러한 MHEG을 이용하여 기존에 구현한 LAN 상에서의 VOD 시스템을 멀티미디어 서비스로 확장시켜보았다.

2. VOD 시스템

기존의 VOD 시스템은 크게 세가지 부분으로 나뉘어져 있다. 첫째 서버에 연결하여 자료선택을 위한 메타데이터를 전송받는 과정과 둘째 데이터를 선택하여 비디오 스트림을 전송받는 과정, 셋째 전송받는 비디오 스트림에 대해 사용자 제어를 처리하는 과정으로 요약될 수 있다.

이를 위해 우선 전송계층에서는 TCP (Transmission Control Protocol)을 사용하는데 이는 접속형 프로토콜이므로 스트림 형태의 비디오 데이터에 대한 전송의 신뢰성이 보장된다. UDP(User Datagram Protocol)는 비접속형 프로토콜로서 비디오 데이터의 크기가 작은 제어 신호 전송용으로 사용하였다.

그림 1은 VOD 시스템의 데이터 수신구조에 관한 블록다이어그램이다. 간단히 요약하면 우선 시스템을 초기화하고 버퍼를 할당하는 데 VOD 시스템의 버퍼관리는 DOS 모드에서의 버퍼관리와 MPEG-1 Decoder Board 인 RealMagic 드라이버에 의한 인터럽트로 나눌 수 있다. 초기화가 끝나면 소켓을 이용하여 메타데이터를 전

송받게 된다.

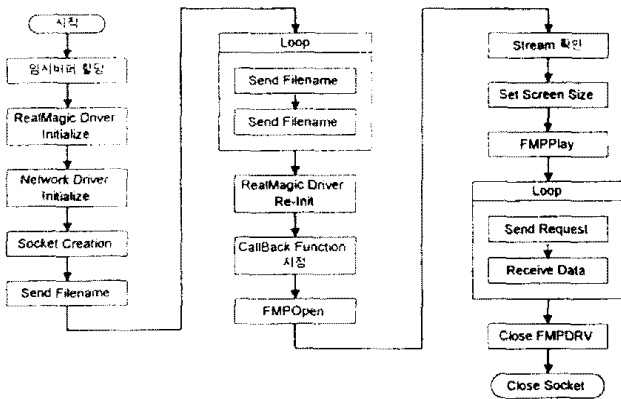


그림 1. VOD 시스템의 데이터 수신구조

메타데이터를 전송받고 데이터를 선택하면 Callback Function을 지정되고 스트림을 열어서 크기를 지정하여 재생을 하게 된다. MPEG 스트림의 전송은 일반 데이터와는 달리 파일의 크기에 시간의 단위가 추가된다. 즉 MPEG 스트림에 대해 버퍼의 크기로 분할해서 전송할 경우 버퍼 관리상의 문제점 때문에 분할기준을 시간단위로 처리해야한다. 따라서 전송시와 수신시의 데이터의 내용을 분석하여 I 프레임간의 간격으로 내용을 분할하여 처리한다. 스트림의 재생은 인터럽트를 통하여 후면작업으로 진행되므로 이후의 흐름은 전면작업으로 진행된다. 이후 계속적으로 버퍼를 네트워크에서 전송 받으며 버퍼의 스위칭은 Callback Function을 통해 인터럽트를 이용한 후면작업과 재진입의 반복으로 이루어진다.

사용자에 의한 제어 신호는 신호의 발생시 그에 해당되는 메시지가 Callback Function으로 전달되어 처리를 하게 되고, 이러한 제어 신호로는 play, stop, FF, RE 등의 VCR 기능이 포함되어 있다.

그러나 이러한 VOD 시스템은 다른 멀티미디어를 위한 확장성이 거의 없다. 즉, 다른 여러 가지 서비스를 구현하고자 한다면 기존의 시스템을 다른 특정한 서비스들의 요구 조건에 맞도록 모든 구조에 수정을 가해야 한다. 따라서 이러한 제약성을 없애도록 MHEG이라는 표준을 사용하여 확장성이 용이한 멀티미디어 시스템으로 구현해 본다.

3. MHEG

MHEG은 멀티미디어 정보들의 이용에 대한 조합을 하나의 객체로 부호화해서 멀티미디어·하이퍼미디어 타이틀로 제작하기 위한 시나리오 기술 방법에 관한 국제 표준이다. 이러한 표준이 등장하게 된 배경은 JPEG이나 MPEG같은 압축표준의 등장과 대용량의 데이터를

저장하고 교환할 수 있는 저장매체와 통신매체의 발달을 들 수 있는데, MHEG이 제공하는 것은 실행할 수 있는 코드라기보다는 멀티미디어·하이퍼미디어 표현과 그 교환방식이므로 실제로 각 미디어 정보가 어떻게 표시되는지는 응용에 맡겨지고 있다. 이러한 표현된 데이터의 구조를 MHEG 객체라고 하는데 각 MHEG 객체를 관리하거나 MHEG 객체를 해석하여 응용이 이해할 수 있도록 바꾸어 주는 기능을 MHEG 엔진을 이용하여 구현한다.

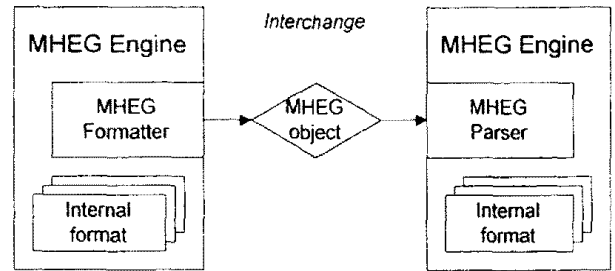


그림 2. MHEG 엔진간의 객체 교환

그림 2는 MHEG 시스템의 간단한 구조를 나타낸다. 교환된 MHEG 객체는 MHEG 엔진을 통하여 응용이 표현할 수 있는 최종양식(final-form representation)으로 변환되므로 응용에서는 어떠한 추가적인 처리도 필요하지 않다.

MHEG의 주요한 특징을 살펴보면, 우선 시스템에 대하여 독립적이고, 또한 멀티미디어 정보는 실시간으로 교환되고 표현되어야 한다. 멀티미디어 데이터는 시·공간상에서 구성하고 동기화가 가능해야 하며, 각 멀티미디어 데이터 요소들간에는 link에 의해 서로 연관되어 있다. 이러한 멀티미디어 데이터의 교환을 위해 MHEG 클래스를 정의하고 실제 교환과 표현을 위해 이 클래스로부터 객체를 생성한다. 객체가 사용자에 의해 감지되고 표현되는 논리적인 공간인 채널은 MHEG 엔진에 의해 실제 표현미디어에서 표현된다.

처음에 표준화된 MHEG-1은 너무 일반적이고 추상적이라서 특정한 응용에 대해 MHEG-1의 부하를 줄이기 위해 정의된 것이 MHEG-5인데 이 MHEG-5의 클래스 구조는 그림 3과 같다.

각 클래스는 action의 결과로 이벤트를 생성하는데, 이벤트는 하나의 특정객체로부터 생성되고, 각 MHEG-5 클래스에서는 그 클래스가 특정한 이벤트를 생성하는 환경을 정의한다. 각 이벤트들중 어떤 것들은 그들과 관련된 데이터 값을 가져 그 값들에 의해 관련된 link가 실행될지 말지를 결정한다. 따라서 MHEG 객체는 멀티미디어 데이터를 위한 객체의 구성과 각 객체들의

action의 결과로 생성되는 이벤트, 그리고 그 이벤트에 의한 link로 구성되어있다고 볼 수 있다.

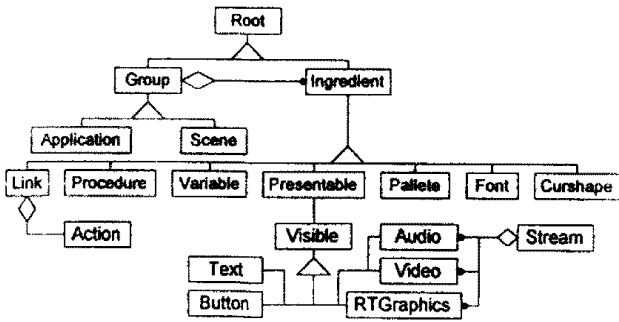


그림 3. MPEG-5의 클래스 구조

4. MPEG-5를 이용한 VOD 시스템의 구현

앞에서도 말했듯이 MPEG에서 정의하는 것은 상호교환을 위한 멀티미디어·하이퍼미디어의 표현이므로 VOD 시스템을 위한 객체의 구성에 대해서 살펴보면, 우선 VOD 서비스를 위한 객체는 메타데이터를 전송받는 Scene1객체, 스트림을 보여주는 Application객체, 사용자에게 의한 제어 버튼을 위한 Scene2객체로 구성된다. 그림 4는 이 객체를 나타낸다.

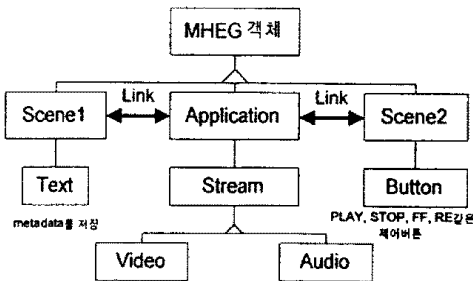


그림 4. VOD 시스템을 위한 MPEG 객체

Application 객체와 Scene 객체는 모두 Ingredient 클래스의 객체들을 Grouping하는데 Application은 응용전체에 걸쳐서 공유되는 객체의 set을 Grouping해주므로 Stream 객체는 Application객체에 포함해주었다. Application객체는 우선 비디오를 보여줘야 하므로 Presentable객체의 하위 객체인 Visible객체를 포함한다. Visible객체 중에서도 MPEG-1 스트림을 처리해야 하므로 Stream객체를 사용하는데 Stream객체는 Audio객체와 Video객체가 Multiplex되어 있다. 이렇게 비디오 스트림을 위한 객체 외에 다른 객체와의 상호작용을 위해 Link객체를 사용한다. Link객체는 주어진 조건에 대해

Scene객체로 Transition하여 각각의 이벤트를 처리하게 된다.

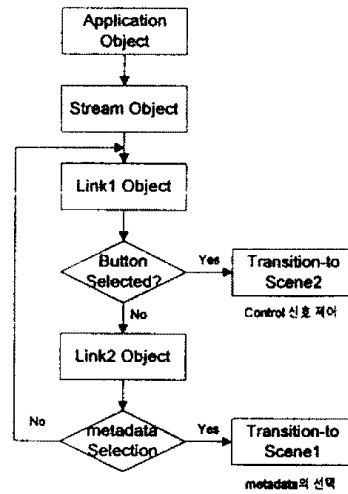


그림 5. Application 객체의 구성

Scene1객체는 메타데이터를 전송받아 선택하기 위한 객체이므로 메타데이터를 위한 Text객체와 데이터의 선택 후 Application으로 Transition하기 위한 Link객체로 구성되어 있다. 따라서 메타데이터가 선택되기만 하면 Link객체에 의해 선택된 데이터를 Application 객체로 전달하는 기능을 한다.

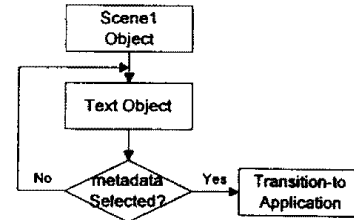


그림 6. Scene1 객체의 구조

Scene2객체는 사용자 입력을 위한 제어 버튼을 만들고 그들의 선택 여하에 따라 Stream객체에 action을 가하게 된다. 따라서 Visible 클래스의 하위 클래스인 Button 클래스로부터 Push-button 객체와 버튼이 선택되었을 경우 생성된 이벤트 처리를 위한 Link객체로 구성된다.

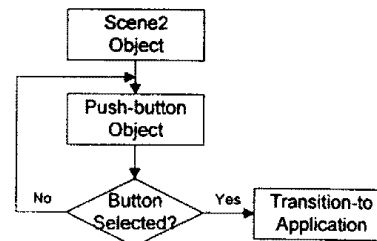


그림 7. Scene2 객체의 구조

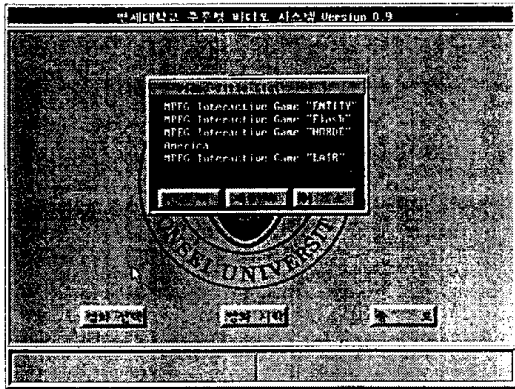
Scene2객체에서는 Button이 선택되었을 때 각 Button에 해당되는 action을 Application 객체로 전달해준다. 이러한 객체의 구현은 JAVA와 같은 객체 지향형 언어로 구현될 수 있으며, 일 예로 Application객체의 구현을 위한 Formal Description은 다음과 같다.

```

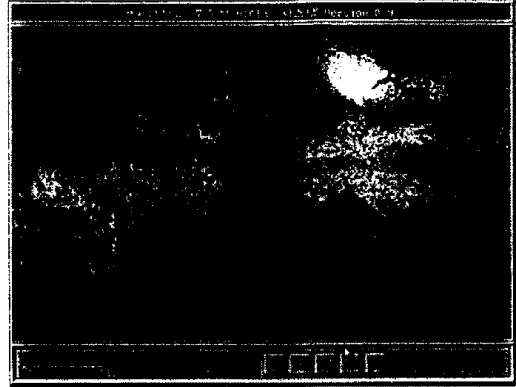
(:application
 :object-identifier (VOD 0)
 :on-start-up (:transition-to ((scenel 0) 1) )
 :group-items{
 (:stream
 :object-identifier 1
 :initially-active false
 :content-hook :#MHEG-content-register
 :content-data :referenced-content "MPEG-1 data"
 )
 (:link
 :object-identifier 2
 :event-source (scenel Text-object)
 :event-type IsSelected
 :link-effect (
 :transition-to((scenel 0) 1)
 )
 )
 (:link
 :object-identifier 3
 :event-source 0
 :event-type User Input
 :link-effect (
 :transition-to((scene2 0) 1)
 )
 )
 )
 )

```

그림 8은 실제로 MHEG 엔진에 의해 구현한 VOD 시스템이다. 기본적인 틀은 이전에 LAN상에서 구현한 VOD 시스템과 동일하다. 그러나 실제 구현에 있어서는 MHEG의 객체 구조를 이용하여 간단한 MHEG 엔진을 구성하였다



(a) 메타데이터를 전송받는 화면



(b) 자료를 선택한 후 전송받은 스트림을 보여주는 화면

그림 8. 구현한 VOD 시스템

5. 결론

본 논문에서는 MHEG 엔진을 이용하여 대화형 미디어 서비스인 VOD 시스템을 구현해 보았다. 기존의 VOD 시스템이 확장성이 없다는 점을 고려하여 MHEG이라는 멀티미디어·하이퍼미디어 표준을 적용하였다. MHEG객체를 이용하여 VOD 시스템을 표현하기위해서 스트림을 보여주는 Application객체, 메타데이터를 전송받아 선택하는 Scene1객체, 사용자에게 의한 제어 버튼을 위한 Scene2객체로 구성하였다.

여기서 구현한 MHEG 객체는 보다 나은 서비스를 위하여 수정이 가능하다. 향후 멀티미디어 서비스에서는 실시간 교환과 처리가 필수적이므로 이를 고려한 MHEG 객체의 설계는 모든 멀티미디어 서비스들에 대해서 MHEG의 개념을 적용할 수 있는 MHEG 엔진의 구현과 다른 서비스들에의 적용 연구를 통하여 이루어질 수 있을 것이다.

참고문헌

- [1] ISO/IEC DIS 13522-1 Part 1 : *MHEG object representation*, July 1996.
- [2] ISO/IEC DIS 13522-5 Part 5 : *Support for Base-Level Interactive Applications*, December 1995
- [3] Roger Price, "MHEG: An Introduction to the future INternational Standard for Hypermedia Object Interchange". ACM Multimedia 93, July 1993
- [4] "주동형 비디오 VOD". 전자저널, 1994년 8월.
- [5] W. Richard Stevens, "UNIX Network Programming", Prentice Hall, 1994.