

MPEG-4 동적서비스 프레임워크 설계 및 구현

이 광 의^{*}

요 약

MPEG-4 동화상은 계층적인 형태로 구조화되는 여러 개의 미디어 객체로 구성되며, 각각의 미디어 객체는 기본 스트림의 형태로 클라이언트에게 제공된다. 클라이언트 재생기는 장면 그래프라는 메타정보를 이용하여 기본 스트림들을 구성함으로써 동화상을 상영한다. BIFS 와 OD 기본 스트림이 메타정보를 전달하는데 이용된다. 우리는 동적으로 생성된 BIFS 와 OD 스트림을 사용함으로써 기존의 파일서비스와는 차별화 된 서비스를 제공할 수 있다. 예를 들면, 이미 제작된 동화상의 아래 부분에 날씨나 주식정보를 삽입할 수 있다.

본 논문에서는 동적서비스 프레임워크와 동적 서버를 제안한다. 동적서비스 프레임워크는 외부의 데이터베이스로부터 제공되는 정보에 기반 하여 BIFS와 OD 스트림을 동적으로 생성하기 위한 객체지향 프레임워크이며, 동적서버는 서버를 관리하기 위한 그래픽 사용자 인터페이스와 동적서비스를 등록하는 인터페이스를 제공한다. 이 프레임워크하에서 동적서비스는 파일서비스와 동일한 인터페이스를 가지므로 클라이언트와 다른 서비스에게 기존의 파일서비스와 동일하게 보여 진다.

Design and Implementation of An MPEG-4 Dynamic Service Framework

Kwang Eui Lee[†]

ABSTRACT

MPEG-4 movies are composed of several media objects, organized in a hierarchical fashion. Those media objects are served to the clients as elementary streams. To play the movie, client players compose the elementary streams according to the meta-information called the scene graph. The meta-information streams are delivered as BIFS and OD elementary stream. Using dynamically generated BIFS and OD streams, we can provide a service something differs from traditional file services. For example, we can insert weather or stock information into the bottom of the screen while an existing movie was played in the screen.

In this paper, we propose a dynamic service framework and dynamic server. Dynamic service framework is an object-oriented framework dynamically generating BIFS and OD streams based on the external DB information. Dynamic server provides a GUI for the server management and interface for registering dynamic services. In this framework, the dynamic service has the same interface as a file service. So, a dynamic service is considered as a file service by clients and other services.

Key words: MPEG-4, DMIF, 동적서비스

1. 서 론

낮은 대역폭을 이용한 고화질 서비스라는 목적으로 시작된 MPEG-4는 이제 사용자 상호작용이라는 목표를 향해 가고 있다. 이 두 목표를 모두를 만족시키기 위하여 MPEG-4에서는 객체기반의 압축방법

을 사용하게 되었다[1]. 객체기반의 압축방법에서는 하나의 MPEG-4 콘텐츠가 각 객체를 재생하기 위한 여러 개의 스트림들로 구성되어진다. 즉, 하나의 MPEG-4 콘텐츠 또는 MPEG-4 파일은 하나의 동화상이며, 이는 다수의 스트림들의 조합으로 구성된다[2]. 더 나가서 하나의 서비스는 다시 여러 개의 서비스로 구성될 수 있다. 즉, 여러 개의 MPEG-4 서비스가 좀더 규모가 큰 하나의 MPEG-4 서비스를 구성할 수 있다[1].

접수일 : 2002년 3월 12일, 완료일 : 2002년 6월 1일

^{*} 정회원, 동의대학교 멀티미디어공학과 전임강사

MPEG-4 콘텐츠를 구성하는 스트림들은 크게 네 종류로 분류할 수 있다. 첫 번째 종류의 스트림은 콘텐츠상의 각 객체에 대해 오디오, 비디오 또는 오디오/비디오(이하 A/V) 데이터를 포함하는 A/V 스트림으로 하나의 A/V 객체에 대하여 하나의 스트림이 대응된다. 두 번째 종류의 스트림은 BIFS (Binary Format for Scene) 스트림이다. BIFS 스트림은 장면 그래프를 구성하기 위한 데이터로 구성된 스트림으로 각 MPEG-4 객체에 대하여 상대적인 위치 등을 포함하는 다양한 메타정보로 구성된다. 세 번째 종류는 OD (Object Descriptor) 스트림으로, 이는 BIFS 장면 그래프에 나타나는 객체가 스트림을 갖는 A/V 객체인 경우 각 객체에 대한 재생방법 등의 메타정보로 구성되는 스트림이다. 마지막으로 OCI 스트림은 A/V 스트림 자체에 대한 정보를 포함하는 스트림이다[2].

기존의 MPEG-4 서비스에서 대부분의 경우 스트림들은 정적으로 이미 생성되어 제공되거나, 동적으로 제공되더라도 충분히 유연하지 못했다[3-5]. 그러나 MPEG-4의 서비스를 제공할 때 전체 스트림의 일부분만이라도 실시간에 동적으로 생성하여 제공한다면 다양한 응용에 활용할 수 있음을 예측할 수 있다. 실제로 앞서 발표된 논문인 “온라인 저작”에서의 기본 아이디어는 A/V 스트림들의 재생시간 및 위치 등의 데이터를 에이전트가 동적으로 결정하자는 것이다 [6]. 논문 6에서는 클라이언트가 특정 MPEG-4 서비스에 접근하게 되면 동일한 화면이 에이전트에게 보여지며, 또한 에이전트의 필요에 따라 BIFS 스트림을 동적으로 생성하여 클라이언트와 에이전트의 화면을 동시에 갱신할 수 있도록 하는 프레임워크를 제안하였다[6].

본 논문에서 다루는 동적서비스의 내용은 상기 [6]의 에이전트 서비스를 자동화하기 위하여 고안된 것이다. 기본적인 아이디어는 에이전트에 의하여 수행되던 동적 제어를 프로그램에 의하여 자동화하는 것이다. 동적서비스 프레임워크에서는 BIFS와 OD 스트림을 파일서비스 시에 주어진 데이터를 이용하여 프로그램으로 동적으로 생성하게 된다. 또한 이 내용은 현재 제공중인 MPEG-4 서비스(MPEG-4 파일서비스)에 자연스럽게 통합된다. 예를 들어, 현재의 날씨와 같은 정보는 미리 MPEG-4 파일로 만들 수 없으므로 날씨 정보를 제외한 모든 내용을 MPEG-4 동화상으로 구성한 후, 실제로 동화상을 서비스할 때 날씨 정보를 기본 스트림으로 생성하여 이미 구성된

MPEG-4 동화상에 포함시킬 수 있다.

2. 동적서비스 프레임워크의 설계

기본적인 MPEG-4 파일서비스 프레임워크 상에서 동적서비스를 제공하기 위한 추가적인 모듈들은 크게 세 부분으로 나누어진다. 첫 번째는 데이터 소스로서 실제로 데이터를 생성하는 부분이다. 이 부분은 어떤 데이터가 생성되는가, 또는 어떻게 데이터를 생성하는가에 종속적이므로 프레임워크에 포함될 수 없다. 따라서 본 논문에서는 이 부분을 데이터베이스로 대체 하였다. 데이터베이스로 대체 함으로서 다양한 소스에 대하여 일관되기도 유연한 인터페이스를 유지할 수 있다. 두 번째 부분은 MPEG-4 동적서비스 모듈로, 파일서비스와 동일한 인터페이스를 가지고 동적으로 생성되는 스트림을 이용하여 동적서비스를 하나의 파일서비스처럼 제공할 수 있도록 하는 부분이다. 프레임워크에서는 동적서비스를 쉽게 구현할 수 있도록 상위수준 인터페이스(interface)와 부분적인 구현을 제공하므로 프로그래머는 동적서비스의 run() 메소드 만을 구현하면 된다. 마지막 부분은 MPEG-4 동적서비스 부분으로 파일서비스를 포함하여 동적서비스를 유지 관리하는 부분이다.

그림 1은 서비스 프레임워크의 전체적인 구성을 보여주고 있다.

그림에서 서버 GUI는 서버측 관리자가 동적서비스를 등록할 수 있도록 인터페이스를 제공한다. 이때 등록되는 것은 서비스에 대한 생성객체(factory object)이다. 생성객체는 실제 서비스에 대한 요구가 있는

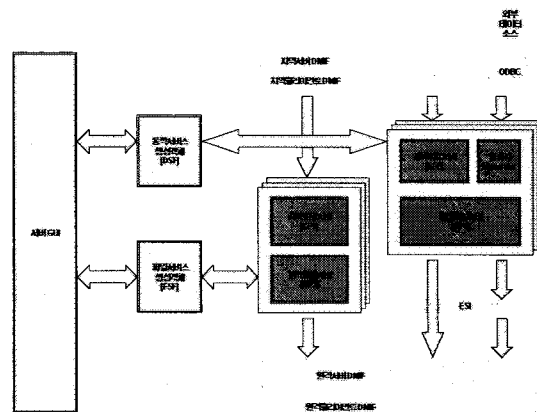


그림 1. MPEG-4 동적서비스 프레임워크

경우에 필요한 객체를 생성하여 서비스 할 수 있도록 해준다. 파일서비스와 동적서비스에서 원격서버객체는 클라이언트와 통신을 위하여 사용되며, 지역서버객체는 지역파일시스템에 접근하기 위한 객체이다. 동적서비스에서 생성자(generator)는 데이터베이스로부터 읽은 데이터를 MPEG-4 명령어로 변환하는 역할을 수행하며, 이렇게 변경된 데이터를 전송하기 위하여 표준의 ESI (Elementary Stream Interface)를 이용한다.

본 논문에서 다루는 동적서비스는 기본적으로 MPEG-4 서비스이므로, 클라이언트 입장에서는 MPEG-4 표준에서 권고하는 파일서비스와 동일한 절차를 통하여 서비스되어야 한다. 따라서 동적서비스는 파일서비스와 동일한 외부 인터페이스를 가져야 하며, 이에 따라 MPEG-4의 시스템 표준에 의하여 서비스되어야 한다[2]. 다음의 소절에서는 파일서비스 과정, 동적서비스 방법 및 동적서비스 프레임워크, 데이터베이스 구성 방법 등에 대하여 그림 1을 기준으로 설명한다.

2.1 파일서비스 접근절차

본 논문에서는 모든 프로그램이 이미 구현된 DMIF (Delivery Multimedia Integration Framework)를 기반으로 하여 작성되는 것을 가정한다[8]. DMIF는 MPEG-4 표준의 여섯 번째 부분으로 네트워크 상에서 MPEG-4 데이터를 전달하는 방법에 대한 표준이다. DMIF는 네트워크, 지역, 그리고 방송(broadcast)되는 스트림에 대하여 투명한 접근을 허용하며, 프로그래머는 DMIF 인터페이스를 이용하여 하위 네트워크에 독립적인 프로그래밍을 할 수 있게 된다. DMIF는 DAI (DMIF Application Interface)와 DNI (DMIF Network Interface)로 구성된다. 이중 실제로 사용자에게 보여지는 인터페이스는 DAI이다. DAI는 서비스 접속과 해제를 위한 인터페이스와 채널을 추가, 삭제하기 위한 인터페이스, 사용자 명령어와 데이터를 전송하기 위한 인터페이스 등을 제공한다. 인터페이스를 보면, 사용자 응용프로그램이 DMIF를 호출하는 여섯 개의 인터페이스

```
DA_ServiceAttach ()
DA_ServiceDetach ()
DA_ChannelAdd ()
DA_ChannelDelete ()
```

```
DA_UserCommand ()
DA_Data ()
```

와 DMIF가 사용자 응용프로그램에 콜백(callback)해주는 대응되는 여섯 개의 인터페이스

```
DA_ServiceAttachCallback ()
DA_ServiceDetachCallback ()
DA_ChannelAddCallback ()
DA_ChannelDeleteCallback ()
DA_UserCommandCallback ()
DA_DataCallback ()
```

등 총 12개로 구성된다[7].

표준에서 권고하는 MPEG-4 파일서비스의 과정은, 설명의 편의를 위하여 다음의 세 개의 기본적인 단계와 하나의 추가적인 단계로 구분이 가능한데, 그 첫 번째 단계는 서비스 제공자가 제공하고자 하는 MPEG-4 서비스(이하 서비스라 하며, 실제로 하나의 파일을 의미한다)를 MPEG-4 서버(이하 서버)에 등록하는 단계로 그 구체적인 방법은 표준에서 제공하지 않는다[2].

두 번째 단계는 MPEG-4 클라이언트(이하 클라이언트)가 서비스에 접속하는 단계이다. 먼저 클라이언트는 서비스 URL을 이용하여 서버에 접근한다. 서버는 클라이언트에서 요청한 파일에서 IOD (Initial Object Descriptor) 부분을 읽은 후 클라이언트에 전송한다. 이때 IOD는 두 가지 타입이 있을 수 있는데, 클라이언트에 실제 서비스를 제공하기 위한 BIFS 스트림과 OD 스트림에 대한 OD를 포함하는 경우와 서비스를 재지정하기 위하여 새로운 URL을 포함하는 경우가 그것이다. 뒤의 방법을 URL 재지정 기법이라 한다. DMIF 기반의 응용프로그램(또는 서비스)에서는 이 URL 재지정 기법을 이용하여 다양한 기능을 수행 할 수 있다.

마지막 단계는 클라이언트가 서비스를 받는 단계이다. 클라이언트는 IOD의 내용을 이용하여 서버에 BIFS 스트림과 OD 스트림을 위한 채널을 요청한다. 채널이 설치되면 서버는 두 개의 채널을 통하여 BIFS 스트림과 OD 스트림을 전송한다. 클라이언트는 BIFS 스트림과 OD 스트림의 내용을 기반으로 화면을 구성하며, 추가적인 스트림이 필요한 경우 앞의 과정과 동일한 과정을 거쳐서 채널을 요청하고 스트

림을 전송 받는다. 서비스가 종료되면, 채널 설정을 해제하고 서비스 접속을 해제한다.

추가적인 단계는 하나의 MPEG-4 서비스가 두 개 이상의 파일로 구성되어 하나의 서비스가 다른 서비스를 포함하고 있는 경우, 즉 MPEG-4 동화상의 일부가 다른 동화상으로 구성되어 있는 경우에 적용된다. 이 단계에서 클라이언트 프로그램은 두 개의 서비스 중 부분으로 포함된 서비스를 하나의 새로운 서비스로 간주하여 앞의 단계에 따라서 서버에 요청한 후 두개의 서비스를 하나로 통합하여 사용자에게 보여지게 된다.

그림 2는 파일서비스시의 동적인 객체의 구성과 순서를 보여 준다.

그림은 초기에 서비스 등록의 과정과 클라이언트에서 DA_ServiceAttach() 함수를 통하여 서비스 접근이 이루어졌을 때의 상황이다. 그림에서 보여지지는 않지만, 다음 단계로 원격파일서비스는 지역파일서비스로부터 데이터를 받아서 원격지의 클라이언트에게 데이터를 전송한다. 마지막으로 파일서비스 생성객체에 DA_ServiceDetach() 명령이 전달되면, 파일서비스객체는 삭제되고, 서비스는 종료된다. 이때 파일서비스 생성객체는 계속 남아서 다음 서비스를 준비한다.

2.2 동적서비스 방법

동적서비스는 클라이언트가 자신의 코드에 대한 변경 없이 접근할 수 있도록, 클라이언트에 대해서 파일 서비스와 동일한 인터페이스를 가져야 한다. 또한 동적서비스의 모든 내용을 실시간에 구성하여 제공하는 것은 비효율적이므로, 기본적인 파일서비스 위에 동적 서비스를 구현한다. 즉, 클라이언트는 하나의 파일 서비스를 받은 후, 동적으로 적당한 갱신 명령어를 통하여 동적으로 변하는 서비스를 제공받게 되는 것이다.

이를 위하여 동적서비스는 파일서비스를 상속받아서 구현되는데, 먼저 파일서비스와 동일한 절차에 따라서 동적서비스에 연결된 파일서비스를 받는다. 파일서비스가 모두 종료되면, 기존의 파일서비스에서는 연결을 끊고 새로운 서비스를 위하여 자원을 할당하지만, 동적서비스에서는 이때부터 실질적인 서비스를 시작하게 된다. 실제로 앞의 파일서비스 단계에서 동적서비스를 위한 초기 동화상이 전송된다.

이후의 동작순서는 파일서비스와 유사하며, 기본적인 파일서비스가 수행된 후, 생성자에 의하여 동적인 데이터가 전송된다.

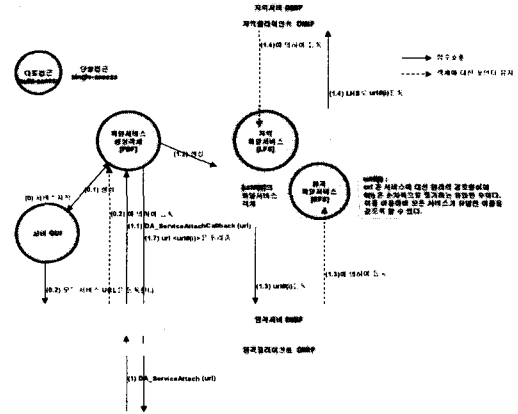


그림 2. MPEG-4 파일서비스 절차

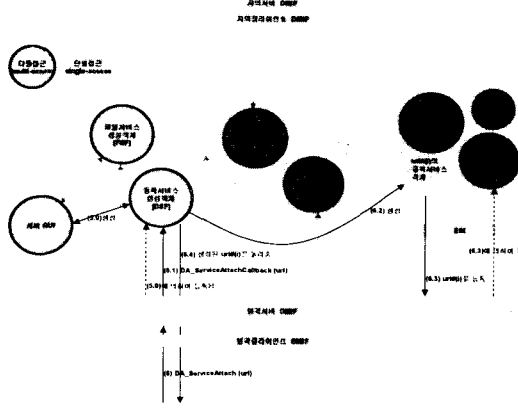


그림 3. MPEG-4 동적서비스 절차

동적으로 생성되는 데이터는 다음절의 형식에 따라서 실시간으로 데이터베이스에 저장된다. 이렇게 저장된 데이터베이스의 내용을 동적서비스 모듈이 읽어서 다음의 세 종류의 갱신 명령어를 생성한다.

- replace_visual : 비주얼 객체를 갱신 (type1)
- replace_text : 문자열의 내용을 갱신 (type2)
- replace_color : 비주얼 객체 색 변경 (type3)

실제로 이러한 명령어는 BIFS 갱신 명령어와 OD 갱신 명령어를 필요로 하는데 이를 생성하기 위하여 논문[9]의 라이브러리를 이용하였다.

2.3 데이터베이스 저장 형식

본 논문에서는 데이터베이스에 저장하는 부분과 저장된 데이터를 이용하여 서비스하는 부분을 분리하기 위하여 데이터베이스에 저장하는 방식에 대한

형식을 정의하였다. 먼저 데이터베이스는 등록된 서비스를 통합관리하기 위한 하나의 테이블을 갖는다. 각 필드의 의미는 다음과 같다.

```
table service {
type_service      :서비스의 종류. 예를 들면,
                   Weather 또는 Stock.
refresh_rate      :데이터의 갱신속도
}
```

또한 각 서비스 당 하나의 데이터 테이블을 갖게 되는데 그 양식을 모두 동일하다. 다음은 실시간 주식 현황 서비스를 위한 Stock 서비스의 테이블이다.

```
table stock {
type              :연산의 종류를 정의한다.
myID              :생성 또는 갱신될 객체의 ID
pID               :myID 객체의 부모 객체의 ID
xPos              :생성 또는 갱신될 객체의 x 좌표, 또는
                   type 1연산의 경우 red_color 값
yPos              :생성 또는 갱신될 객체의 y 좌표, 또는
                   type 1연산의 경우 green_color 값
width             :생성 또는 갱신될 객체의 width, 또는
                   type 1연산의 경우 blue_color 값
height            :생성 또는 갱신될 객체의 height
data              :연산에서 사용될 데이터. type 0의 경우
                   text, type 2의 경우 변경될 새 객체 이름
}
```

3. 동적서버의 기능 및 동적서비스 프레임워크의 구현

동적서버 및 동적서비스 프레임워크는 Pentium III, Windows2000 환경에서, 이미 구현된 DMIF 라이브러리를 사용하여 구현되었으며[8], 동적인 데이터의 생성은 확장성을 위하여 마이크로소프트사의 액세스 데이터베이스로부터 데이터를 추출하여 필요한 BIFS 명령어를 생성하도록 하였고, BIFS 명령어의 생성을 위하여 BFIS 변환 명령어 생성모듈을 이용하였다[9]. 또한 윈도우즈 환경에서 수행되도록 하기 위하여 MFC에서 제공되는 프로젝트를 이용하여 생성하였으나, 이식성을 고려하여 실행에 관련된 코드는 모두 C++ 표준 라이브러리를 이용하여 작성되었다. 다음 소절에서는 실제 구현내용에 관련하여 필요한 부분을 보충 설명한다.

3.1 데이터베이스의 구현

데이터베이스는 ODBC를 이용하여 앞에 정의된 형식으로 구현하였으며, 정상적인 동작을 확인하기 위하여 프로그램에서 가상의 데이터를 생성하고, 이를 데이터베이스에 저장하는 프로그램은 VC++의 데이터베이스 응용 프로그램을 생성하였다.

3.2 동적서버의 구현

동적서버는 윈도우 응용프로그램으로 윈도우 메시지를 처리하는 부분과 서비스를 처리하기 위한 서비스메니저 부분으로 나누어져 있다. 실제 서버에 해당하는 부분은 서비스메니저로 역시 DMIF 응용프로그램이다. 서비스메니저를 통하여 동적서비스를 제공하는 단계는 다음과 같다.

3.3 동적서비스 프레임워크의 구현

동적서비스는 클라이언트 입장에서 파일서비스와 동일한 인터페이스를 가져야 하며, 단지 동적으로

알고리즘 동적서비스

단계 1 : 서비스 제공자가 동적서버에 동적서비스를 등록한다 : 현재 프로그램에 하드코딩 되어 있으나, 동적서비스를 동적으로 등록할 수 있도록 하기 위하여 필요한 인터페이스를 제공할 예정이다.

단계 2 : 클라이언트가 등록된 동적서비스에 접근하면, 서버는 서비스에 필요한 URL을 생성하여 되돌려준다 : 클라이언트는 실제로는 동적서비스 서버에 접근하게 되며, 동적서비스 서버는 factory method를 이용하여 접근된 동적서비스를 생성하고, 생성된 서비스에 대한 유일한 이름을 지정한다. 다음 지정된 이름을 이용하여 새로운 URL을 구성하고, IOD 재지정 방법에 따라서 클라이언트에게 돌려준다.

단계 3 : 클라이언트는 표준에 따라 새로운 URL에 접근을 시도하여, 동적서비스를 받게 된다 : 이 단계에서 채널설정 등의 메카니즘은 파일서비스와 동일하다

단계 4 : 클라이언트가 서비스를 종료하면, 동적서비스도 종료된다 : 동적서버는 이 사실을 DMIF를 통하여 탐지한 후 동적서비스를 종료한다.

스트림이 생성되는 부분만 다르므로 동적서비스 프레임워크를 이용하는 서비스 제공자가 스트림을 생성하는 부분만 고려할 수 있도록 작성하였다.

본 프레임워크에서는 동적서비스를 위한 상위 클래스로 DynamicService라는 클래스를 구현하였는데, DynamicService 클래스는 서비스 접근, 채널 추가 삭제 등 스트림 생성에 직접적으로 필요하지 않은 부분은 모두 파일서비스에서 상속받았으며, 실제 스트림(일반적으로 BIFS 스트림)을 생성하는 부분만을 run() 메소드로 남겨 놓았다. 또한 BIFS와 OD 채널, BIFS 스트림을 생성해주는 위한 객체에 대한 인터페이스 등이 클래스 멤버 변수로 제공된다. 동적서비스를 제작하는 프로그래머는 DynamicService 클래스를 상속받은 후 run() 메소드를 재정의(overriding)함으로써 동적서비스를 생성할 수 있는데, 방법은 스트림생성 인터페이스를 이용하여 생성된 스트림을 BIFS채널로 전송하는 것이다.

4. 결 론

동적서비스 프레임워크는 MPEG-4의 기본 구성 요소인 오디오/비디오 스트림들을 동적으로 재배치하거나, BIFS 스트림을 동적으로 생성하는 프로그램을 간단히 생성할 수 있도록 제공되는 객체 기반 프레임워크로 실시간에 BIFS 스트림을 생성하는 작업 이외의 모든 절차는 프레임워크가 담당한다. MPEG-4 동적서비스를 생성하기 위하여 프로그래머는 제공되는 DynamicService 클래스를 상속받은 후 BIFS 명령어를 동적으로 생성하도록 run() 메소드를 재정의 한다. 프로그래머는 동적으로 생성되는 BIFS 명령어를 이용하여 오디오/비디오 스트림들을 실시간에 재배치하거나 지오메트리 객체를 제어할 수 있다.

현재 전체적인 프레임워크에서 데이터베이스에 자료를 동적으로 등록하기 위한 부분이 구현되어 있지 않다. 그러나 자료를 등록하는 부분은 응용에 종속적인 것으로 예상되므로 구현 가능성에 대해서는 좀 더 검토되어야 한다. 또한 사용자를 위하여 동적서비스 생성을 프로그램 수준이 아닌 저작도구 수준에서 제공할 필요가 있다.

참 고 문 헌

[1] Rob Koenen, Fernando Pereira, Leonardo Chi-

ariglione, "MPEG-4: Context and objectives," *Signal Processing: Image Communications*, vol. 9 pp. 295-304, 1997.

[2] ISO/IEC JTC1/SC29/WG11 FCD 14496-1, Information Technology Generic Coding of Audio-visual Objects Part 1: Systems, Mar. 1998.

[3] ISO/IEC JTC1/SC29/WG11 M311, Zvi Lifshitz, "APIs for System Software Implementation," Nov. 1997.

[4] J. Deicke, U. Mayer, M. Glesner, "A Client/Server application as an example for MPEG-4 systems," *Computer Communicatins*, vol. 21, pp. 1302-1309, 1998.

[5] Yong He, Ishfaq Ahmad, Ming L. Liou, "Real-Time Interactive MPEG-4 System Encoder Using a Cluster of Workstations," *IEEE Trans. On Multimedia*, vol. 1, no. 2, pp. 217-233, June 1999

[6] 이광의, 민옥기, 이범식, 지동해, "MPEG-4 온라인저작을 위한 서비스 프레임워크," NCS'99, Dec. 1999.

[7] ISO/IEC JTC1/SC29/WG11 FCD 14496-4, Information Technology Generic Coding of Audio-visual Objects Part 4: Delivery Multimedia Integration Framework, Mar. 1998.

[8] 이광의, 민옥기, 이범식, 지동해, "DirectX를 기반으로 하는 DMIF의 설계," 제12회 산학연 멀티미디어학술대회 학술논문집, Nov. 1998.

[9] 민옥기, 이광의, 이범식, 지동해, "BIFS 변환 명령어 생성모듈 설계 및 구현," NCS'99, Dec. 1999.



이 광 의

1990년 서강대학교 컴퓨터학과 (공학사)
 1992년 서강대학교 컴퓨터학과 (공학석사)
 1997년 서강대학교 컴퓨터학과 (공학박사)
 1997.12-2001.2 한국전자통신연구원 선임연구원

2001.3-현재 동의대학교 멀티미디어공학과 전임강사
 관심분야 : 멀티미디어 시스템 및 응용, 저작도구, 계산이론