

TRSG 모델을 기반으로 한 멀티미디어 프리젠테이션 및 저작 도구 개발

(Development of Multimedia Presentation and Authoring Tool based on TRSG Model)

나 인 호 [†]

(Ra In Ho)

요 약 본 논문은 고속통신망을 통하여 사용자 개입을 허용하는 멀티미디어 프리젠테이션과 멀티미디어 응용에 포함된 다양한 미디어의 저작을 동시에 수행할 수 있는 도구의 개발에 관한 것이다. 도구 개발을 위해 실시간 동기식 멀티미디어 프리젠테이션을 지원하기 위해 동적 버퍼링 기법, 송수신 시스템간의 데이터 전송률을 동기화를 위해 버퍼 관리 기법과 QoS 파라미터를 기반으로 한 적응형 전송 알고리즘을 적용하였다. 또한, TRSG 모델을 이용하여 사용자가 프리젠테이션 도중에 참여할 수 있도록 하였다. 마지막으로 제안된 도구가 QoS 유지 및 동기화에 영향을 미치는 요소들을 계속적으로 모니터링하여 멀티미디어 프리젠테이션의 수행 상태를 지속적으로 제어하는 이벤트 감시 스레드 기법을 적용하여 연속적인 출력 보장과 최소한의 QoS를 지원할 수 있도록 하였다.

Abstract In this paper, we describe the developing of a tool which supports both multimedia presentation with user's participation through a high speed network and authoring of various media using a single authoring tool. To support real-time synchronous multimedia presentation, we adopt dynamic synchronization method and adaptive transmission algorithm for synchronizing data transfer rate between sender and receiver using buffer management algorithm based on QoS parameters. And we also allow user's participation in the presentation using TRSG(Temporal Relationship Specification Graph) model. Finally, the proposed tool supports the minimal level of QoS and its continuous play-out using event auditing threads which control the current state of a multimedia presentation continuously by monitoring of negative factors effecting on QoS, and synchronization.

1. 서 론

최근 들어 고성능 저가의 멀티미디어 시스템의 보급 확대와 B-ISDN, ATM 등과 같은 초고속 통신망 기술의 눈부신 발전으로 원격지 시스템간의 멀티미디어 전송 및 공유가 가능해짐에 따라 멀티미디어 타이틀 및 원격 교육 등과 같이 여러 형태의 멀티미디어 응용을 사용자가 쉽게 제작할 수 있는 멀티미디어 콘텐츠 저작 및 프리젠테이션을 위한 도구 개발에 관한 연구가 국내 외적으로 활발히 추진되고 있다[1].

멀티미디어 데이터는 텍스트나 이미지처럼 시간의 흐름에 독립적인 이산 미디어와 시간의 흐름에 따라 연속적으로 변화되는 오디오 데이터나 비디오 데이터 등과 같은 연속 미디어들로 구성된다[2]. 이러한 형태의 미디어들을 하나의 멀티미디어로 혼합하여 프리젠테이션하기 위해서는 각 미디어간의 시간과 공간적 출력 관계를 고려해야 하며, 특히 시간적 관계는 각 미디어들을 정확한 시간에 동기화하여 연속적으로 출력하기 위해 필수적으로 만족되어야 하는 조건으로 취급되고 있다[2]. 이것은 분산 환경에서 멀티미디어 프리젠테이션을 수행할 때 멀티미디어 응용의 서비스 품질은 네트워크 상의 데이터 송수신 속도와 각 미디어 데이터의 동기식 출력 여부에 의해 큰 영향을 받는 것을 의미한다[2].

본 논문의 목적은 원격지에 분산적으로 저장된 멀티

[†] 종신회원 : 군산대학교 정보통신전파공학부 교수

ihra@ks.kunsan.ac.kr

논문접수 : 1999년 5월 6일

심사완료 : 1999년 10월 11일

미디어 데이터를 미디어 간의 시·공간적인 관계에 따라 통합된 시나리오를 구축하고, 이것을 기반으로 실시간 멀티미디어 정보 전송 및 프리젠테이션을 수행할 수 있는 도구와 사용자가 다양한 미디어 데이터를 하나의 저작 도구를 통해 제작 및 편집할 수 있는 도구를 통합적으로 개발하고자 하는데 있다. 이를 위해 각 미디어 데이터의 다양한 편집 기능을 통합적으로 구현하고, 인터럽트를 기반으로 한 사용자의 프리젠테이션 개입 기법, QoS를 기반으로 한 수신자 위주의 동기화 기법과 이벤트를 기반으로 한 프리젠테이션 수행 및 제어 기법 등을 사용하여 프리젠테이션을 수행하도록 함으로서 분산 시스템 환경에서 멀티미디어 응용의 저작 및 프리젠테이션에 이용할 수 있도록 하였다[2].

논문의 구성은 다음과 같다. 먼저, 2장에서는 본 개발에 적용된 관련 기법을 기술하였고 3장에서는 제안된 도구의 시스템 구성 및 각부의 기능에 관하여 기술하였다. 4장에서 개발된 시스템의 내용을 중심으로 자료 구조 및 시행 화면 등에 관하여 기술하였다. 마지막으로 5장에서는 결론 및 향후 연구 방향에 대해 기술하였다.

2. 개발에 적용한 기법

멀티미디어 저작 도구는 크게 정적인 문서에 멀티미디어 데이터를 삽입함으로써 하나의 통합된 문서를 제작하기 위한 도구와 각 미디어들간의 시간 및 공간적 컴포지션을 통해 멀티미디어 시나리오를 구성하고 이것을 통해 여러 사용자에게 프리젠테이션을 수행할 수 있도록 지원하는 멀티미디어 프리젠테이션용 저작 도구로 분류된다[2].

제안된 도구는 후자의 경우에 해당하며 멀티미디어 저작 도구에서 제공해야 할 기능에 중점을 두고 다양한 형태의 미디어를 편집 및 저장할 수 있는 통합된 편집기를 제작하여 프리젠테이션을 위한 시나리오 제작에 이용할 수 있도록 하였다. 또한, 원격 및 지역 시스템에 저장된 미디어 데이터뿐만 아니라 캠코더나 오디오 입력장치와 같이 실시간에 생성되는 멀티미디어 데이터를 실시간으로 처리하여 프리젠테이션에 사용될 수 있도록 하였다.

이러한 도구 개발을 위해서는 프리젠테이션 과정에 사용자의 동적 참여가 허용되어야만 하고, 특히 프리젠테이션 QoS를 유지하기 위해서는 구성된 시나리오에 따른 프리젠테이션 동기화가 반드시 수행되어야만 한다 [2] [3] [4] [6] [7].

2.1 TRSG

프리젠테이션 과정상에 이루어지는 사용자 참여를 고

려한 기술은 아직까지 국내외적으로 완벽한 단계는 아니며, 일부 실용화 소프트웨어에서 사용자 참여를 부분적으로나마 허용하고 있는 추세이다.

지금까지 원격 멀티미디어 프리젠테이션 과정에 사용자 개입을 고려하여 제안된 기법들은 거의 Petri-Net 기반의 토큰 전달 방식에 의한 병렬 트랜잭션 실행 제어 방식을 이용하고 있으며, 이 방식은 고정된 형태의 프리젠테이션을 수행하기 위한 정적 프리젠테이션 수행에 적합하다[2] [4] [5]. 따라서 멀티미디어 프리젠테이션 수행 과정에 실시간 발생하는 사용자 입력을 프리젠테이션에 동적으로 반영하기에는 적합하지 않다는 단점을 가지고 있다[2].

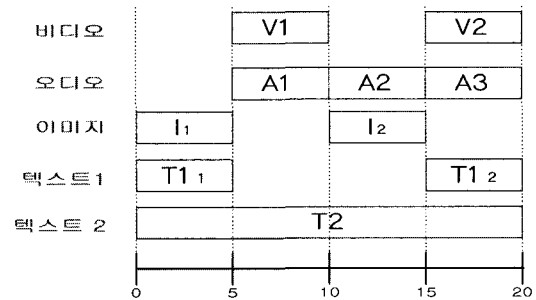


그림 1 프리젠테이션 시나리오 작성 예

본 논문에서는 멀티미디어 프리젠테이션에 사용자의 동적인 참여를 허용하기 위해 TRSG 모델을 사용하였다. TRSG 모델은 비순환 유한 그래프를 기반으로 한 미디어 데이터간의 복잡한 시간 관계 및 사용자 개입으로 인한 동적 프리젠테이션 순서 변화를 현재의 프로세스 수행 모델에 직접 적용할 수 있는 시간 관계 명세 모델이다[2].

그림 1은 프리젠테이션 시나리오의 작성 예를 출력시간을 기준으로 보인 것이다. 여기서 I₁ 과 T_{1_1}, T₂는 동일한 시점에 출력이 시작되어야 하는 미디어이며, V₂, T_{1_2}, A₃는 동시에 출력이 종료되어야 하는 미디어들을 의미한다. 또한, T₂는 프리젠테이션 과정 전반에 걸쳐 출력되지만 나머지 미디어들은 중간에 시작하거나 종료하는 것을 나타내고 있다.

그림 2는 그림 1의 가상 시나리오를 TRSG 모델에 적용한 결과이다.

이러한 TRSG 모델에 사용자의 동적 개입을 처리하는 인터럽트 노드를 해당 미디어 노드 및 이와 관련된 미디어 노드로 연결하여 처리함으로써 사용자 개입을 프리젠테이션에 반영할 수 있다.

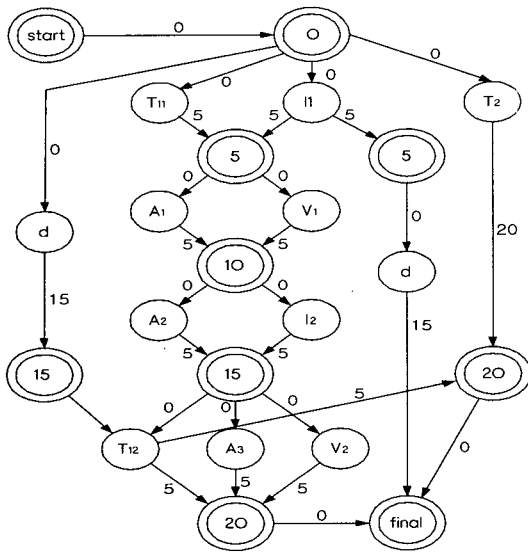


그림 2 그림 1에 TRSG 모델을 적용한 결과

그림 3에서는 프리젠테이션에 사용자의 입력이 발생 하였을 때 발생하는 TRSG 모델을 나타낸다.

사용자의 입력은 인터럽트 노드로서 명령을 받을 직접적인 노드 M_i 와 간접적으로 영향을 받아야 할 M_j 및 M_k 를 함께 처리함으로써 사용자 입력으로 발생할 수 있는 비 동기화 현상을 최소화하면서 원활한 사용자 개입을 허용하도록 하였다.

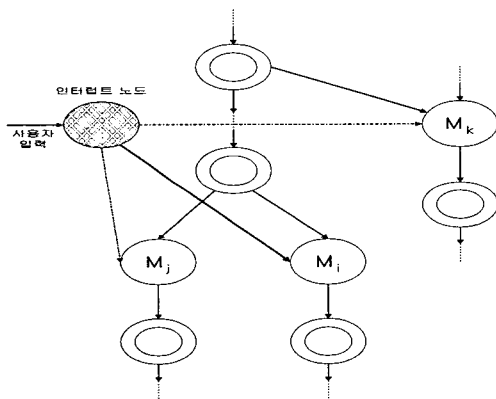


그림 3 사용자 입력 처리

2.2 적응형 동기화

네트워크는 그 상태를 예측할 수 없고, 네트워크 자체 지연시간 등의 일반적 특성으로 인하여 멀티미디어 데이터를 정해진 시간에 프리젠테이션 하기에는 해결해

야할 비동기화 요소가 많이 있다. 이러한 문제를 해결하기 위한 수많은 동기화 관련 연구 결과가 발표되었고 대표적인 동기화 기법은 정적 동기화 기법, 송신자에 의한 적응형 동기화 기법, 수신자에 의한 적응형 동기화 기법 등이 있다. 하지만 이러한 동기화 기법들은 특정 멀티미디어 응용을 대상으로 하는 기법들이 대부분이어서 일반적인 응용에 적용하기에는 더욱 다양한 요소가 고려되어야 한다[2].

본 논문에서는 이러한 문제점을 해결하기 위해 적응형 동기화 기법을 적용하고 이는 크게 3단계로 나누어진다. 첫 번째 단계는 동기화 정보 추출 단계이고 두 번째 단계인 정적 동기화를 거쳐 마지막 세 번째 단계인 동적 동기화를 수행하게 된다[2].

2.2.1 동기화 정보 추출

분산 환경에서 미디어 데이터의 전송 또는 프리젠테이션 과정에서 만족되어야 할 미디어 데이터 간의 동기화 요구 조건을 추출한다.

시나리오에 등록된 각 미디어간의 시공간 관계는 TRSG에 의해 명세화되고 다시 미디어간 시간 관계를 한눈에 파악할 수 있는 다중 시간 관계 테이블을 작성하여 프리젠테이션 될 미디어의 동기화 정보를 추출하게 된다.

2.2.2 정적 동기화 단계

미디어 전송에 필요한 전송 환경 구성 및 스케줄을 결정하는 단계이다. 즉 이 단계에서는 프리젠테이션에

<p>Algorithm:Determin_Buffer_Requirement(PTT,BRT) Input: PTT(Presentation Time Table) Output: BRT(Buffer Requirement Table)</p> <p>Let m be the number of media in PTT; Let X(i) be the size of media i in bits; Let R_f be the fixed amount of buffer space for accommodating initial presentation;</p> <p>Let R_v be the variable amount of buffer space for adapting to the variation of channel delays on a given network;</p> <p>for(i = 1 to m) { if(i is a continuous media) { determine upper bound of R_v that is sufficient for some delays; determine R_f needed for presentation of media i; BRT[i] = $R_f + R_v$; } else BRT[i] = X(i); }</p>

그림 4 버퍼 요구량 계산 알고리즘

참여하는 각 미디어 데이터들이 그것들 간의 시간 관계 및 프리젠테이션 특성에 따라 검색, 전송, 그리고 프리젠테이션 되도록 이에 필요한 모든 요구사항을 결정하고 스케줄링한다. 또한 이 단계의 마지막에서는 전송된 미디어 데이터를 저장하기 위한 버퍼 공간의 크기를 계산한다.

그림 4는 정적 동기화 단계에서 멀티미디어 데이터의 송수신에 필요한 버퍼 요구량을 결정하는 알고리즘을 나타낸 것이다.

2.2.3 동적 동기화 단계

동적 동기화 단계는 전송된 각 미디어 데이터들이 그것들간의 시간 동기화 관계에 따라 프리젠테이션장치로 출력되는 단계이다.

Algorithm Synchronization_server(S)

Input: S(scenario for a given multimedia presentation);

```

Let n be the number of media in a given PTT;
Let m be the number of packets in media i;
Let T be the local clock of a presentation system;
Let LTU(i) be the logical time unit for media i;
Let P(i,j) be jth packet of media i to presented;
/* phase of construction the temporal relationships */
construct TRSG for the S;
call Get_Presentation_Time_Table(TRSG, PTT);
call Get_Multiple_Temporal_Relationship_Tible(PTT,
MTRT);
/*phase of static synchronization */
call Get_Packet_presentation_Time_Table(PTT, PPTT);
call Get_Packet_Transfer_Time(PTT, PTRT);
call Get_Packet_Retrieval_Time(PTT, PPTT, PTRT,
PRTT);
call Determine_Buffer_Requirement(PTT, BRT);
/* phase of dynamic synchronization */
establish network connection between senders and
receivers;
call Buffer_Management();
initiate the presentation for S and activate T;
    
```

```

i = j = 1;
while(i is less than or equal to n)
{
    assign LTU(i) to media i;
    set LTU(i) to the presentation start time of media i;
    if(LTU(i) is not less than the value of T)
    {
        i++;
        while(j is less than or equal to m)
        {
            scan the jth packet from the buffer of media
            i;
            if(jth packet is available)
            {
                Intramedia_Synchronization(P(i, j));
                Intermedia_Synchronization(P(i, j));
                j++;
            }
        }
    }
}
    
```

그림 5 전체 동기화 수행 알고리즘

이 단계의 궁극적인 목표는 실시간에 동적으로 발생하는 미디어 데이터 간의 동기화 이상 현상을 처리하여 서로 관련된 미디어들이 동기화되어 순차적 또는 병렬적으로 출력되도록 보장하는데 있다. 이 단계에서 적용하는 동기화 요소는 미디어간 및 미디어내 동기화와 버퍼관리 등이 있다.

그림 5는 본 논문에서 적용한 적응형 동기화 기법의 전체 수행 구조를 나타내는 알고리즘으로서 동기화 정보 추출, 정적 동기화, 동적 동기화를 거쳐 미디어간 논리 시간 시스템에 의해 프리젠테이션이 순차적 또는 병렬적으로 수행되는 과정을 차례대로 기술한 것이다.

3. 시스템 구성 및 기능

제안된 멀티미디어 프리젠테이션 및 저작 도구는 그림 6과 같이 미디어 편집부, 시나리오 제작부, 프리젠테이션 제어부, 네트워크 동기화부, 출력 동기화부로 나누어져 있으며, 각각의 기능은 다음과 같다.

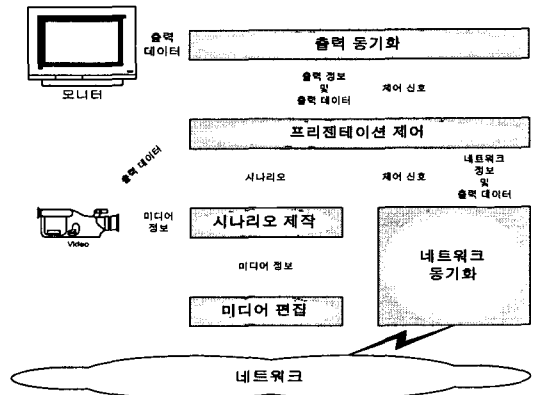


그림 6 전체 시스템 구조

3.1 미디어 편집부

미디어 편집부는 텍스트, 이미지, 그래픽, 사운드, 동영상 편집기 및 플레이어로 구성된다. 이러한 미디어 편집기는 멀티미디어 저작 도구에 있어 반드시 필요한 요소들이다. 본 개발에서는 이러한 다양한 미디어 편집기를 통합 구현함으로써 저작 도구를 개발하였다. 또한 미디어 편집부에서는 프리젠테이션에 사용하게 될 미디어에 대하여 사용자가 원하는 형태로 각 미디어를 편집하도록 하여 프리젠테이션 저작의 효율성을 도모하였다.

개발된 미디어 편집기의 종류는 일반 문서 편집기,

그래픽 편집기, 동영상 편집기, 사운드 편집기가 있다. 특히 동영상 편집기는 멀티미디어 저작 도구 및 프리젠테이션을 위해 핵심적인 도구라 할 수 있다. 동영상 편집기에서는 동영상과 관계된 내부 사운드의 처리도 고려하였고, 프레임의 화소 단위 및 프레임 단위 편집 및 처리를 고려하여 개발하였으며 각 편집기는 서로 연동하여 작업이 가능하도록 개발하였다.

3.2 시나리오 제작부

멀티미디어 데이터를 프리젠테이션하기 위해서는 먼저 각 미디어간의 시·공간적 관계를 정의하여 하나의 멀티미디어 데이터로 결합하여야 한다. 이것을 멀티미디어 컴포지션(composition)이라고 하며, 시간 및 공간 컴포지션으로 나누어진다[2] [8] [10].

시간 컴포지션을 통해 프리젠테이션에 포함될 미디어 데이터들 간의 시간적 관계를 설정하고, 공간 컴포지션을 통하여 각 미디어들이 출력되어야 할 디스플레이 공간상의 위치와 형태를 지정한다.

시나리오 제작부에서는 그림 13과 같이 프리젠테이션에 사용할 미디어에 대한 시간적 관계를 막대 그래프 형태로 표시하도록 처리하였고, 공간적 관계는 각 미디어의 출력 위치를 설정하도록 하였다.

이렇게 작성된 시각적인 프리젠테이션 시나리오는 TRSG 모델에 적용되어 본 논문에서 사용한 1단계 동기화 과정인 동기화 정보를 추출할 수 있는 형태로 가공 처리된다.

3.3 프리젠테이션 제어부

프리젠테이션 제어부는 구축된 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 여러가지 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.

이를 위해 각 미디어 데이터의 출력 데드라인, 네트워크 전송지연시간, 멀티미디어 서비스 품질에 따른 데이터 요구량 및 허용 가능한 최대 전송지연시간, 에러 허용율, 컴퓨터 시스템의 부하 상태 등의 체크 요소들을 하나의 제어부를 통해 통합적으로 신속하게 처리하여 프리젠테이션이 안정적으로 수행될 수 있도록 하였다. 그림 7은 개발된 프리젠테이션 제어부의 핵심 부분들을

나타낸 것이다.

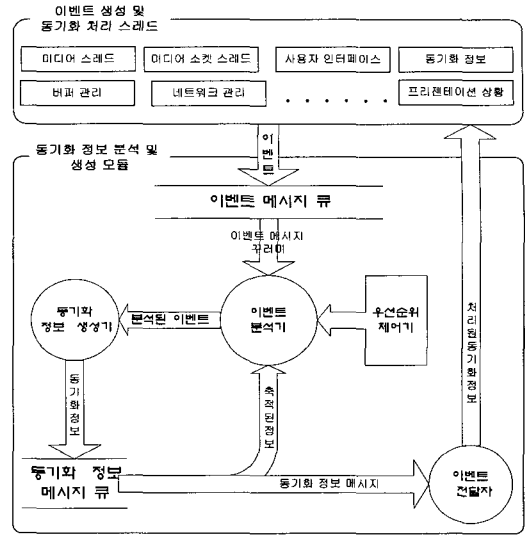


그림 7 프리젠테이션 제어부 구조

프리젠테이션 제어는 사용자 개입에 의해 요구된 명령과 1, 2단계를 거치며 형성된 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 우선순위 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.

동기화 정보 메시지 큐의 미처리된 동기화 정보들은 이벤트 분석기에 의해 다시 재고될 수 있도록 함으로써 현 단계에서 최적의 대화식 프리젠테이션이 유지될 수 있도록 고려하였다.

사용자 참여, 출력 동기화, 네트워크 동기화는 프리젠테이션의 QoS의 유지와 밀접한 관계를 이루고 있으며, 이들 세 가지 요소에 대한 요구는 프리젠테이션 도중 임의의 시점에서 발생할 수 있고 또한 반복적으로 요청될 수도 있다. 이러한 요구에 즉각적으로 대응하기 위해서는 스레드가 이벤트를 처리하는 도중에 중지 및 대기 상태로 즉시 전환될 수 있도록 하여야 한다. 프리젠테이션 제어부에서는 사용자 개입에 따라 새로운 사건이 발생하였을 때 시나리오를 기반으로 미리 정의된 사건 리스트를 검색하여 현재 처리중인 모든 스레드를

일시중지하고 발생한 사건을 검사하여 적절한 작업을 수행하는 스레드에게 사건 제어 메시지를 전달한다.

이러한 프리젠테이션 제어부의 운용은 다양한 동기화 정보를 처리할 수 있으며, 동적으로 발생하는 프리젠테이션 과정상의 다양한 이벤트 처리, 더욱이 실시간 대화식 처리를 지향함으로써 멀티미디어 프리젠테이션 효율이 향상될 수 있도록 하였다.

3.4 네트워크 동기화부

실시간 네트워크 동기화 부분은 가장 해결하기 어려운 부분으로서 동기화 알고리즘이나 구현 방법의 특징이 아닌 네트워크 전송지연 시간의 변동 폭 수준에 따라 좌우된다. 특히, 인터넷과 같이 전송 거리가 멀고 네트워크 트래픽 부하가 시간에 따라 수시로 변화하여 초당 전송 시간을 예측할 수 없는 네트워크 환경에서는 매우 심각한 문제로 발생한다. 본 논문에서는 네트워크 지연 시간 변동에 따라 각 미디어의 검색 및 전송 시간을 재조정하는 스케줄링 기법을 적용하여 이러한 문제를 해결하고자 하였으나, 기본적으로 압축된 동영상의 연속적 전송에 적합하지 않을 정도로 낮은 속도의 네트워크 환경에서는 비동기화 현상 및 출력 중단 문제가 어쩔 수 없이 발생한다. 이러한 문제는 ATM, B-ISDN, ADSL과 같은 고속통신망이 구축됨에 따라 점차 해결될 것으로 예상된다.

본 논문에서는 네트워크 동기화를 구현하기 위하여 네트워크 송수신 처리부에서 데이터 전송률과 수신률을 감시하며 네트워크 트래픽 상태 및 시스템 작업 부하에 따라 프리젠테이션에 필요한 이벤트를 프리젠테이션 제어부의 이벤트 모니터 스레드로 전달하고, 이벤트 모니터 스레드는 수행해야 할 프리젠테이션 작업을 식별 및 직간접적으로 관련있는 작업 스레드에 메시지를 전달함으로써 작업 수행을 지시하도록 하여 네트워크 상태에 의한 동기화 문제를 감소시키도록 하였다.

3.5 출력 동기화부

출력 동기화부 또한 프리젠테이션 제어부와 면밀히 통신하면서 현재 상황에 맞는 프리젠테이션을 수행하도록 하였다. 출력 동기화부의 대부분 구성 요소는 실제 프리젠테이션에 사용될 미디어에 대한 프리젠테이션 스레드로 이루어져 있다.

이들 각 미디어 스레드는 프리젠테이션 제어부가 관찰하는 논리시간에 따른 프리젠테이션을 수행하도록 하였고, 원격지로부터 전송된 멀티미디어 데이터는 네트워크 제어부 버퍼로부터 출력 데이터를 전달받아 처리하도록 하였다.

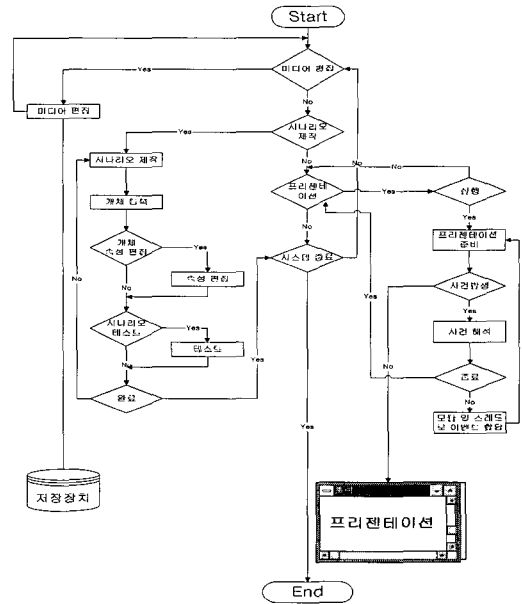


그림 8 제안된 도구의 수행 흐름도

4. 시스템 구현

그림 8은 제안된 저작 도구의 모든 수행 과정을 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.

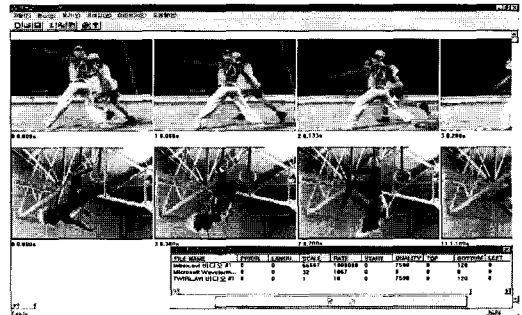


그림 9 동영상 편집기

그림 9는 동영상 편집기 화면을 나타낸 것이다. 동영상

상 편집기는 동영상을 각 프레임 단위 이동, 삭제할 수 있고 다른 동영상과 연계된 작업을 수행할 수 있으며, 또한 자체 제작된 그래픽 편집기를 통하여 프레임 화소 단위로 편집할 수 있다.

또한 해당 동영상 프레임에 관련된 사운드 데이터를 서로 연동하여 편집할 수 있도록 하였다. 그림 9에서 현재 두 개의 동영상을 동시에 편집하고 있고, 그림의 하단에 나타나 있는 대화상자는 현재 편집중인 동영상 데이터에 대한 핵심 정보를 나타내고 있다.

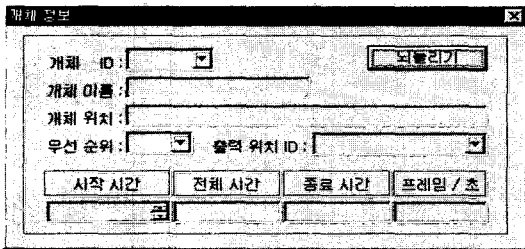


그림 10 미디어 정보 변경을 위한 대화상자

그림 10은 프리젠테이션에 사용할 미디어 객체의 정보를 표시하는 대화 상자이며, 그림 11은 각 미디어 객체에 대한 정보를 나타내기 위한 자료구조를 나타낸 것이다.

```

struct ItemInfo
{
    LPSTR lpID;           // 객체의 식별자
    UINT nClass;         // 객체의 종류
    LPSTR lpFullPath;    // 객체의 위치
    LPSTR lpFileName;    // 객체 이름.
    CTimeStruct *lpStartTime; // 시작 시간.
    CTimeStruct *lpTotalTime; // 전체 수행 시간
    CTimeStruct *lpEndTime; // 종료 시간.
    UINT nPriority;      // 우선순위
    int nFrame;         // 프레임 수.
    int nFrameRate;     // 초당 프레임 수
    CWndAttribute *lpOutput; // 출력할 윈도우
                        //포인터
};
class CWndAttribute
{
private:
    CString m_strWndID; // 윈도우 ID
    HWND hWnd;         // 윈도우 핸들
    CRect *m_pWndPos;  // 윈도우 위치(크기)
    UINT m_nAttribute; // 출력 속성
public:
    CWndAttribute(); // 생성자
    virtual ~CWndAttribute(); // 소멸자
};

```

그림 11 미디어 실행 정보 구성을 위한 자료구조

대화 상자는 프리젠테이션을 위한 시나리오 제작 과정에서 사용되며, 등록된 미디어 객체가 현재 가지고 있는 정보를 표시한다. 이때, 기존에 설정된 값을 대화상자를 통해 언제든지 변경할 수 있다. 그리고 우선순위는 해당 미디어가 프리젠테이션되는 과정에서 디스플레이 공간상의 출력 우선 순위를 나타내며, 이것은 또한 예상하지 않은 네트워크 전송지연이 발생하였을 때 프리젠테이션 속도 및 품질 조절을 위해 선택할 미디어 객체의 순위를 결정하는데 사용된다. 출력 위치 ID는 객체가 출력될 윈도우 ID를 의미한다. CTimeStruct은 미디어의 시간과 관계된 처리를 가지고 있는 클래스이다. nFrame은 미디어 객체에 포함된 전체 프레임 수를 표시하며, 이 멤버 변수는 객체의 nClass에 따라 다른 값을 가진다. 즉, 프레임 구분이 명확한 비디오 데이터의 경우에는 nFrame 변수의 값은 전체 프레임 수가되지만 텍스트 문서, 그래픽, 이미지와 같은 이산 미디어에 대해서는 1로 정의된다. nFrame 변수에 접근하기 위해서는 nClass 멤버 변수의 값도 함께 접근해야만 한다. nFrameRate 역시 nClass에 따라서 무시되거나 활성화된다. lpOutput 멤버변수는 해당 객체가 출력될 윈도우에 대한 포인터와 출력되는 속성 값들을 지정한다.

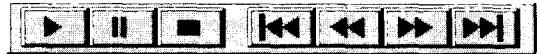


그림 12 사용자 제어 도구바

그림 12는 프리젠테이션 수행 과정에 사용자가 프리젠테이션을 직접 제어 할 수 있는 사용자 제어 도구바를 보인 것이다. 각 아이콘은 왼쪽에서부터 차례대로 프리젠테이션 실행, 일시 정지, 중단, 처음 시작점으로, 뒤로 탐색, 앞으로 탐색, 끝지점에서의 이동을 의미한다. 사용자는 프리젠테이션 과정의 어느 시점에서든 각 아이콘 기능을 선택할 수 있으며, 선택된 명령에 따라 프리젠테이션 제어부로 해당 이벤트가 전달되고 프리젠테이션 제어부에서는 지정된 명령을 수행한다.

그림 13은 프리젠테이션 시나리오를 작성하기 전에 각 미디어 객체에 대한 ID와 객체 이름, 그리고 전체 수행 시간에 대한 정보를 구성하는 초기 화면을 나타낸 것이다.

그림 14는 그림 1과같은 시나리오를 개발된 시나리오 제작부에서 6개의 미디어 객체를 시나리오로 구성한 예를 화면으로 나타낸 것이다. 여기서 각 객체별 시작 시간, 종료시간, 그리고 수행 시간이 막대 그래프 형태로 보여준다.

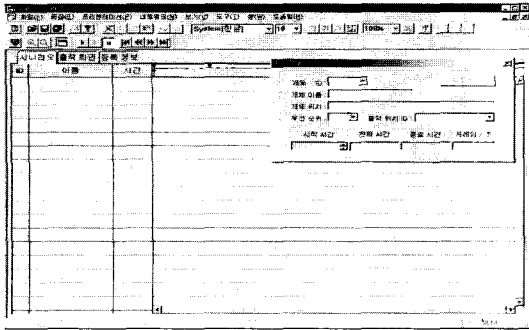


그림 13 초기 화면

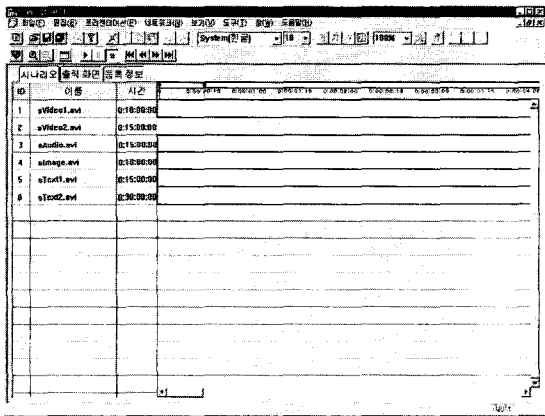


그림 14 구성된 시나리오 화면



그림 15 작성된 시나리오에 따라 실행되는 장면

그림 15는 그림 14에서 작성한 시나리오에 따라 프리젠테이션되는 과정상의 시점에서 화면의 내용을 나타

낸 것으로서 2개의 비디오, 텍스트, 그리고 그래픽 데이터가 정의된 디스플레이 공간 상의 위치에 따라 출력되는 것을 보인 것이다.

그림 16은 비디오 플레이어의 기능을 나타낸 것이다. Play 함수를 통해 동영상의 반복 실행을 설정할 수 있고, 출력될 윈도우상의 위치(SetPosition 함수)에 관계없이 출력 윈도우의 풀 스크린 기능이 가능하다. SeekTo 함수를 사용하여 프레임 단위로 탐색할 수 있도록 구현하였고, 전방향 및 후방향으로의 탐색이 가능하다. 또한 설정된 속도에 관계없이 자체적으로 실행 속도를 조절할 수 있도록 하였으며, 동영상에 포함되어 있는 사운드(음성)의 출력을 활성화 또는 비활성화 시킬 수 있다.

```

void SetPosition(int nX, int nY, int nCX, int nCY); // 실행 위치 설정
BOOL Play (BOOL bLoop = FALSE, BOOL bFullScreen = FALSE); // 실행
void Stop (void); // 멈춤
void Pause (void); // 일시 중지
int GetTotalFrames (void); // 전체 프레임
int GetCurrentFrame (void); // 현재 프레임
BOOL SeekTo (int);
void SkipForward (int); // 앞으로
void SkipBack (int); // 뒤로
int GetSpeed (void); // 실행 속도
void SetSpeed (int); // 실행 속도 설정
int GetFrameRate(void); // 프레임 비율
void SetSound (BOOL); // 오디오 발생 유무
BOOL GetSound (void);
    
```

그림 16 비디오 플레이어의 기능

5. 결론 및 향후 연구 방향

본 논문에서는 사용자 참여가 가능하고 실시간에 멀티미디어 프리젠테이션을 수행하고 멀티미디어 데이터를 저작할 수 있는 도구를 개발하여 다양한 미디어 데이터들을 자체 개발한 미디어 편집기를 사용하여 저작할 수 있도록 하였다. 또한, 프리젠테이션의 핵심기술인 사용자 개입 허용을 위해서 TRSG 모델에 인터럽트를 사용하여 처리하였고, 동기화 부분에 있어서 적용형 동기화 알고리즘을 적용하였으며, 프리젠테이션 제어에 있어 이벤트 검사 스레드에 의한 통합 제어 기법을 사용하여 프리젠테이션 동안에 요구된 최소한의 QoS를 유지시킬 수 있도록 하였다.

본 연구 개발 및 개발에 사용된 동기화와 사용자 개입에 관한 기술은 멀티미디어 타이틀 제작, 화상 회의, 원격 의료, 사내 방송 및 인터넷 방송, 실시간 온라인

게임, VOD, AOD 등 네트워크와 멀티미디어 콘텐츠가 연동된 산업 분야에 다양하게 활용될 수 있을 것이다.

현재 국내외적으로 실시간을 고려한 네트워크 프리젠테이션에 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 추가하여야 할 것이다.

제한된 도구를 실용화하기 위해서는 다양한 멀티미디어 데이터 포맷을 지원할 수 있도록 미디어 편집부를 보완하여야 하고, 분산환경에서 네트워크의 전송지연시간 변동에 보다 신속적으로 대응하면서 멀티미디어 응용에서 요구하는 서비스 품질을 최적으로 유지할 수 있도록 동적 버퍼링 기법과 네트워크 정보 피드백을 기반으로 한 재전송 스케줄링 기법의 구현을 보완하여 연속적인 실시간 멀티미디어 전송이 이루어지면서 최소한의 네트워크 및 시스템 자원이 할당되도록 네트워크 동기화부를 보강하는 것이 필요하다. 마지막으로 멀티캐스팅 기능과 화이트 보드 기능이 추가되어야 할 것이다.

Transactions on Networking, Vol. 1, No. 2, Apr. 1993

- [8] T. D. C. Little and A. Ghafoor, "Spatio-Temporal Composition of Distributed Multimedia Objects for Value-Added Networks," *IEEE Computer*, Vol. 4, No. 10, Oct. 1991.
- [9] L. C. Yau and D. G. Messerschmitt, "On Architectures for Video Composing," *Multimedia Systems*, Springer-Verlog, pp. 181-191, 1994.
- [10] K. Woosaeng, L. Ee-Peng, S. Jaideep, "Timing Specification and Synchronization for Multimedia Information System," *Proc. of the IEEE Workshop on Architectural Aspect of Real-Time Systems*, San Antonio, Tx., Dec. 1991.

나 인 호



1988년 울산대학교 전자계산학과 학사.
1991년 중앙대학교 전자계산학과 석사.
1995년 중앙대학교 전자계산학과 박사.
1999년부터 현재까지 군산대학교 정보통신전과공학부 조교수. 관심분야는 멀티미디어 통신, 멀티미디어 시스템소프트웨어, ATM 프로토콜설계

참 고 문 헌

- [1] W. Bauerfeld and H. Westbrock, "Multimedia Communication with High-Speed Protocols," *Computer Networks and ISDN Systems* 23, North-Holland, pp. 143-151, 1991.
- [2] 나인호, "분산 시스템 환경에서 프리젠테이션 특성 및 동기화 구간 조정을 이용한 멀티미디어 동기화 기법", 박사학위논문, 중앙대학교, 1995.
- [3] R. Steinmetz, "Synchronization Properties in Multimedia Systems," *IEEE Journal on Selected Area in Communication*. vol. 8, No. 3. pp. 401-412, 1990.
- [4] S. Ricarel, W. Tonathen, M. David, "A Quality-of-Service Specification for Multimedia Presentation System," Springer-Verlag, pp. 251-263, 1995
- [5] B. Prabhakaran, S. V. Raghovan, "Synchronization Models for Multimedia Presentation with User Participation," *Multimedia Systems* Springer-Verlag, pp. 53-62, 1994.
- [6] S. Ramanathan and P. V. Rangan, "Feedback Techniques for Intra-Media Continuity and Inter-Media Synchronization in Distributed Multimedia Systems," *The Computer Journal*, Vol. 36, No. 1, 1993.
- [7] S. Ramanathan and P. V. Rangan, "Adaptive Feedback Techniques for Synchronized Multimedia Retrieval Over Integrated Networks," *IEEE / ACM*