

# MPEG-4 기반 멀티미디어 프리젠테이션 엔진

김상욱, 김희선, 차경애, 권순동<sup>0</sup>, 배수영, 정영우, 최상길  
경북대학교 컴퓨터과학과 컴퓨터언어/멀티미디어 연구실

## Multimedia Presentation Engine based on MPEG-4

Sangwook Kim, Heesun Kim, Kyungae Cha, Soondong Kwon<sup>0</sup>,  
Suyoung Bae, Youngwoo Jung, Sangkil Choi  
Computer Languages & Multimedia Laboratory  
Department of Computer Science, Kyungpook National University

### 요 약

본산 환경에서 멀티미디어 데이터를 효율적으로 전송하고 프리젠테이션하기 위한 연구가 활발히 진행되고 있다. 이런 연구에 관한 표준안인 MPEG-4는 다양한 형태의 멀티미디어 데이터를 투명하게 접근하고 사용자의 상호작용이 가능한 시청각 장면을 전송한다.

본 논문에서는 MPEG-4의 기본 스트림을 파싱하고, 이를 바탕으로 상호작용 가능한 장면을 구성하여 프리젠테이션하는 MPEG-4 기반 멀티미디어 프리젠테이션 엔진을 제안한다. 제안하는 엔진은 MPEG-4의 다양한 스트림을 해석하여 멀티미디어 객체를 생성하고 합성하여 프리젠테이션할 장면을 구성한다. 객체간의 동기화된 렌더링을 제공하고 프리젠테이션된 장면을 통한 사용자 상호작용을 지원한다.

## 1. 서 론

MPEG-4[1-3]는 대화형 멀티미디어 어플리케이션의 지원을 목표로 하여 상호작용 가능한 시청각 장면(scene)을 효율적으로 전송하고 표현한다. MPEG-4 표준은 사용자로 하여금 다양한 어플리케이션을 구성할 수 있게 시스템을 구성 및 제어하는 기능을 제공한다. 즉, 자연 영상, 음향 정보, 합성 영상 등의 다양한 형태의 시청각 미디어를 유연성 있고 확장성 있게 통합하여 대화형으로 사용자에게 제공하고, 이에 따라 사용자는 단순히 보기만 하던 것을 탈피하여 그들과 상호작용 할 수 있다.

대화형 멀티미디어 어플리케이션을 위해 MPEG-4는 미디어 스트림을 네트워크 상에서 전송, 인코딩, 디코딩, 동기화 할뿐만 아니라 스트림으로부터 멀티미디어 정보를 추출하여 이를 사용자의 출력 장치에 렌더링하고, 사용자와의 상호 작용을 처리하는 부분도 필요하다.

본 논문에서는 MPEG-4의 기본 스트림을 파싱하고, 이를 바탕으로 대화형 장면(scene)을 구성하여 프리젠테이션하는 MPEG-4 기반 멀티미디어 프리젠테이션 엔진을 제안한다.

기존에 개발된 MPEG-4 플레이어로는 Telenor사의 MPEG-4 버전1 플레이어[4]와 CSELT의 2차원 MPEG-4 플레이어[5]가 있다.

Telenor사의 MPEG-4 버전1 플레이어는 MPEG-4 표준안이 공식적으로 발표되기 전 MPEG-4 버전1 표준의 구현 가능성을 보이기 위해 Telenor사가 AT&T lab, CSELT, France Telecom CCETT, HHI, Hyundai Electronics Industries 등의 MPEG 관계자들과 협력하여 개발한 플랫폼 독립적인 플레이어로서, 이 플레이어는 현재 Windows 95/NT, IRIX, Solaris, Linux의 플랫폼을 지원한다. CSELT의 2차원 MPEG-4 플레이어[5]는 CSELT, VDO, ENST, QTEAM, FZI 등의 멤버로 구성된 IM1 그룹내에서 수행된 프로젝트의 결과로서 MPEG-4 CD 표준을 미리 구현되었으나, 3차원을 지원하지 못하는 문제점을 가진다.

본 논문에서 제안하는 MPEG-4 기반 멀티미디어 프리젠테이션 엔진은 MPEG-4 기본 스트림에서 멀티미디어 객체를 추출하고 합성하여 장면 트리와 이벤트, 액션 테이블을 구성하여 객체를 관리한다. 장면을 렌더링하기 위해서 MicroSort사의 DirectShow[6]를 이용한다. 렌더링된 객체에 대한 사용자 이벤트를 처리하여 장면을 재구성하고 표현함으로써, 대화형 멀티미디어 프리젠테이션을 지원한다.

본 논문의 2절에서는 MPEG-4 기반 멀티미디어 프리젠테이션 엔진의 구조와 각 모듈에 대하여 설명하고 3절에서 결론을 맺는다.

## 2. MPEG-4 기반 멀티미디어 프리젠테이션 엔진

### 2.1 엔진 구조

멀티미디어 프리젠테이션 엔진 구조는 Scene 컴포지터, 상호작용 처리기, 필터 관리기로 구성된다. 멀티미디어 스트림에 대한 표준은 MPEG-4를 따르고 멀티미디어 데이터에 대한 프리젠테이션을 위해 DirectShow를 이용한다.

그림 1은 멀티미디어 프리젠테이션 엔진 구조를 나타낸다.

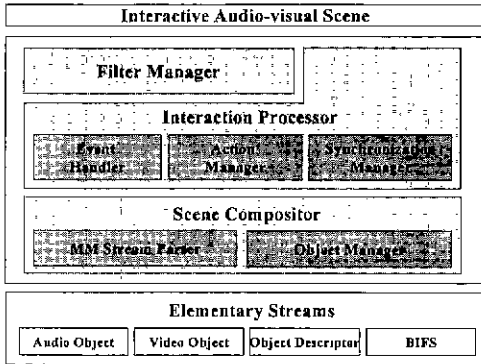


그림 1 멀티미디어 프리젠테이션 엔진 구조

컴포지터[Scene Composer]는 멀티미디어 스트림을 파싱하여 프리젠테이션할 장면을 구성한다.

상호작용 처리기(Interaction Processor)는 구성된 장면 정보를 필터와 매핑하고, 사용자 이벤트를 처리한다.

필터 관리자(Filter Manager)는 전송 받은 멀티미디어 객체를 이용하여 3차원으로 프리젠테이션 한다.

### 2.2 컴포지터[Scene Composer]

컴포지터[Scene Composer]는 스트림으로 표현된 미디어 정보를 파싱하여 객체 트리를 생성 및 관리 한다. 컴포지터는 멀티미디어 스트림 파서, 객체 관리기, 액션 테이블, 이벤트 테이블로 구성된다.

그림 2는 컴포지터의 구성 모듈과 흐름을 나타낸 것이다

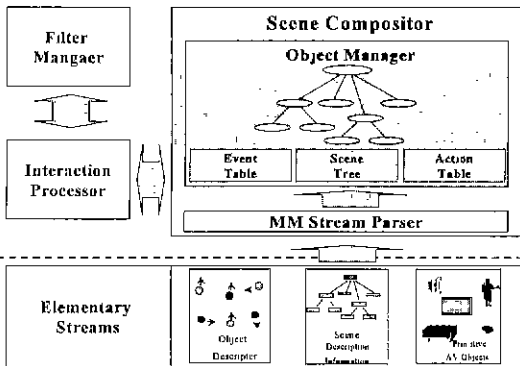


그림 2 컴포지터의 구조

멀티미디어 스트림 파서는 장면을 나타내는 BIFS( Binary Format for Scenes ) 스트림과 원시 미디어 객체 스트림을 받아 이를 파싱하고 그 결과를 트리와 테이블 구조로 변환한다. 즉, 하부 구조로부터 받은 스트림 정보를 해석해서 객체 관리기에서 처리할 수 있는 장면 트리와 이벤트 테이블, 액션 테이블을 생성한다.

객체 관리기는 파서로부터 생성된 장면 트리와 이벤트, 액션 테이블을 사용자 상호작용과 관련하여 갱신, 저장하는 역할을 수행한다.

파서를 통해 생성된 장면 트리의 모든 노드들은 객체를 나타낸다. 트리의 말단 노드는 오디오나 비디오 등의 원시 미디어 객체를 나타내고, 중간 노드는 하위 노드들의 제어 정보를 가진 콤플렉스 객체를 나타낸다. 이벤트와 액션에 대한 내용은 이벤트 테이블과 액션 테이블을 통해서 관리하고 각 노드는 두 테이블의 해당 필드에 대한 포인터를 가진다.

아래 그림 3은 이벤트 테이블과 액션 테이블의 구조를 나타낸다.

Event Type	Object ID	Action ID
...	...	...

(a) 이벤트 테이블

Action ID	Object ID	Action	Next Event	Sync Info
...	...	...	...	...

(b) 액션 테이블

그림 3 이벤트 테이블과 액션 테이블 구조

이벤트 테이블은 발생한 이벤트의 종류를 나타내는 Event Type 필드, 이벤트가 발생한 객체의 ID를 나타내는 Object ID 필드, 그리고 이벤트가 발생했을 때 취할 액션의 ID를 나타내는 Action ID 필드가 있다.

액션 테이블은 발생할 이벤트에 대한 액션 정보를 테이블 구조로 저장하고, 저장된 정보는 상호작용 처리기에서 사용된다. 액션 테이블에는 액션의 종류를 나타내는 Action ID 필드, 이벤트가 발생한 객체를 나타내는 Object ID 필드, 실제 수행할 액션 나타내는 Action 필드, 액션 수행 후 부수적으로 발생하는 이벤트를 위한 Next Event 필드, 그리고 이벤트의 동기화 정보를 가리키는 Sync Info 필드가 있다. Next Event 필드는 여러 개의 이벤트가 발생할 경우를 위해 연결 리스트의 형태를 가진다.

### 2.3 상호작용 처리기(Interaction Processor)

상호작용 처리기는 장면 합성기로부터 얻은 객체 정보를 필터 관리기와 연결하고, 3차원 장면을 통해 입력된 사용자 이벤트를 해석하고 처리하며, 객체간의 동기화를 수행한다.

상호작용 처리기는 그림 4와 같이 이벤트 핸들러, 동기화 관리기, 액션 관리기로 구성되어 있다.

객체 관리기가 장면 트리를 완성했을 때, 프리젠테이션 초기화 이벤트를 이벤트 핸들러에게 전달한다. 이벤트를 수신한 이벤트 핸들러는 동기화 관리기와 액션 관리기를 초기화시키고, 장면 트리의 루트 노드에서 프리미티브 객체에

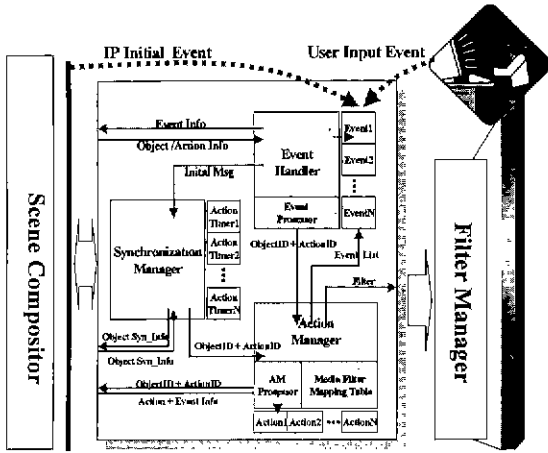


그림 4 상호작용 처리기

이르는 컴플렉스 객체들의 속성들을 모아서 프리미티브 객체의 속성을 설정한다. 설정된 프리미티브 객체의 객체 타입과 객체 정보는 액션 관리기로 전해진다. 액션 관리기는 미디어 필터 매핑 테이블을 참조하여 객체 타입에 해당하는 필터를 찾아 객체 정보를 필터에게 넘겨주고 필터는 해당 객체를 프리젠테이션 한다.

사용자에 의해서 발생한 이벤트는 이벤트 큐에 저장된다. 큐로부터 이벤트를 읽어 들인 이벤트 프로세서는 이벤트 내부에 든 정보를 바탕으로 객체 관리기에게 이벤트가 발생한 객체 아이디와 객체 타입을 요구한다. 이와 함께 이벤트 프로세서는 이벤트 매핑을 통해 현 시스템 이벤트를 프리미티브 객체가 인식할 수 있는 이벤트로 매핑 한다. 객체 관리기로부터 전해진 객체 아이디와 매핑 결과 나온 이벤트는 이벤트 프로세서가 이벤트 데이터로부터 액션 아이디를 얻는데 사용된다. 객체 관리기는 액션 아이디와 객체 아이디를 이용해 액션 테이블에서 액션 및 그에 따른 이벤트를 찾아 액션 관리기에게 전달한다. 액션 관리기는 객체에 액션을 수행한 후 객체와 객체 타입을 필터에게 넘겨주어 사용자와 상호작용한다.

#### 2.4 필터 관리기

장면 트리의 말단 노드는 원시 미디어 객체를 나타낸다. 미디어 객체에는 오디오, 비디오, 그래픽, 이미지 등의 다양한 형태의 객체들이 있다. Microsoft사의 DirectShow는 멀티미디어 스트림을 재생할 수 있는 서비스를 제공한다. 이러한 DirectShow의 멀티미디어 스트림 재생 서비스는 필터 그래프 내에 적절한 순서로 연결된 필터라는 COM 객체에 의해 지원된다 따라서 DirectShow를 사용하여 미디어 객체를 렌더링하기 위해서는 각 미디어 형식에 적합한 필터 그래프를 연결한다.

이러한 과정을 효과적으로 수행하기 위해서 미디어 필터 매핑 테이블을 구성한다 이 테이블을 참조하여 미디어 타입에 해당하는 필터 그래프의 ID를 구한다. 필터 그래프의 ID

가 구체지면, 소스 필터에 해당하는 그래프의 첫 번째 필터는 시청각 스트림으로부터 미디어 스트림을 얻는다 얻어진 스트림은 미디어 타입의 파일 형식에 따라 적절한 필터로 전달된 후, 필터 그래프에 정의된 흐름에 따라 트랜스폼 필터와 렌더링 필터를 거쳐 최종적으로 해당 출력 장치에 렌더링 된다. 그림 5는 미디어 스트림이 출력 장치에 렌더링 되는 과정을 보여준다

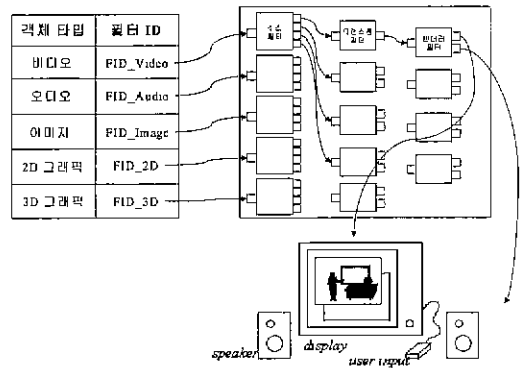


그림 5 미디어 스트림 출력 과정

### 3. 결 론

본 논문에서는 분산환경에서 MPEG-4 기반 스트림을 전송 받아서 사용자와 상호작용 가능한 장면으로 프리젠테이션 하는 엔진을 설계 하였다. 본 엔진은 멀티미디어 스트림을 파싱하여 멀티미디어 객체를 생성하고 장면을 구성하기 위해서 객체를 트리 형태로 관리한다. 전송되는 스트림을 MPEG-4에 기반함으로써 다른 환경에 대한 확장성을 제공한다 장면을 프리젠테이션할 때 객체간 동기화를 지원하고 렌더링된 장면을 통한 사용자 상호작용을 지원한다.

앞으로의 연구 방향은 제한한 엔진을 구현하고 상호작용 가능한 장면을 통한 다중 사용자의 공동작업 지원에 관하여 연구할 것이다

### 4. 참고 문헌

- [1] Atul Puri and Alexandros Eleftheriadis, "MPEG-4: An object-based multimedia coding standard supporting mobile applications", Mobile Networks and Applications, 1998.
- [2] ISO/IEC FCD 14496-1, Information technology Generic Coding of Audio-Visual Objects Part 1 Systems, ISO/IEC JTC1/SC29/WG11, 1998, 5.
- [3] ISO/IEC CD 14496-6, Information technology Generic Coding of Moving Pictures and Associated Audio Information Part 6 Delivery Multimedia Integration Framework, ISO/IEC JTC1/SC29/WG11, 1997, 10.
- [4] <http://televr.fou.telenor.no/~karlo/compositor/>
- [5] <http://drogo.csel.ste.lt/ufv/uml/>
- [6] <http://www.microsoft.com/directx/dxm/help/ds/>