

# 실시간 멀티미디어 프리젠테이션 및 콘텐츠 제작을 위한

## 통합 시스템 설계

양종운 · 이규남 · 나인호

군산대학교

### A Design of Integrated System for Real Time Multimedia Presentation and Content Authoring Tool

Jong-un Yang · Kyu-nam Lee · In-ho Ra

Kunsan National University

E-mail : ihra@ks.kunsan.ac.kr

#### 요약

본 논문에서는 윈도우즈 시스템에서 네트워크를 통한 멀티미디어 콘텐츠 제작 및 프리젠테이션을 위한 통합 시스템 구축을 위한 설계 방법에 대해 제안하였다. 특히, 오디오, 비디오와 같은 연속 미디어를 비롯한 다양한 미디어 데이터를 하나의 생성 도구를 통해 제작 및 편집하기 위한 통합형 편집기 설계 방안에 대해 기술하였다. 또한, 네트워크를 통해 전송된 멀티미디어 데이터를 프리젠테이션 하기 위해 필요한 버퍼링 및 시나리오 기반 프리젠테이션 기법 설계 방안과 멀티미디어 편집기와 프리젠테이션 기능을 유기적으로 통합하기 위한 시스템 모델을 제시하였다. 마지막으로 멀티미디어 프리젠테이션 과정에서 QoS를 유지하기 위해 필요한 프리젠테이션 제어, 동기화 제어, 사용자 입력 제어 등의 요구를 효율적으로 처리하기 위한 이벤트 모니터링 기반 스레드 처리 기법을 제안하였다.

#### ABSTRACT

This paper presents methods to design an integrated system for supporting both multimedia contents creations and its presentation in a windows system. Especially, it describes methods for designing an integrated editor that can be used to create and edit various media data including continuous media such as audio and video. And it also proposes a system model for systematically integrating multimedia creating tool with multimedia presentation system where the buffering and scenario-based presentation methods are included in the propose system for supporting effective multimedia presentation through a network. Finally, we describe a threads processing technique based on event monitoring to satisfy effectively needs for presentation control, synchronization control, and user's input control during a multimedia presentation.

#### 1. 서론

일반인을 대상으로 한 고기능 저가의 멀티미디어 컴퓨터의 보급과 인터넷을 비롯한 네트워크의 대중화는 멀티미디어 기술을 다각화시키고 있다. 단순한 콘텐츠 제작에 머물렀던 기존 멀티미디어 기술은 현재 네트워크를 통하여 주문형 비디오 (Video On Demand), 멀티미디어 타이틀, 원격 교육, 원격 의료 시스템, 화상회의, 인터넷 교육 등과 같은 사용자 요구에 부응하는 여러 형태의 멀티미디어 데이터를 처리 및 관리할 수 있는 제작 도구가 연구 개발되고 있다.

멀티미디어 데이터는 텍스트나 이미지처럼 시간의 흐름에 독립적인 이산 미디어(discrete Media)와 시간의 흐름에 따라 연속적으로 변화되는 오디오 데이터나 비디오 데이터 등과 같은 연속 미디어(continuous Media)를 의미한다. 이러한 두 부류의 미디어를 프리젠테이션에 응용하기 위해서는 미디어간의 시간과 공간적 요소를 고려해야 한다[1][2][3].

논문의 구성은 다음과 같다. 먼저, II장에서는 설계목표에 대해 기술하였고, III장 및 IV장에서는 본 연구에서 설계된 통합 시스템의 전체 구조를 비롯한 각 부의 특징에 대하여 기술하였다. 마지막으로 V장에서는 향후 연구 방향에 대하여 기술하였다.

본 연구는 정보통신부 2000년도 대학기초 지원 사업에 의해 수행되었습니다.

### II. 설계 목표

멀티미디어 저작 도구는 크게 정적인 문서에 멀티미디어 데이터를 삽입함으로써 하나의 통합된 정적 문서를 저작하기 위한 도구와 각 미디어들간의 시간 및 공간적 컴포지션을 통해 멀티미디어 시나리오를 구성하고 이것을 통해 여러 사용자에게 프리젠테이션을 수행할 수 있도록 지원하는 멀티미디어 프리젠테이션용 저작 도구가 있다[4].

본 연구는 멀티미디어 프리젠테이션용 저작 도구에 해당하며, 멀티미디어 저작 시스템의 특성을 살려 다양한 형태의 미디어를 편집 및 저작할 수 있는 편집기를 제작하여 프리젠테이션을 위한 시나리오 구성에 이용하고, 원격 및 지역 저장 장치에 저장되어 있는 멀티미디어 데이터 처리뿐만 아니라 캠코더나 오디오 입력장치와 같이 실시간에 생성되는 멀티미디어 데이터를 공유 자원으로 취급하여 프리젠테이션에 사용될 수 있도록 설계하였다. 특히, 본 시스템은 프리젠테이션 실행 도중에 사용자가 프리젠테이션 되는 내용을 일시 정지, 종료, 빠른 탐색, 역 탐색, 일정 구간 반복 등과 같은 행위로서 프리젠테이션 수행에 직접 개입할 수 있는 사용자 개입이 가능한 프리젠테이션용 저작 도구를 구현하는데 중점을 두었다[5].

### III. 시스템 설계

본 연구의 실시간 멀티미디어 프리젠테이션 및 콘텐츠 제작을 위한 통합 시스템은 그림 1과 같이 미디어 편집부, 시나리오 제작부, 프리젠테이션 제어부, 네트워크 동기화부, 출력 동기화부로 크게 다섯 부분으로 설계하였고 각 부분의 설계 이슈는 다음과 같다.

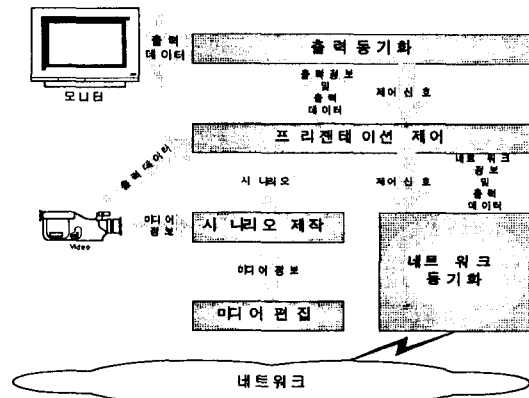


그림 1. 시스템 구조

#### 1. 미디어 편집부

미디어 편집부는 텍스트 편집기, 그래픽 편집기, 사운드 편집기 및 플레이어, 동영상 편집기

및 플레이어로 구성하였다. 그래픽 편집기는 BMP, JPEG와 같은 그래픽 데이터를 편집기를 이용하여 제작 및 편집하는데 사용된다. 그래픽 편집기는 그리기에 대한 객체화 처리 기능과 이미지 프로세싱 기능을 통합하였다.

동영상 편집기는 동일한 비디오 데이터나 이중의 비디오 데이터에 대하여 프레임간 블록 지정 방식을 통하여 프레임 단위의 이동과 삭제, 삽입 기능을 수행한다. 하나의 동영상 프레임에 대한 이미지 조작은 그래픽 편집기를 사용하여 처리하며 이것을 독립된 그래픽 파일로 저장할 수 있다. 시작 프레임과 종료 프레임을 사용자가 지정하여 임의의 지점까지 실행하기와 역 실행, 일정한 간격을 두고 프레임을 건너 띄어 실행할 수도 있다.

이러한 데이터 편집부의 다양한 기능을 통해 사용자는 자신이 원하는 프리젠테이션을 수행하기 위한 새로운 시나리오를 구축할 수 있고, 추가적인 조작을 통해 기존의 시나리오를 쉽게 변경할 수 있도록 설계하였으며 다양한 형태의 멀티미디어 콘텐츠를 저작할 수 있도록 하였다.

#### 2. 시나리오 제작부

멀티미디어 데이터를 프리젠테이션하기 위해서는 각 미디어 데이터에 대한 시공간적 관계를 정의하여 하나의 멀티미디어 데이터로 결합하여야 한다. 이렇게 미디어 데이터들을 시공간적으로 결합하는 행위를 멀티미디어 컴포지션(Composition)이라고 하며, 시간 컴포지션과 공간 컴포지션으로 나눌 수 있다. 시간 컴포지션은 프리젠테이션에 포함될 미디어 데이터들 간의 시간적 관계를 설정하는 것이며, 공간 컴포지션은 각 미디어들이 디스플레이 공간에 출력되는 위치와 형태를 지정하는 것을 의미한다[3][4][5].

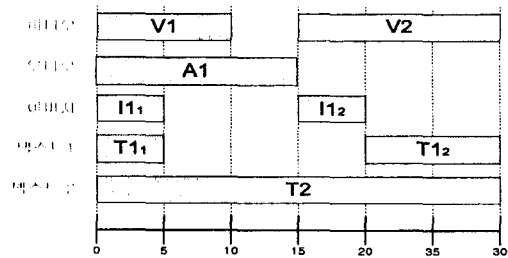


그림 2. 프리젠테이션 시나리오 예

그림 2는 프리젠테이션에 사용할 각 미디어 데이터에 대한 시간 관계를 표현한 것이다. 여기서 V1, A1, I1, T1, T2는 동일한 시간에 플레이해야 하는 미디어이며, V2, T1, T2는 동시에 끝나야 하는 미디어들을 나타낸다. 또한, T2는 프리젠테이션 과정 전반에 걸쳐 출력되지만 나머지 미디어들은 중간에 시작하거나 종료하고 있음을 알 수 있다. 이러한 시간관계 및 공간관계를 효율적으로 시각화하기 위하여 그림 3과 같은 프레임웍을 설계하였다[3][5].

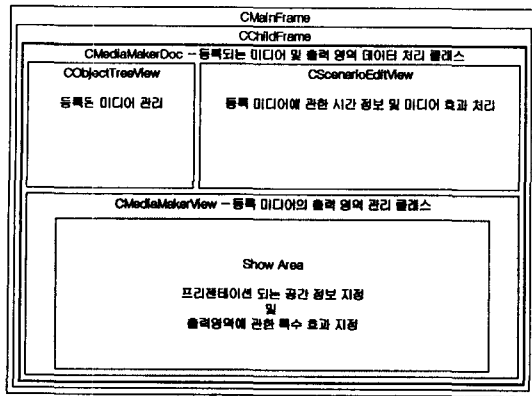


그림 3. 시나리오 제작 프레임 위

공간 킴포지션과 시간 킴포지션에 사용되는 윈도우 등록 미디어에 대하여 유일한 키에(Key)에 의해 운용되는 Hashing 구조의 저장 클래스를 사용하였으며, 그림 4는 미디어의 등록과 제거에 사용되는 키의 구조를 나타낸 것이다.

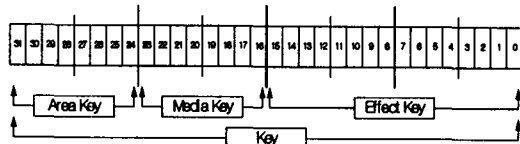


그림 4. 등록 객체에 대한 Key 구조

키로 사용되는 데이터는 32비트 DWORD를 사용하였으며, 논리적으로 하나의 출력 윈도우에 256개의 미디어와 각 미디어에 65536개의 논리적인 효과를 지정할 수 있도록 설계하였다. 이러한 구조를 사용함으로써 유일한 키를 생성할 수 있으며, 출력 영역에 따른 미디어 및 해당 효과를 제어하는데 효율성을 제공하게 된다.

3. 네트워크 동기화부

사용자에게 다양한 멀티미디어 데이터의 처리 및 전송, 고품질 서비스를 제공하기 위해서는 네트워크를 통해 각각의 미디어들이 시나리오에 정의된 시공간적 관계에 따라 동기적으로 수신 시스템으로 전송되어야 한다. 또한, 전송된 미디어 데이터를 제시간에 출력하기 위한 출력 동기화가 반드시 수행되어야 연속 미디어 데이터를 끊기는 현상 없이 연속적으로 보여줄 수 있다[5][6][7].

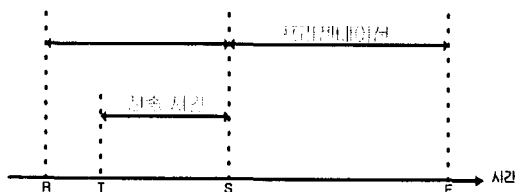


그림 5. 프리젠테이션의 시간 관계

그림 5는 미디어 간 동기화를 위해 만족되어야 할 미디어 검색시간, 네트워크 전송지연시간, 프리젠테이션 출력시간 및 지속시간 간의 기본적인 관계를 보인 것이다. 여기서 R은 미디어 검색 시작 시간, T는 정의 및 검색을 완료하고 전송을 시작하는 시간, S는 프리젠테이션 시작 시간, E는 프리젠테이션의 종료 시간을 의미한다. 어떤 미디어가 정해진 시간에 정확하게 프리젠테이션을 수행하기 위해서는 해당 미디어 데이터의 검색 시점 R은  $R \leq S - T$  조건을 만족해야 한다. 이 조건은 어떤 미디어 데이터가 프리젠테이션하기 위한 목적지에 도착하기 전에 프리젠테이션 될 수 없다는 것을 의미하며, 이러한 조건 만족을 위해서는 T와 관련된 요소들을 고려하여 정확한 검색 시간을 계산하여야 한다. 본 연구에서는 이러한 네트워크의 동기화를 구현하기 위하여 그림 6과 같은 구조를 설계하였다.

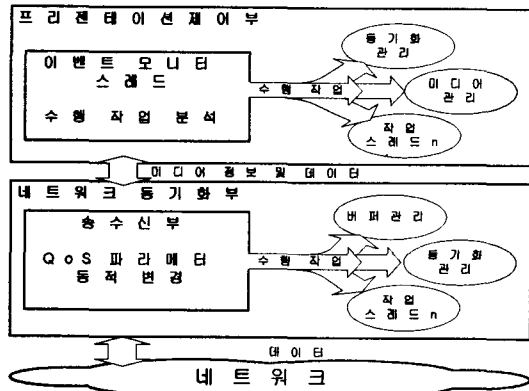


그림 6. 네트워크 동기화부 구조

또한 본 연구에서는 네트워크를 통한 멀티미디어 프리젠테이션 데이터에 대한 효율적인 버퍼링을 위하여 다음과 같은 기능을 갖는 다중 스트림 버퍼링 시스템을 설계하였다.

- ① 멀티미디어 응용의 서비스 품질에 따라 여러 가지 서비스 등급으로 분류하고 이에 따라 필요한 만큼의 버퍼를 할당하는 기능.
- ② 데이터 소비율에 따라 버퍼 관리를 윈도우 단위로 수행하는 버퍼 관리 기능.
- ③ 단위 시간에 소비되는 블록량을 반영하여 데이터 소비율에 따라서 버퍼를 동적으로 할당하는 기능.
- ④ 각 미디어 스트림에 대한 초기 버퍼 요구량을 계산하고 미디어 스트림 데이터가 전송되는 도중이라도 버퍼 요구량을 동적으로 변경할 수 있는 버퍼 요구량 계산 기능.
- ⑤ 디스크 지연시간을 줄이고 버퍼 이용률을 높이기 위한 선반입 기능.
- ⑥ 사이클 단위로 스트림 요구를 처리하여 연속 미디어의 주기적인 재생특성을 이용하여 멀티미디어 서버가 서비스 사이클에 따라서 병행

다중 스트림을 서비스할 수 있도록 지원하는 기능.

- ⑦ 각 버퍼 요구 사이클에서 필요로 하는 실질적인 데이터 소비량에 따라 데이터를 검색하는 기능.
- ⑧ 버퍼의 분할 및 합병 기법을 적용하여 버퍼 요구량을 최소화하는 기능.
- ⑨ VBR 형태의 데이터 스트림을 균등한 크기의 버퍼 슬롯에 할당하고 회수하는 기능.

그림 7은 이러한 기능을 수행하기 위한 다중 스트림 버퍼링 시스템 구조를 나타낸 것이다.

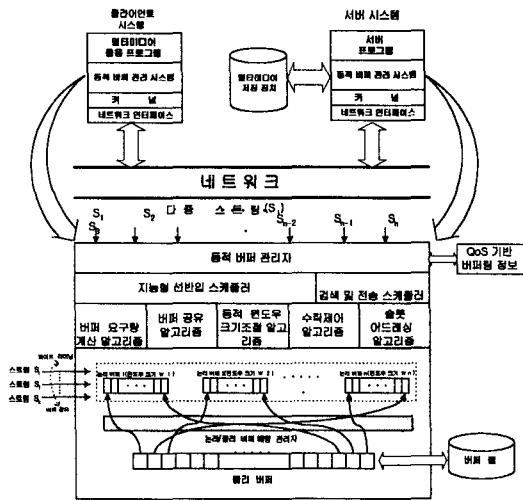


그림 7. 다중스트림 버퍼링 시스템 구조

그림 7에서는 크게 지능형 선반입 스케줄러, 검색 및 전송 스케줄러, 논리/물리 버퍼 매핑 관리자로 구성된다. 여기서 선반입 스케줄러는 멀티미디어 응용프로그램에서 요구한 서비스 품질 조건을 만족시킬 수 있도록 선반입 계획(prefetching plan)을 작성하여 데이터를 미리 읽어 들인다.

선반입 스케줄러는 멀티미디어 응용에게 일종의 메타 인터페이스를 제공한다. 이것은 의미적으로 볼 때 파일 시스템의 I/O 인터페이스가 변경되지 않기 때문에 선반입 스케줄러의 인터페이스를 메타 인터페이스라고 할 수 있는 것으로서 I/O 요구에 대한 서비스 제공 방법을 변경할 수 있다는 장점을 제공한다[3][8][9][10]. 또한, 선반입 스케줄러는 응용프로그램이 가지고 있는 데이터 접근에 대한 패턴과 시간정보(timing)를 가지고 데이터가 요구된 시간에 시스템 버퍼에 확실히 저장될 수 있도록 보장하기 위해 필요하며 비록 요구된 데이터가 저장장치에 있더라도 여러 가지 정보를 가지고 요구된 데이터에 고속으로 접근할 수 있는 기능을 제공하는데 사용된다[11][12][13].

검색 및 전송 스케줄러는 멀티미디어 데이터의 검색, 전송, 출력 등에 대한 정보를 가지고 주어진 시간내에 여러 개의 미디어 스트림을 동기화

하여 프리젠테이션할 수 있도록 멀티미디어 데이터의 검색 및 전송 스케줄을 미리 작성하고 네트워크 작업부하의 변동에 따라 발생하는 지터나 스쿠의 영향을 최소화하기 위해 필요에 따라 미리 작성된 스케줄을 네트워크 상황에 따라 동적으로 재구성할 수 있도록 해준다.

논리/물리 버퍼 매핑 관리자는 물리적으로 1차원 배열인 시스템 버퍼를 논리적으로 분할하여 여러개의 독립적인 버퍼로 분할하고 이것들을 동일한 생산/소비 시간주기를 갖는 여러 미디어 스트림이 파이프라이닝 방식으로 공유할 수 있도록 해준다. 또한, 물리 버퍼의 분할 및 합병이 용이하도록 실제의 물리 버퍼 주소 공간에 독립적으로 논리 버퍼가 동적으로 확장 및 축소될 수 있도록 다수의 논리 버퍼와 하나의 물리 버퍼 간의 매핑을 수행한다.

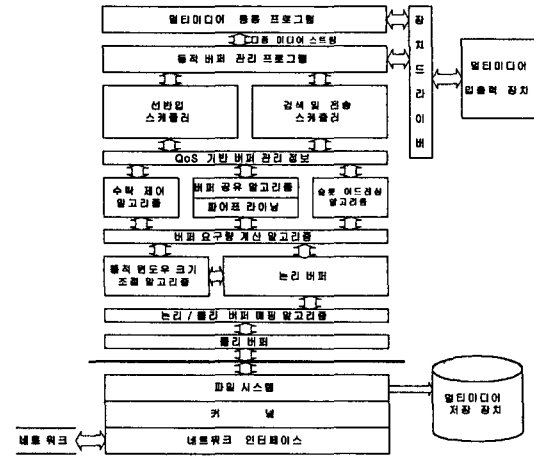


그림 8. 다중스트림 버퍼 시스템의 모듈 구조

그림 8은 다중스트림 버퍼링 시스템상에서 작동하는 서비스 모듈 및 알고리즘의 종류와 모듈 간 인터페이스를 나타낸 것이다.

#### 4. 프리젠테이션 제어부

현재 국내외적으로 사용자 참여를 허용하는 멀티미디어 프리젠테이션 기법에 대한 연구는 아직 초보적인 단계에 있으며, 실용화에 직접 적용할 수 있는 기술은 아직 개발되어 있지 않다. 프리젠테이션 제어부에서는 구축된 시나리오를 기반으로 프리젠테이션을 정확히 주어진 시간 관계에 따라 수행하기 위한 전반적인 제어를 담당한다. 고품질의 프리젠테이션을 유지하기 위해서는 시스템 자원(메모리, 버퍼, 입출력장치)과 네트워크 자원(소켓, 채널, 전송지연) 등과 같이 비동기화를 유발하는 여러 가지 요소들을 체크하여 실시간 멀티미디어 데이터 처리에 필요한 자원을 유지하여야 하며, 현재의 자원 보유량과 프리젠테이션 품질 유지에 필요한 자원 요구량 변동에 따라 프리젠테이션을 위한 데이터 검색, 전송, 버퍼링, 출력의 속도 및 데이터량을 조절하여야 한다.

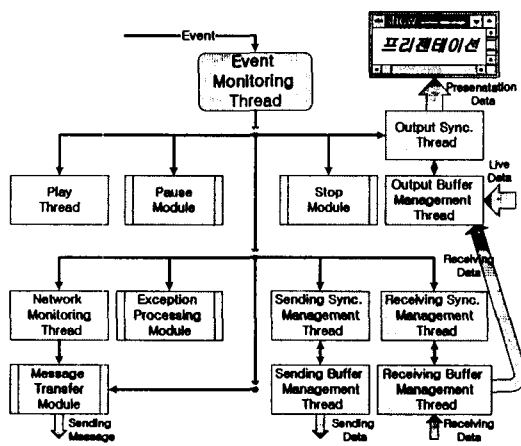


그림 9. 프리젠테이션 제어부 구조

프리젠테이션 제어는 두 가지로 구분된다. 하나는 사용자 인터페이스 부분이고, 나머지 하나는 프리젠테이션 품질 유지를 위한 프리젠테이션 전반에 걸친 제어부분이다. 사용자 인터페이스 부분은 사용자의 프리젠테이션 실행, 일시 정지, 멈춤, 연속 재생과 같은 실시간 참여에 대하여 사용자 인터페이스를 통한 제어가 이루어진다. 내적인 프리젠테이션 제어는 사용자 개입에 의해 입력된 명령과 출력 동기화, 그리고 네트워크 동기화를 위한 조건들의 조합으로 구성되며 모든 스레드 제어가 이러한 방식으로 수행된다. 그림 9는 프리젠테이션 제어부의 구조를 나타내고 있다.

사용자 참여, 출력 동기화, 네트워크 동기화의 세 가지 요소는 프리젠테이션을 위한 QoS의 유지와 밀접한 관계를 이루고 있다. 이들 세 가지 요소에 대한 요구는 프리젠테이션 도중 임의의 시점에서 발생할 수 있으며, 반복적으로 요청될 수도 있다. 이러한 요구에 즉각적으로 응답하기 위해서는 스레드가 이벤트를 처리하는 도중에 중지 및 대기 상태로 즉시 전환될 수 있도록 해야 한다. 프리젠테이션 제어부에서는 사용자 개입에 따라 새로운 사건이 발생하였을 때 시나리오를 기반으로 미리 정의된 사건 리스트를 검색하여 현재 처리중인 모든 스레드를 일시중지하고 발생된 사건을 검사하여 적절한 작업을 수행하는 스레드에게 사건 제어 메시지를 전달하여 가장 적합한 환경을 유지할 수 있도록 설계하였다.

### 5. 출력 동기화부

주어진 시나리오에 따라 정확한 프리젠테이션을 수행하기 위해서는 프리젠테이션 출력 요청 시간과 네트워크의 전송 지연 시간을 기반으로 한 멀티미디어 동기화가 반드시 수행되어야 한다. 여기서 출력 동기화는 비퍼워링 여러 데이터를 하나의 시간 펄스를 사용하여 정확한 시간에 프리젠테이션하는 것을 의미하며, 각 객체는 독립된 스레드로 처리되며 이것은 다른 객체와 함께 병

렬적으로 수행된다. 각 객체(미디어)는 독립된 단위의 스레드 형태로 실행되며, 하나의 시간 펄스에 따라 운영 및 제어되는 공유 변수에 의해 모든 스레드의 실행을 제어하며, 공유 변수의 시간 진행에 의해 프리젠테이션이 시간축에 따라 각 객체가 병렬적으로 출력된다. 본 연구에서 각 객체에 대한 스레드가 이벤트 정보를 참조하는 방법을 사용하여 이러한 형태의 프리젠테이션을 지원할 수 있도록 설계하였다.

### IV. 전체 순서도

그림 10은 제안된 시스템의 모든 수행과정을 순서도 형식으로 나타낸 것이다. 사용자는 언제든지 각 미디어에 대한 편집과 프리젠테이션에 사용할 시나리오의 제작 그리고 기존에 제작된 시나리오의 프리젠테이션을 실시간에 수행할 수 있고, 프리젠테이션 과정에 사용자가 직접 일시 정지, 역실행, 탐색 등의 행위로 프리젠테이션에 참여할 수 있다.

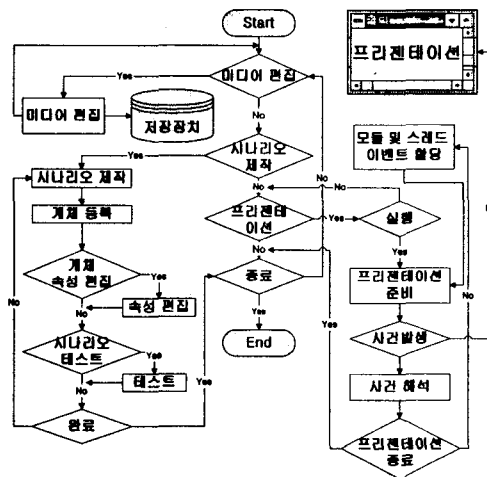


그림 10. 전체 순서도

그림 11은 미디어와 출력 윈도우에 대하여 설계 및 구현된 클래스 계층도를 나타낸 것이다.

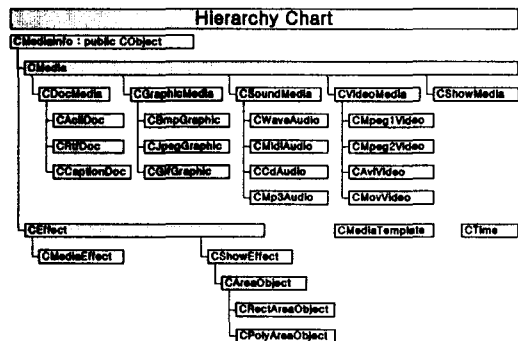


그림 11. Hierarchy Chart

본 연구에서 미디어의 일련의 이벤트에 대하여 효과(Effect)로서 정의하였다. 즉, 예를 들어 비디오나 사운드의 플레이 기능은 플레이라는 효과로써 정의하고 시작 시간과 끝 시간 동안 플레이 효과를 지속하게 된다. 텍스트의 출력은 역시 플레이라는 효과로 정의함으로써 다양한 미디어의 각 사건에 대하여 동일한 인터페이스로 접근하도록 설계 및 구현하였다. 그림 12는 각 미디어와 출력 영역에 적용될 CEffect 클래스와 열거형을 나타내고 있다.

```
enum EFFECT (
// Media Effect
    ET_NULL          = 0x00000000,
    ET_PLAY          = 0x00000001,
    ET_PAUSE = 0x00000002,
    ET_STOP          = 0x00000004,
    :
// Show Effect
    ET_SHOW = 0x00010000,
    ET_HIDE          = 0x00020000,
    :
    ET_GRID          = 0x40000000,
    ET_RANDOM        = 0x80000000,
);
class CEffect : public CMediaInfo {
// Attribute
protected:
    CTimeStruct m_StartTime, m_EndTime;
    UINT m_nRepeatCount, m_nSpeed;
    EFFECT m_eEffectType;
// Operation
public:
    CEffect();
    CEffect(const CEffect &Effect);
    const CTimeStruct GetStartTime(void) const;
    void SetStartTime(CTimeStruct const &Time);
    const CTimeStruct GetEndTime(void) const;
    void SetEndTime(CTimeStruct const &Time);
    const UINT GetRepeatCount(void) const;
    void SetRepeatCount(UINT const Count);
    const UINT GetSpeed(void) const;
    void SetSpeed(UINT const Speed);
    EFFECT GetEffectType(void) const;
    void SetEffectType(EFFECT Effect);
    const CEffect GetEffect(void) const;
    ~CEffect();
    void operator=(const CEffect &effect);
};
```

그림 12. 미디어 및 출력 윈도우의 효과

## V. 결론 및 향후 연구 방향

본 논문에서는 사용자 참여가 가능한 실시간에 멀티미디어 프리젠테이션을 수행하고 멀티미디어 콘텐츠를 제작할 수 있는 통합 시스템 설계에 대하여 다루었다. 통합형 시스템을 구현하기 위하여 다양한 미디어에 대한 편집기와 멀티미디어 프리젠테이션 시나리오 제작, 네트워크 제어 및 프리젠테이션 제어 등으로 각 부분을 분할하여 유기적으로 결합될 수 있도록 설계하였다. 또한 프리젠테이션 과정에서 발생할 수 있는 사용자 입력과 네트워크 상태 변동을 비롯한 다양한 발생 가능한 사건에 대하여 각 사건을 처리하는 스레드

를 두고 이를 효율적으로 관리 할 수 있는 이벤트 모니터 스레드를 두어 프리젠테이션 QoS 유지를 고려하였다. 현재 국내외적으로 실시간을 고려한 네트워크 전송 동기화에 관한 연구는 매우 활발하게 진행되고 있지만, 이를 개발에 적용하여 실제로 이용하고 있는 사례는 거의 없다. 본 논문에서 개발한 사용자 개입 실시간 멀티미디어 프리젠테이션 및 저작 도구도 마찬가지로 아직 실용화할 수 있는 단계까지는 많은 부분을 보완 및 추가하여 구현되어야만 네트워크를 통한 프리젠테이션 및 화상 통신, 원격 교육 등과 같은 응용 분야에 현실적으로 이용될 수 있을 것이다. 현재 우리는 본 논문에서 제안한 시스템 설계 모델을 기반으로 윈도우즈 환경에서 Visual C++ 6.0을 사용하여 통합시스템을 구현하고 있다.

## VI. 참고문헌

- [1] Orji CU, Bobbie PO, Nwosu KC, "Spatio-temporal effects of multimedia objects storage and delivery for video-on-demand systems", ACM Multimedia Systems, Vol. 5, NO. 1, pp. 39-52, 1997.
- [2] E. Biersack, W. Geyer, "Synchronized delivery and playout of distributed stored multimedia streams", ACM Multimedia Systems, Vol. 7, No. 1, pp. 70-90, 1999.
- [3] 나인호, "분산 시스템 환경에서 프리젠테이션 특성 및 동기화 구간 조정을 이용한 멀티미디어 동기화 기법," 중앙대학교, 1995년 8월.
- [4] 박승철, 최양희, "실시간 멀티미디어 동기화 기술," 한국통신학회지 제 11권 제 10호, pp. 56-67, 1994.
- [5] 나인호, 김성조, 박승규, "사용자 참여 멀티미디어 프리젠테이션을 위한 시간 관계 명세 모델," 한국정보과학회 연구지 제 21권 제 11호, pp. 1990-2002, 1994.
- [6] Goyal P, Lam SS, Vin HM, "Determining end-to-end delay bounds in heterogeneous networks", ACM Multimedia Systems, Vol. 5, No. 3, pp. 157-163, 1997.
- [7] S. B. Moon, J. Kurose, D. Towsley, "Packet audio playout delay adjustment: performance bounds and algorithms", ACM Multimedia Systems, Vol. 6, No. 1, pp. 17-28, 1998.
- [8] Kenchamma-Hosekote DR, Srivastava J, "I/O scheduling for digital continuous media", ACM Multimedia Systems, Vol. 5, No. 4, pp. 213-237, 1997.
- [9] Kwon T-G, Choi Y, Lee S, "Disk placement for arbitrary-rate playback in an interactive video server", ACM Multimedia Systems,

- Vol. 5, No. 4, pp. 271-281, 1997.
- [10] Ng RT, Yang J, "An analysis of buffer sharing and prefetching techniques for multimedia systems", ACM Multimedia Systems, Vol. 4, No. 2, pp. 55-69, 1996.
- [11] Yen W, Akyildiz IF, "A hierarchical architecture for buffer management in integrated services networks", ACM Multimedia Systems, Vol. 4, No. 3, pp. 131-139, 1996.
- [12] Feng W, Jahanian F, Sechrest S, "An optimal bandwidth allocation strategy for the delivery of compressed prerecorded video", ACM Multimedia Systems, Vol. 5, No. 5, pp.297-309, 1997.
- [13] S. Gollapudi, A. Zhang, "Buffer model and management in distributed multimedia presentation systems", ACM Multimedia Systems, Vol. 6, No. 3, pp. 206-218, 1998.