

멀티미디어 프리젠테이션을 위한 저작 도구 설계 및 구현

이규남*, 나인호*

*군산대학교 전자정보공학부

목 차

- I. 서 론
- II. 관련 연구
- III. 시스템 설계

- IV. 시스템 구현 결과
- V. 결 론

I. 서 론

멀티미디어 시스템(Multimedia System) 및 ATM과 같은 기술의 발전으로 각 가정을 비롯한 사회 전반에 걸쳐 다양한 형태의 멀티미디어 네트워크 서비스에 대한 요구가 증가하고 있으며, 이러한 수요를 만족시킬 수 있는 멀티미디어 저작 도구가 많이 개발되어 사용되고 있다.

멀티미디어 데이터는 다양한 미디어의 혼재로 인해 방대한 양의 데이터 크기와 시간에 종속적인 특성을 가지고 있으며, 이로 인하여 멀티미디어 프리젠테이션 수행에 있어 반드시 동기화(Synchronization) 과정을 거쳐야 되는 특징을 가지고 있다[1][2].

멀티미디어 데이터는 시간의 흐름에 독립적인 정적 미디어(Static Media) 데이터와 시간의 흐름에 종속적인 동적 미디어(Dynamic Media) 데이터로 구분할 수 있으며, 정적 미디어는 텍스트, 그래픽, 이미지처럼 시간의 흐름에 관계없이 그 형태가 변하지 않는 이산 미디어(discrete Media)이며, 동적 미디어는 시간의 흐름과 밀접하게 관련된, 즉 시간의 흐름에 따라 연속적으로 변화되는 오디오 데이터나 비디오 데이터 등과 같은 연속 미디어(continuous Media)를 의미한다[3]. 이러한 두 부류의 미디어를 하나의 시나리오로 구성하여 프리젠테이션을 수행하기 위해서는 프리젠테이션을 위한 미디어 데이터의 공간적 조합(spatial Composition)과 시간적 조합(temporal Composition)을 고려하여야 한다[3][4][5][6]. 즉, 멀티미디어 프리젠테이션을 위해서는 사용자가 프리젠테이션

에 사용될 다양한 미디어에 대하여 미디어간 시간 정보와 공간 정보를 어떠한 형태로든 프리젠테이션 이전에 정의하여야 하며, 이 정의는 멀티미디어 프리젠테이션을 위한 시나리오로 사용된다. 이렇게 정의된 멀티미디어 프리젠테이션 시나리오는 시간을 기준으로, 혹은 이벤트를 기준으로 명세된 후 프리젠테이션이 이루어지게 된다.

본 연구에서는 이러한 멀티미디어에 대한 원활한 프리젠테이션을 수행할 수 있고, 사용자가 원하는 형태의 멀티미디어 자료를 저작할 수 있는 시스템의 구현에 관하여 연구하였다.

본 논문의 구성은 다음과 같다. II장에서 멀티미디어 저작 도구와 관련 있는 내용에 관하여 기술하였다. III장에서 구현된 멀티미디어 프리젠테이션 저작 도구의 설계를 기술하였으며, IV장에서 구현된 내용에 관하여 설명하였다. 그리고 마지막으로 맺는다.

II. 관련 연구

1. 멀티미디어 저작 도구

멀티미디어 저작 도구는 크게 정적인 문서에 멀티미디어 데이터를 삽입함으로써 하나의 통합된 정적 문서를 저작하기 위한 도구와 각 미디어들간의 시간 및 공간적 혼합을 통해 멀티미디어 시나리오를 구성하고 이것을 통해 여러 사용자에게 프리젠테이션을 수행할 수 있도록 지원하는 멀티미디어 프리젠테이션용 저작 도구가 있다[2][4].

오소링 툴(Authoring Tool)이라고 부르는 개발용 프로그램인 저작 도구는 쉬운 사용 방법과 사용자 인터페이스(User Interface) 등을 이용하여 프로그래밍 언어에 비해 손쉽게 각종 응용 프로그램을 만들 수 있는 개발용 프로그램으로서 그래픽, 사운드, 비디오 등 다양한 멀티미디어 구성 요소들을 서로 결합하여 멀티미디어 타이틀을 일반 사용자들도 쉽게 저작할 수 있도록 도와주는 응용프로그램을 말한다. 이러한 멀티미디어 저작도구는 저작 방식에 따라 크게 3가지로 구분할 수 있으며, 다음과 같다[4].

○ 책 방식 저작 도구

책 방식 저작 도구는 텍스트나 사운드와 같은 멀티미디어 데이터들을 각각 한 장의 카드와 같이 취급하여 저작하는 도구로써 각 미디어 데이터의 카드에 순서를 갖도록 구성하여 마치 책을 한 장, 한 장, 읽어나가듯이 프리젠테이션 되는 저작도구를 말한다.

○ 흐름 방식 저작 도구

흐름 방식의 저작 도구는 아이콘과 흐름선을 사용하게 되는데, 고유한 기능을 갖는 아이콘(Icon)들을 흐름선(Flow Line) 위에 올려놓고 진행 방향을 흐름도 형식(Flow Diagram)으로 조절하여 프로그램을 작성하는 방식이다.

○ 시간선 방식 저작 도구

시간선 방식 저작 도구는 가장 일반적인 저작 도구의 형태로써 프리젠테이션 될 미디어의 구성을 세분화된 시간선 상에 나열하여 플레이 될 수 있도록 구성하는 저작도구이다. 시간선 방식의 저작도구는 여러 미디어간의 동시 실행 관계를 프레임 단위로 세부적으로 기술할 수 있다는 장점이 있다.

국내외적으로 많이 활용되는 기존 멀티미디어 저작 도구의 종류를 표 1에 나타내었다.

표 1. 기존 저작 도구의 종류

제품명	프로그래밍 모델	대상 플랫폼	제조 회사명
Multimedia Tool Book	책 방식	MS-Windows	Asymetrix Corp.
HyperCard	책 방식	Macintosh	Apple
Storyboard Live	책 방식	MS-DOS	IBM
Authorware Professional	흐름도 방식	MS-Windows Macintosh	MacroMedia
Icon Author	흐름도 방식	Macintosh	AimTech Corp.
Director	시간선 방식	MS-Windows Macintosh	MacroMedia

2. 멀티미디어 프리젠테이션

멀티미디어 프리젠테이션은 실시간에 생성되거나 저장 장치에 저장된 멀티미디어 데이터를 처리하여 사람이 인식할 수 있는 형태로 보여주는 것을 의미한다. 멀티미디어 데이터에 함축된 의미를 사람에게 효과적으로 전달하기 위해서는 미디어 데이터간의 시간 및 공간적인 정보를 효율적으로 표현하고 처리할 수 있어야 하며, 이렇게 정의된 시공간 정보를 기반으로 각 미디어를 비롯한 다양한 동기화 요소를 고려하여 프리젠테이션을 진행함으로써 사용자에 대한 QoS를 유지 및 향상시킬 수 있다[5][6].

2.1 시간관계 표현

시간 정보는 여러 미디어 데이터간의 시간 관계 또는 시간 동기화 정보를 의미하며, 공간 정보는 멀티미디어 데이터간의 공간 관계 또는 디스플레이 공간상에서 레이아웃 구조에 대한 정보를 의미한다.

멀티미디어를 구성하는 각 미디어 데이터들이 시간적으로 동기화되어 프리젠테이션 되지 않으면 시나리오 작성자가 원하는 프리젠테이션 효과를 얻을 수 없을 것이다. 따라서, 멀티미디어 데이터를 프리젠테이션 할 때에는 시간관계 명세모델(Temporal Relationship Specific Model)이 필수적으로 제공되어야 한다[6].

Allen은 임의의 두 미디어 데이터 사이에는 표 3과 같이 13개의 시간관계가 존재한다고 주장하였으며, 이들 관계 중에서 equals를 제외한 다른 모든 시간 관계는 역(inverse) 관계를 가진다. 모든 역 관계는 정상적인 시간 관계로 변환될 수 있기 때문에 기본적인 시간관계는 before, equals, meets, overlaps, during, starts, finishes까지 7가지로 정의할 수 있다[2][3][6].

표 2. X 미디어와 Y 미디어 사이의 시간관계

시간 관계	그래픽 표현	역 시간 관계	그래픽 표현
X before Y		Y before ⁻¹ X	
X meets Y		Y meets ⁻¹ X	
X overlaps Y		Y overlaps ⁻¹ X	
X during Y		Y during ⁻¹ X	
X starts Y		Y starts ⁻¹ X	
X finishes Y		Y finishes ⁻¹ X	
X equals Y			

2.2 프리젠테이션 동기화

프리젠테이션 동기화란 주어진 멀티미디어 구성 요소를 지정된 시간에 화면이나 스피커와 같은 지정된 출력 장치로 출력하는 기법을 의미한다. 멀티미디어 동기화는 대용량 멀티미디어 구성 요소를 저장 장치로부터 검색해 플레이 백(Play-back) 장치로 보여주기까지의 과정에 필수적으로 수반되는 지연시간(Delay) 때문에 발생하는 것으로 생각할 수 있다 [2][3][7][8][9]. 멀티미디어 동기화는 분산 멀티미디어(Distributed Multimedia) 또는 네트워크 기반 멀티미디어(Networked Multimedia) 등 통신 네트워크 상의 멀티미디어 응용에서 특히 중요한 이슈가 되고 있다. 멀티미디어 동기화는 그 유형에 따라 몇가지로 구분할 수 있으며 다음과 같다[4][6].

- 내용 동기화(Content Synch.)
여러 미디어 객체들이 하나의 내용을 표현하는 경우
- 공간 동기화(Spatial Synch.)
여러 미디어 객체들이 하나의 화면에 배치된 경우
- 시간 동기화(Time Synch.)
여러 미디어 객체들이 시간선을 따라가며 시간적인 전후 관계를 갖는 경우
- 미디어내 동기화(Intramedia Synch.)
비디오 데이터와 같이 하나의 시간-종속 미디어 객체내에서 프리젠테이션 단위들간에 시간관계가 성립하는 경우
- 미디어간 동기화(Intermedia Synch.)
여러 미디어 객체들이 시간선을 따라가며 서로 시간적인 관계를 갖는 경우
- 라이브 동기화(Live Synch.)
캡춰(Capture) 과정 중 존재한 시간관계를 프리젠테이션 시 정확히 다시 생성하는 것을 목표로 하는 동기화
- 이 외에도 립 동기화(Lip Synch.), 포인트 동기화(Point Synch.) 또는 사건-구동 동기화(Event-Driven Synch.) 등 다양한 동기화 유형이 존재한다.

III. 시스템 설계

1. 전체 흐름도 및 클래스 구조

그림 1은 본 논문에서 구현한 멀티미디어 프리젠테이션 저작 도구의 모든 수행과정을 순서도로 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 일시정지, 역실행, 탐색 등의 행위로 프리젠테이션에 참여할 수 있도록 설계하였다.

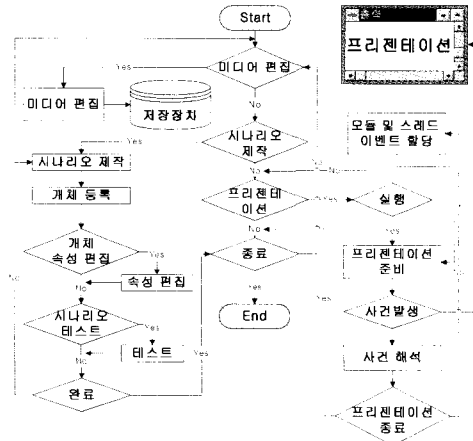


그림 1. 전체 순서도

그림 2는 각 미디어 편집 과정과 각 미디어가 프리젠테이션 데이터로 등록되어 프리젠테이션이 진행될 때 사용 가능한 클래스 계층 구조를 나타낸 것이다.

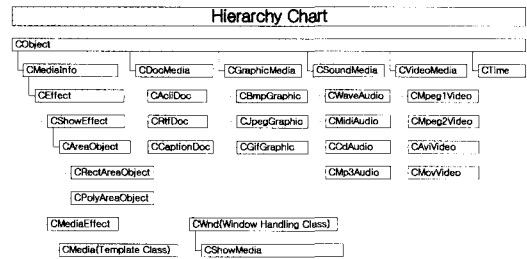


그림 2. 클래스 계층 구조

그림 2에서 CObject 클래스는 MFC(Microsoft Foundation Class)에서 제공되는 클래스로서 가장 기본이 되는 클래스이다. CObject 클래스의 하위 클래스로써 텍스트 미디어를 처리할 수 있는 CDocMedia 클래스, 그래픽 미디어를 처리할 수 있는 CGraphicMedia 클래스, 사운드 미디어를 처리할 수 있는 CSoundMedia 클래스 그리고 동영상 미디어를 처리할 수 있는 CVideoMedia 클래스를 두었다. 이들 4가지 특정 미디어 클래스는 각 클래스 별로 미디어의 성격에 따라 구분된 것이며, 각각의 하위(Child) 클래스는 실제 각 미디어 포맷을 다루는 클래스이다. 현재 매우 다양한 문서의 포맷이 플랫폼에 따라 사용되고 있으며, 본 연구에서는 현재 많이 사용되고 있는 미디어 포맷을

선별하여 본 연구에 적용하였다.

2. 각 미디어에 대한 저작기능 설계

2.1 텍스트 편집기

텍스트 편집기에서 지원 가능한 문서 포맷은 표준 ASCII로 이루어진 텍스트 문서와 윈도우즈 포맷인 RTF(Rich Text Format) 포맷과 본 논문에서 비디오의 캡션으로 사용할 수 있도록 설계한 CTF(Caption Text Format) 포맷이다.

표준 텍스트 포맷은 서식이 없는 문자열을 저장하고, 구역 구분, 쪽 구분, 줄 바꿈이 가능한 ASCII 문자를 사용한 경우 저장되는 형식으로, 대부분의 편집기나 워드프로세서에서 TXT로 생성 및 저장되는 문서 포맷이다.

RTF 포맷은 서식지정이 가능한 문자열을 표현하는 방식으로 일반 문자열뿐만 아니라 폰트, 단락 유형, 그림, 수식과 같은 정보를 포함한다.

CTF 포맷은 비디오 데이터에 대한 캡션 기능을 위한 포맷으로써 RTF와 같이 다양한 서식을 지정할 수 있고, 비디오와 결합하여 시간 정보를 포함할 수 있는 문서 포맷이다.

2.2 그래픽 편집기

그래픽 편집기에서 지원 가능한 문서 포맷은 BMP(Bitmap) 포맷, JPEG(Joint Photographic Group) 포맷, GIF(Graphic Interface Format) 포맷에 대하여 고려하였다. 그래픽 에디터는 세가지 상이한 문서 포맷간 변환이 가능하다. 라인(Line)을 비롯한 다양한 그리기(Drawing) 기능과 그리는 과정에서 객체화 처리 기능, 다중 선택 기능, 그룹화(Grouping) 기능, 그리고 포토샵과 같은 레이어(Layer) 형태로 처리가 가능하다. 또한 간단한 이미지 프로세싱(Image Processing) 작업을 처리할 수 있도록 하여, 이용자의 다양한 편집 요구를 만족시킬 수 있도록 하였다. 특히, 그래픽 에디터는 본 연구에서 수행된 비디오 편집기와 연동된 작업을 수행할 수 있도록 하였는데, 비디오 프레임 내의 화소(Pixel) 단위 편집 과정에서 그래픽 편집기를 사용할 수 있도록 설계하였다.

2.3 사운드 편집기 및 플레이어

사운드 편집기 및 플레이어에서 지원 가능한 문서 포맷은 WAV(Wave) 포맷, MIDI(Music Instrument Digital Interface) 포맷, CD-Audio(Compact Disk Audio) 포맷, Mp3 포맷 등이다. 사운드에 대한 편집 기능은

Wave 포맷으로 한정하였으며, 나머지 미디어 포맷은 플레이 기능만 지원하도록 하였다.

Wave 편집기 및 플레이어는 개방된 Wave 파일에 대하여 재생을 수행할 수 있다. 또한 역방향 재생을 비롯하여 지정된 영역에 대한 재생을 할수 있다. 그리고 편집을 시각화하기 위하여 Wave의 파형을 출력할 수 있는 기능을 설계하였으며, Wave 편집기의 자세한 기능은 표 3과 같다.

표 3. Wave 편집기의 기능

기능	설명
플레이	Wav 데이터의 재생
역방향 플레이	Wav 데이터의 역방향 재생
파형 출력	Wav 데이터의 파형 출력
블록 편집	블록 지정을 통한 Cut, Past, Copy
선택 영역 반복	선택 영역에 대한 반복 재생
녹음	다이크를 통한 Wav 녹음
에코	에코 추가 및 제거
볼륨 조절	음량 감소 및 증가
포맷 변환	샘플링 주파수 변경

2.4 비디오 편집기 및 플레이어

동영상 편집기에서 지원 가능한 포맷은 AVI(Audio Video Interface) 포맷, MPEG(Moving Picture Coding Experts Group) I과 II 등이다. Avi 포맷에 대하여 동영상 편집기를 설계 및 구현하였으며, MPEG 포맷 등의 재생 기능은 MCI(Multimedia Control Interface)를 이용하여 실행 루틴 설계하였다. MCI는 Windows 플랫폼에서 지원되는 멀티미디어를 제어하기 위한 High Level Service으로써 미디어 포맷에 관계없이 동일한 인터페이스를 제공함으로써 프로그래밍하기 쉬운 장점을 제공하지만 세부적인 제어가 어렵다는 단점을 가지고 있다. 그림 3은 이러한 MCI의 구조를 보여준다.

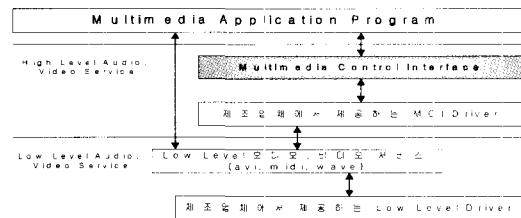


그림 3. MCI 구조

동영상 편집기는 동일한 비디오 데이터나 이종의 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 건너뛰어 실행할 수도 있다. 동영상 편집기의 기능을 표 4에 나타내었다.

표 4. Avi 편집기의 기능

기능	설명
플레이	Avi 데이터의 재생
역방향 플레이	Avi 데이터의 역방향 재생
프레임 출력	Avi 데이터의 프레임 출력
블록 편집	블록 지정을 통한 Cut, Past, Copy
선택 영역 반복	선택 영역에 대한 반복 재생
프레임 편집	Avi 프레임 데이터의 화소 단위 편집
파형 출력	Wave 파형 출력
볼륨 조절	음량 감소 및 증가

3. 프리젠테이션 제어부 설계

프리젠테이션 제어부는 구성된 시나리오를 기반으로 각 미디어들을 설정된 시간에 정확히 출력하기 위한 전반적인 제어를 수행한다. 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 데이터 량을 조절하여야 한다.

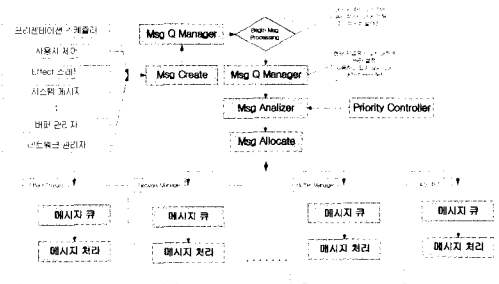


그림 4. 프리젠테이션 제어부의 구조

이를 위해 각 미디어 데이터의 출력 데드라인, 네트워크 전송지연시간, 멀티미디어 서비스 품질에 따른 데이터 요구량 및 허용 가능한 최대 전송지연시간, 예러 허용율, 컴퓨터 시스템의 부하 상태 등의 체크 요소들을 하나의 제어부를 통해 통합적으로 신속하게 처리하여 프리젠테이션이 안정적으로 수행될 수 있도록 하였다. 그림 4는 프리젠테이션 제어부의 구조

를 나타낸 것이다.

프리젠테이션 제어는 사용자 개입에 의해 요구된 명령과 시나리오를 구성하고 TRSG 동기화 모델로 명세된 후 생성된 동기화 정보와, 네트워크 동기화 정보, 프리젠테이션 실행중 동적으로 발생하는 다양한 이벤트, 프리젠테이션 되는 미디어가 보내는 이벤트 등을 이벤트 메시지 큐에 저장하여 주기적으로 이벤트 분석기에 의해 분석된다. 이때 다양한 동기화 요구에 있어 우선 순위 제어기에 의한 이벤트 처리 순서가 정해지고 동기화 정보 생성기에 의해 실제 사용 가능한 동기화 정보로 가공되어 동기화 정보 메시지 큐로 전송된 후 처리를 기다리게 된다.

IV. 시스템 구현 결과

1. 주요 자료 구조

본 연구에서 미디어의 일련의 이벤트에 대하여 효과(Effect)로서 정의하였다. 표 5는 각 미디어와 출력 영역에 적용될 CEffect 클래스와 열거형 데이터 구조를 나타내고 있다.

CEffect 클래스는 하나의 미디어 또는 출력 공간에 대하여 다양한 효과를 지정할 수 있도록 설계된 클래스이다. 그림 2의 미디어의 위치에 대한 정보 등을 정의하는 CMediaInfo 클래스를 상속받은 CEffect 클래스는 시작 시간과 지연시간, 그리고 Effect의 종류를 정의한다. Effect의 종류는 EFFECT 열거형 구조체에 표시되어 있으며, 이러한 Effect Type을 통하여 다양한 미디어에 대한 복잡한 연산을 단순화함으로써 제어와 처리를 간단하고 효율적으로 운용할 수 있는 장점을 가진다. 예를 들면, 연속 미디어인 비디오나 사운드의 플레이 기능은 ET_PLAY라는 효과로서 정의되고, 시작 시간부터 종료 시간까지 지속적으로 플레이백(Playback) 됨을 의미한다. 이산 미디어인 텍스트나 그래픽 미디어에 있어서 ET_PLAY 효과는 지정된 공간상에 지속적으로 출력되고 있음을 의미한다. 그리고 출력 공간에 대하여 ET_SHOW 효과는 출력 공간이 시작 시간과 종료 시간까지 계속 사용자에게 보인다는 것을 정의하게 된다.

표 6은 다양한 미디어를 포함할 수 있는 CMedia 클래스를 나타낸다. 텍스트, 그래픽, 사운드, 비디오 등에 대한 연산은 매우 다양하다. 멀티미디어 프리젠테이션을 수행하기 위해 본 논문에서는 미디어의 종류에 관계없이 시나

리오에 등록하여 동일한 시간과 연산을 적용하여 프리젠테이션을 수행할 수 있도록 하였다. CEffect 클래스와 마찬가지로 CMedia 클래스 역시 이종의 미디어에 대한 연산을 획일화 할 수 있는 클래스로서 템플릿(Template) 클래스로 구현하였다. 템플릿 클래스의 특징은 데이터 타입에 관계없이 동일한 클래스에 값을 정의할 수 있는 저장형 클래스이다.

표 5. 미디어 및 출력 윈도우의 효과

```
enum EFFECT {
// Media Effect
    ET_NULL = 0x00000000,
    ET_PLAY = 0x00000001,
    .
    .
// Show Effect
    ET_SHOW = 0x00010000,
    ET_HIDE = 0x00020000,
    .
    .
    ET_RANDOM = 0x80000000,
};
class CEffect : public CMediaInfo {
// Attribute
protected:
    CTimeStruct m_Start, m_Duration;
    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);
};
```

CMedia 클래스는 CMediaEffect 클래스를 상속받고 있는데, CMediaEffect 클래스는 CEffect 클래스를 상속받아 미디어에 대한 Effect를 특성화 하는 클래스이다. CMedia 클래스의 데이터 멤버인 Type 형의 포인터 변수 m_pMedia는 이종의 미디어 클래스를 저장할 수 있는 포인터 변수이다. m_pMedia 변수를 통하여 실제 미디어에 대한 다양한 연산을 수행할 수 있다. OpenMedia 함수와 PlayMedia 함수에 대하여 예를 들어 Type이

CBmpGraphic 클래스인 경우 OpenMedia에 대한 연산은 CBmpGraphic의 파일 개방을 의미한다. 그리고 PlayMedia에 대한 연산은 개방된 bmp 그래픽 파일에 대한 지정 위치에 지정 시간만큼 출력을 의미한다. 만약 Type이 Live 비디오 형식이라면 OpenMedia 함수는 비디오 데이터를 획득할 수 있는 장치(Device)를 초기화하는 역할을 수행하고, PlayMedia 에 대한 연산은 초기화된 장치로부터 화면을 획득하여 지속시간만큼 계속해서 지정 위치를 갱신(Update) 함을 의미한다.

표 6. CMedia 템플릿 클래스

```
template <class Type>
class CMedia :public CMediaEffect
{
public: //생성자
    CMedia(); // 기본 생성자
    CMedia(const CMedia<Type> &Media); // 복사 생성자
    ~CMedia(); // 소멸자

private: // 멤버 변수
    Type *m_pMedia;

public: // 멤버 함수
    void SetMediaPosition(const RECT &rect);
    void SetMediaType(MEDIATYPE eMediaType);
    LPCSTR GetMediaType(void) const;
    LONG GetTotalFrames(void);
    BOOL OpenMedia(LPCSTR pPath, HWND hWnd);
    BOOL PlayMedia(BOOL bLoop, BOOL bFullScreen);
    BOOL PauseMedia(BOOL bPause);
    BOOL StopMedia(void);
    BOOL CloseMedia(void);
    BOOL PutMedia(void);
    Type *GetMedia(void);
    void SetMedia(const Type &Media);
    int GetWidth(void) const;
    int GetHeight(void) const;
    SIZE *GetSize(void) const;
    POINT *GetPoint(void) const;
    BOOL IsPlaying(void) const;
    BOOL SeekTo(int) const;
    void SkipForward(int);
    void SkipBack(int);
    int GetSpeed(void);
    void SetSpeed(int);
    int GetFrameRate(void);
    void SetSound(BOOL);
    BOOL GetSound(void);
    HWND GetHwnd(void);
};
```

표 7은 프리젠테이션 제어부에서 각 미디어 스레드 및 제어 모듈로 이벤트를 전달하기 위해 사용되는 Message에 대한 자료 구조 있다. dHandle 변수는 처리 대상이 되는 스레드에 및 Effect에 대한 포인터 데이터이다.

dMessage는 메시지의 타입을 의미한다. 그리고 dTime 변수는 이벤트가 발생한 순간의 시간정보를 가지고 있다. wParam과 lParam은 각 부가 정보로 설정된다.

표 7. 메시지 구조

```
typedef struct tagMessage
{
    DWORD dHandle;
    DWORD dMessage;
    DWORD dTime;
    WPARAM wParam;
    LPARAM lParam;
}MESSAGE;
```

표 8은 사용자가 프리젠테이션의 시작을 사용자 인터페이스를 통하여 명령하고, 프리젠테이션이 시작 된 이후, 일시 정지 및 종료까지 수행 할 수 있는 메시지의 내용을 표현하고 있다.

표 8. 사용자 제어에 의해 발생하는 메시지

Message	실 명
ETM_INIT_PRESENTATION	초기화
ETM_START_PRESENTATION	시작
ETM_STOP_PRESENTATION	종료
ETM_PAUSE_PRESENTATION	일시 정지
ETM_REVERSE_PRESENTATION	역 수행
ETM_END_PRESENTATION	완전 종료

2. 구현 화면

그림 5는 구현된 텍스트 편집기를 통하여 문서를 편집하고 있는 과정을 보인 것이다.

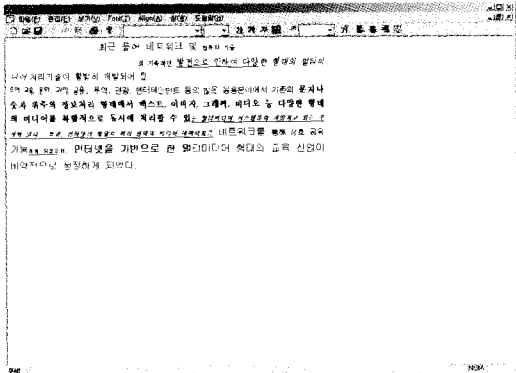


그림 5. 텍스트 편집 화면

그림 6은 그래픽 편집기의 내용을 표현하고 있다. 그림 6에서 현재 몇 개의 도형을 작업 영역에 그려 넣었고, 비트맵 형식의 그림 파일을 사각 객체에 삽입하여 편집하고 있다. 이렇게 삽입된 이미지 객체는 명도 조절이나 채도

조절, 블러링(blurring) 등 간단한 이미지 편집 작업을 수행할 수 있도록 구현하였다.

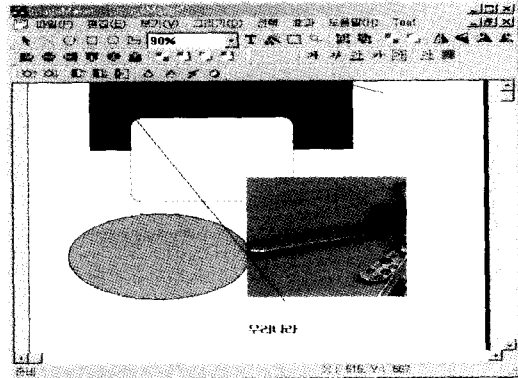


그림 6. 그래픽 편집 화면

그림 7은 Wave에 대한 사운드 편집을 수행하고 있는 사운드 편집기의 실행 화면을 캡처한 그림이다. 사운드 편집기는 개별적인 윈도우에 사운드 문서를 등록하여 플레이 기능을 수행하거나 레코딩을 진행 할 수 있도록 구현하였다. 그림 7에서는 두 개의 사운드 파일에 대하여 편집하고 있는 과정이며, 그중 하나의 사운드 파일에 대하여 플레이 기능을 실행함으로써 출력된 데이터에 대하여 다른 색상으로 파형을 표시하고 있다.

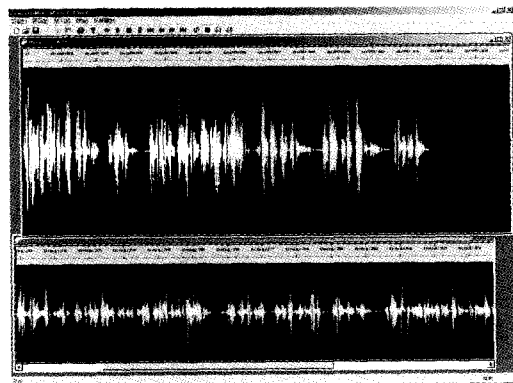


그림 7. 사운드(Wave) 편집기

그림8은 두 Avi 문서를 개방하여 편집을 진행하고 있는 화면이다. 상단의 두 프레임 라인은 개방된 Avi 문서에 대한 프레임 내용을 출력하고 있으며, 각 프레임 번호가 명시되어 있다. 오른쪽 하단의 대화상자는 Avi 문서의 정보를 보여주고 있다. Avi 문서의 프레임 비율이나 전체 프레임 수, 사이즈 등을 출력하고 있다. Avi 편집기를 통하여 프레임 단위의 비디오 편집을 수행한다.

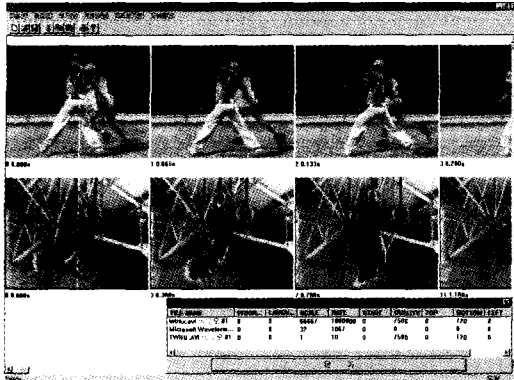


그림 8. 비디오(AVI) 편집기

그림 9는 3개의 출력 공간에 실시간 비디오 입력(좌측 상단)과 두 개의 MPEG 동영상(우측과 하단)을 프리젠테이션 시나리오 정의에 기반하여 프리젠테이션 중인 화면이다. 시나리오를 제작하는 화면은 크게 3개의 프레임으로 구성하였으며, 상단 좌측은 등록 미디어에 대한 간략한 정보를 표현하고, 상단 우측은 등록 미디어의 프리젠테이션 진행 시간을 시간선에 기반하여 표현하도록 하였으며, 하단에 프리젠테이션되는 데이터를 출력하도록 구현하였다. 사용자가 출력 공간 지정을 정확하고 효율적으로 정의할 수 있도록 하기 위하여 Grid 기능을 구현하였고, 출력 공간은 자유 자재로 확대 및 축소가 가능하도록 하였다. 또한 전체 출력 공간은 프리젠테이션 환경을 고려하여 화면의 해상도에 따라 동적으로 변경할 수 있도록 설계 및 구현하였다.

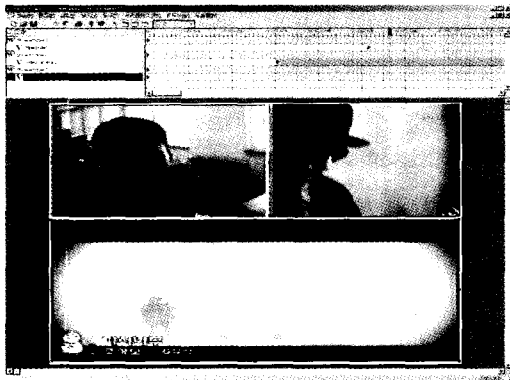


그림 9. 프리젠테이션 진행 화면

V. 결 론

본 논문은 시나리오를 기반으로 사용자 참

여가 가능한 멀티미디어 프리젠테이션을 수행하고 멀티미디어 저작을 수행할 수 있는 멀티미디어 프리젠테이션 저작 도구 설계와 구현에 대하여 다루었다. 멀티미디어 프리젠테이션을 위한 시나리오 작성을 위해 시간 선에 기반한 저작 방식을 사용하였고 특히, 사용자의 프리젠테이션 과정 개입과 네트워크 동기화를 위한 이벤트, 컴퓨터 작업 부하 등 프리젠테이션 과정상에 발생 할 수 있는 동적인 동기화 포인트 제어를 위해 프리젠테이션 제어부에 이벤트를 전달하는 스프레드를 두어 현재까지 발생한 모든 이벤트를 종합 분석하여 처리하도록 함으로써 프리젠테이션 QoS 유지를 고려하였다.

또한, 본 논문에서는 다양한 미디어의 원활한 편집을 위해서 텍스트, 그래픽, 사운드, 비디오 편집기를 각각 구현하였다. 각 미디어간 다양한 문서 포맷에 대하여 현재 가장 많이 사용되는 포맷에 대하여 우선 지원할 수 있도록 하였으며, 각 미디어 편집기 상호간 연계성을 유지하여 효율적인 멀티미디어 저작을 고려하였다.

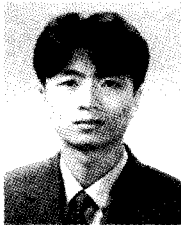
향후 네트워크의 효율적인 활용을 위하여 네트워크 동기화부의 알고리즘을 개선 및 보완하고, 발생 가능한 다양한 사건에 대하여 많은 시뮬레이션과 검증을 통하여 정의 및 추가하게 된다면 근거리 통신망을 벗어나 인터넷과 같은 초대형 네트워크를 통하여 이용 가능하게 될 것이다.

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] 황대현, 여인국, "[개정판] 멀티미디어 시스템", 정익사, pp. 265-271, 1998.
- [5] 박승철, 최양희, "실시간 멀티미디어 동기

- 화 기술," 한국통신학회지 제 11권 제 10호, pp. 56-67, 1994.
- [6] 나인호, 김성조, 박승규, "사용자 참여 멀티미디어 프리젠테이션을 위한 시간 관계 명세 모델," 한국정보과학회 연구지 제 21권 제 11호, pp. 1990-2002, 1994.
- [7] 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.
- [8] 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.
- [9] Kenchammana-Hosekote DR, Srivastava J, "I/O scheduling for digital continuous media", ACM Multimedia Systems, Vol. 5, No. 4, pp. 213-237, 1997.

저자소개



이 규 남

- 1999년 2월 군산대학교 정보통신공학과 졸업(공학사)
- 2001년 2월 : 군산대학교 전자정보공학부(공학 석사)
- 2001년 3월 ~ 현재 : 군산대학교 전자정보공학부 박사과정

정
 관심분야 : 멀티미디어 통신 시스템, 멀티미디어 응용프로 그램, 시스템 프로그래밍



나 인 호

- 1998년 2월 울산대학교 전자계산학과(공학사)
- 1991년 2월 중앙대학교 대학원 전자계산학과(공학석사)
- 1995년 8월 중앙대학교 대학원 전자계산학과(공학박사)
- 1995년 9월~현재 : 군산대학교 전자정보공학부 조교수
- 1997년 7월~현재 : 전주 첨단 영상산업 추진협의회 위원
- 1997년 7월~현재 : 한국 해양정보통신학회 편집위원
- 1997년 12월~1998년 3월 : 전자통신연구원 초빙연구원
- 1999년 10월~2000년 10월 : 전주국제컴퓨터게임축제 조직위원
- 관심분야 : 멀티미디어 통신시스템, 분산시스템, 병렬처리