

불연속미디어 처리를 위한 CM Toolkit의 확장

김 용 대†

요 약

분산 멀티미디어 시스템의 기본적인 요구 사항은 비디오와 오디오 같은 연속미디어의 생성 그리고 검색 시점에서 표현 시점까지의 엄격한 지연과 지터를 요구하는 인트라 미디어 동기화와 다수의 연속 미디어 스트림을 병렬로 표현할 때 트랙들간의 상대적 시간 관계에 관한 제한을 만족시키는 인터미디어 동기화를 제공하는 것이다.

본 논문에서는 분산 환경에서 연속미디어를 이용하는 주문형 시스템 개발도구인 버클리 CMT를 텍스트, 이미지, 그래픽 등의 불연속 미디어와 비디오, 오디오 등의 연속 미디어 형태의 멀티미디어 동기화가 가능하도록 확장-CMT를 설계하고, 이를 분산 원격강의 응용에 구현하였다.

Extension CM Toolkit for Discrete Media Processing

Yong-Dae Kimb†

ABSTRACT

Basic requirements of distributed multimedia system is generating continuous media like videos and audios, synchronizing intra-media required limitations of delay and jitter from searching to display, and representing multiple continuous media streams in parallel. Meeting these, synchronization of inter-media is provided with satisfying the constraints of relative time gaps among the tracks.

Though the Berkeley VOD has excellent qualities, it also has disadvantages. In concrete terms; the CM Toolkit which is known as the development tool for the VOD application service and composition of VOD system of the Berkeley system deals with only the continuous media such as video and audio. So it is incompetent to the composition of on-demand system and the applied service development that is applied by the discrete media such as text, image, and graphics.

In this thesis, the purpose of this research is to study the characteristics of data of the discrete media and continuous media; and to design the extended CMT and to implement LOD with it.

1. 서 론

컴퓨터를 통한 멀티미디어 처리 기술과 고속 네트워크의 발달로 컴퓨터 통신망을 이용한 다양한 분산 멀티미디어 응용들 즉, 멀티미디어 원격강의 시스템, 멀티미디어 화상회의 시스템, 멀티미디어 정보 검색 시스템 등에 대한 많은 연구가 진행되고 있다. 이러한 멀티미디어 응용들을 구현하기 위해서는 해결되어야 하는 여러 가지 문제들이 있다. 그 중 하나가 멀티미디어 데이터의 시간 관계를 유지하는 멀티미디어 동기화 기술이다.

멀티미디어 동기화에 관한 기존 연구를 살펴보면 다음과 같다. 연구 [1]에서는 멀티미디어 데이터의 전송에 관여하는 모든 노드들이 하나의 클럭을 공유하도록 하고, 이 공유 클럭에 맞추어 동기화를 실현하는 방법을 제안하였다. 각

미디어의 전송 단위와 재생되어야 하는 시간을 포함하고 있어 이를 받아 재생시간을 결정하고 있다. 연구 [2]에서는 수신측의 버퍼 레벨을 제어하는 방법을 사용하고 있는데 수신측에서는 데이터가 전송되는 속도에 맞추어 프레임울 폐기하거나 버퍼에 복사하여 동기화를 해결하고 있다.

연구 [3]에서는 연속 데이터 스트림들을 일련의 단위들로 모델화하고, 각 단위는 그 스트림의 시작에 비례하는 각 단위의 시간적 위치에 따라 주어지는 논리적 타임 스탬프를 갖고 있으며 데이터 전송 속도에 따라 이를 진행시킨다. 그리고 논리적 타임 스탬프 내의 다른 논리적 장치들을 스킵(skip)하거나 중단시켜 이의 속도를 조정하고 있다.

버클리 VOD 시스템은 분산유닉스 컴퓨터네트워크환경하에서 기본적으로 TCP/IP와 UDP/IP를 사용하고 있고 스토리보드(storyboard) 개념과 LTS(Logical Time System)를 이용하여 연속 미디어(CM, Continuous Media)간의 동기화 및 VCR 기능을 제공하며 분산된 여러 형태의 비디오 클립

† 정 회 원 : 선문대학교 컴퓨터정보학부
논문접수 : 2000년 7월 5일, 심사완료 : 2001년 1월 26일

파일(clip file)들과 오디오 클립파일들을 편집하여 종합 VOD 서비스를 제공할 수 있는 최첨단의 기능을 갖고 있는 시스템이다.

버클리 VOD는 위와 같은 뛰어난 장점을 갖고 있지만 버클리 VOD시스템 구성 및 VOD 응용서비스 개발도구인 CMT(CM Toolkit)[4,5]가 비디오나 오디오 같은 연속 미디어만 처리하기 때문에 텍스트나 이미지, 그래픽과 같은 불연속 미디어를 이용하는 주문형 시스템의 구성과 응용서비스 개발에는 부적절한 단점을 갖고 있다. 뿐만 아니라 기존의 다른 소프트웨어도 연속미디어 중심으로 연구와 개발이 이루어져 있어 불연속미디어와 연속미디어가 혼합된 진정한 주문형 멀티미디어 서비스 개발에는 이용하기가 어려운 문제점을 갖고 있다.

따라서 본 논문에서는 불연속미디어와 연속미디어의 데이터 특성과 CMT를 분석하여 불연속미디어 처리부분을 CMT 인터프리터에 탑재하여 보다 확장된 CMT를 구현하여 연속미디어와 불연속미디어 사이에 동기화가 요구되는 LOD(Lecture On Demand) 시스템에 적용하려고 한다.

본 연구 논문에서 다루고 있는 내용을 구체적으로 나열하면 다음과 같다. 제2장에서는 버클리 CMT를 설명하고, 제3장에서는 불연속미디어 처리설계를, 제4장에서는 확장-CMT 및 원격강의시스템 구현에 대해 설명하고 마지막으로 제5장에서 결론을 낸다.

2. 버클리 CMT

CMT에서의 개념, 데이터 구조, 기능 등에 대하여 설명한다.

2.1 CMT(Continuous Media Toolkit)

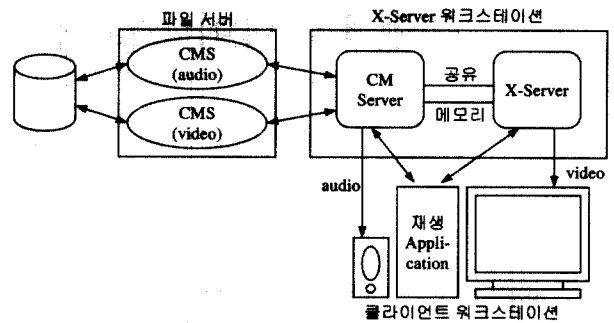
CMT는 연속 미디어를 응용하기 위한 개발환경을 제공하기 위한 것으로, 오디오/비디오 데이터를 로컬/원격으로 처리할 수 있다.

CMT를 이용한 CM Player는 다음 (그림 1)과 같은 사용자 인터페이스를 제공하는데, 사용자의 입력에 따라 X-Server를 통하여 압축된 비디오 복원을 위하여 CMX 프로세스와 연결시키면 CMX 프로세스가 비디오 파일서버 내의 CMS와 연결하여 실시간으로 전송한다. 이 자료는 사용자 클라이언트 내의 CMX 프로세스에 의해 복원되어 재생된다.

애플리케이션 프로세스는 사용자 인터페이스를 생성하고 사용자 입력 상황을 조정하며, CMX는 CMS에 의해 보내진 오디오와 비디오 패킷의 재생을 동기화한다.

CMT는 동기화된 오디오와 비디오를 재생하는데, 시스템은 TCP/IP 기반의 제어연결 프로토콜과 UDP/IP 기반의 데이터전송연결 프로토콜이다.

버클리에서 CMT를 이용한 실험을 보면 LAN 실험에서 JPEG[6,7]로 압축된 320×240 픽셀러의 2.8Mbit/sec 비디오



(그림 1) CM Player 처리 구조

연속 흐름을 10Mbps 이더넷에서 40fps(frame per second)로 재생하고 있고, WAN 실험에서 352×240의 비디오 연속 흐름을 18여개의 홉(hop)을 통해 17fps 이상으로 재생하고 있다.

CMT의 코드는 C 프로그래밍 언어[8]와 Tcl/Tk 언어의 확장 버전으로 작성되어 있다.

2.2 스토리보드

저장된 연속미디어 데이터의 재생 스케줄의 표현을 위해 스토리보드를 사용하며, 연속미디어 데이터의 스트림이 스케줄에 따라서 지정된다.

각 스트림은 클립리스트라는 클립(A, B, C, D)의 연속적 흐름으로 구성되어 있다. 클립은 프레임(재생 가능한 데이터의 단위)의 연속적 흐름을 저장하는 파일인 클립파일의 세그먼트이다.

2.3 클립파일 형식

모든 연속미디어 데이터는 클립파일 안에 저장되며, 클립파일은 임의적인 제어, 효율적인 접근(reading)을 지원하며, 스토리보드에서 클립을 간단히 재사용할 수 있게 설계되어 있다. 샘플 클립 파일은 미디어와 관계 있는 정보를 갖고 있는 헤더 부분과, 미디어마다 다른 헤더, 미디어를 담고 있는 raw data 부분, 프레임 오프셋 표(FOT, the Frame Offset Table), 그리고 프레임 시간표(FTT, the Frame Time Table)의 다섯 부분으로 되어 있다.

2.4 CMT 오브젝트

CMT 애플리케이션은 스토리보드를 지원하는 통신 오브젝트로 구성되어 있다[9].

CMT는 LTS 오브젝트와 하나 이상의 스트림 오브젝트를 이용하여 스토리보드를 구현하고 있으며 각각의 스트림 오브젝트는 클립리스트와 포인터를 가지고 있고 LTS는 Speed와 Offset의 두 슬롯을 갖고 있다[10,11].

2.5 CM Toolkit의 문제점

다양한 형식의 비디오 클립파일들과 오디오 클립파일들로 편집하여 쉽게 제공 자료를 만들 수 있게 해주고, 중복

된 내용의 비디오 및 오디오 자료를 공동으로 활용할 수 있어 저장 공간을 절약할 수 있으며, 오디오와 비디오의 분리 전송 및 동기화할 수 있게 하는 LTS 정보를 활용하여 음성다중 VOD 서비스의 제공시에 원하는 음성만 전송이 가능하기 때문에 전송대역을 절감할 수 있다.

그러나 연속 미디어만을 다룰 수 있기 때문에 텍스트나 이미지, 그래픽과 같은 불연속 미디어를 포함한 보다 다양한 분산형 멀티미디어 응용 프로그램을 개발하는데 부적합하다는 문제점이 있다.

3. 불연속 미디어 처리 설계

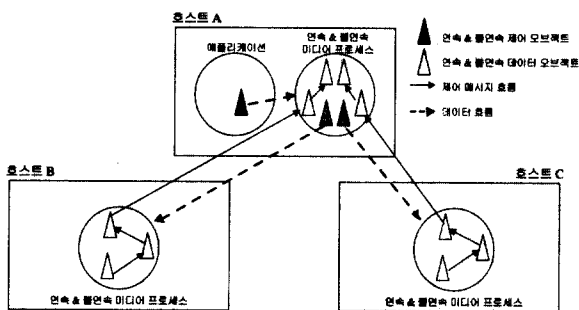
기존 CMT에서 다룰 수 없었던 불연속 미디어의 문제점 해결을 위해 확장 및 재설계하는데, 이를 확장-CMT라 부르기로 한다.

3.1 확장-CMT의 개발 목적과 설계시 고려사항

동시성의 특성을 갖고 있는 연속 미디어 데이터는 실시간 전송과 정확한 동기화가 필요한 반면, 불연속 미디어 데이터는 시간 요구가 엄격하지 않는 대신 높은 신뢰성을 요구한다. 그러나 불연속 미디어 데이터도 타이밍이 요구되는 혼합 데이터일 때는 타이밍과 신뢰성이 모두 필요하게 된다[9].

3.2 확장-CMT 구성 요소와 동작원리

연속 미디어와 불연속 미디어를 사용할 때에는 연속 미디어와 불연속 미디어 오브젝트를 생성하고 사용자가 원격 멀티미디어 자료를 시청할 수 있게 원격의 컴퓨터 상에 있는 멀티미디어 자료를 해당 연속 미디어와 불연속 미디어 오브젝트에 연결하여 fetch한다(그림 2).



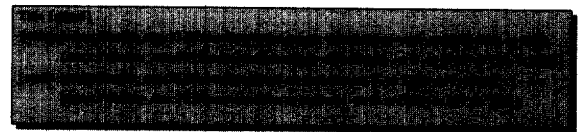
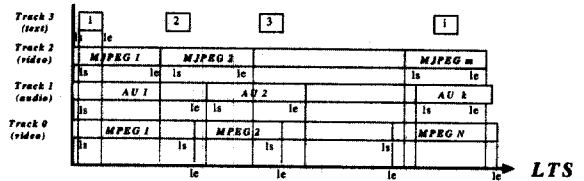
(그림 2) 확장-CMT로 작성한 응용의 구동 원리

그림에서 확장-CMT를 이용하여 연속 미디어와 불연속 미디어를 적용하면 호스트 A, B, C 상의 연속 미디어와 불연속 미디어 프로세스들에 연속 미디어와 불연속 미디어 오브젝트들을 생성한다. 이 때 호스트 B와 C 상의 연속 미디어와 불연속 미디어 프로세스들의 연속 미디어와 불연속 미디어 오브젝트들은 호스트 A의 연속 미디어와 불연속 미디어 프로세스를 통하여 연결할 수 있다.

3.3 확장-스토리보드 설계

버클리 CMT를 확장한 스토리보드 모델은 다음 (그림 3)과 같이 설계하였다.

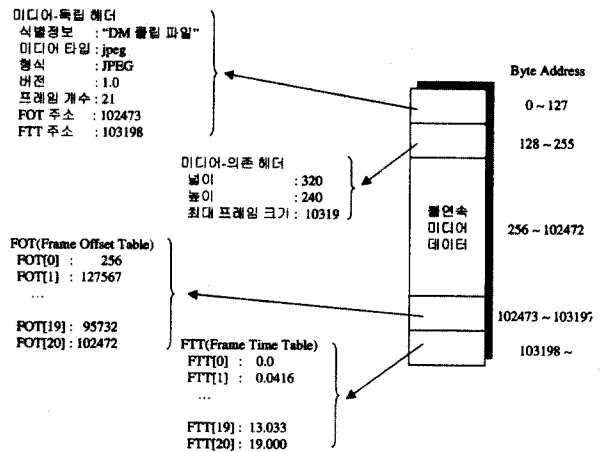
이는 상호 관련성이 있는 이미지, 그래픽, 문자들을 영상과 음성, 음악과 동기적으로 표현하는 메커니즘으로 확장된 스토리보드이다[12].



(그림 3) 확장-CMT용 스토리보드

3.4 불연속 미디어 클립파일 형식 설계

불연속 미디어 데이터도 연속 미디어와 같이 임의의 제어, 효율적인 접근(reading)을 지원하고, 그리고 스토리보드에서 클립을 간단히 재사용하기 위하여 클립파일 형태로 저장된다. 불연속 미디어 클립파일도 미디어 독립 헤더 부분, 미디어 의존 헤더 부분, raw data 부분, 프레임 오프셋 표(FOT), 프레임 시간 표(FTT)의 다섯 부분으로 설계하였다.



(그림 4) 불연속 미디어 클립파일 형식

3.5 불연속 미디어 처리과정 설계

불연속 미디어의 스토리보드는 두 가지 형태의 오브젝트, 즉 LTS 오브젝트와 스트림 오브젝트를 갖고 있다. 각각의 스트림 오브젝트는 스토리보드에 불연속 미디어 데이터 스트림을 표현하고 있다. 스트림 오브젝트는 LTS와 클립리스트의 두 슬롯을 갖고 있는데 이는 다음과 같다.

〈표 1〉 스트림 오브젝트 정리

슬롯	내용
LTS	스트림의 동기화를 위한 논리 시간 명세
클립리스트	클립들의 리스트 각 클립은 한 클립파일의 세그먼트들로 구성되어 있음

CMT에서의 스트림 오브젝트는 일반적으로 애플리케이션과 CMX 사이에 공유하고 있는 분산 오브젝트인 Tcl-DP이다. 애플리케이션이 공유 스트림 오브젝트의 클립 리스트 슬롯을 정의하고 있을 때 CMX에서의 상태를 이용하여 이러한 변화를 찾고, 선택된 데이터를 재생할 수 있다. 이러한 슬롯과 상태를 이용하여 두 프로세스, 즉 implicit control이라는 애플리케이션과 CMX 사이에 통신을 제공할 수 있다.

3.5.1 미디어 소스/목적지/자원 그리고 패킷 소스/목적지 불연속 미디어를 재생하기 위하여 공급자/사용자 모델을 사용한다. CMS는 공급자이고, CMX는 사용자이다. 공급자/사용자 모델은 미디어 소스, 미디어 목적지, 패킷 소스, 패킷 목적지, 미디어 자원의 다섯 오브젝트로 구현되고 있다.

프레임은 미디어 소스 오브젝트에 의해서 두 번째 저장소에서 읽혀진다. 미디어 소스는 클립 파일의 형태와 미디어의 특성을 가지고 있다. CMS와 CMX 사이에 대역폭이 충분하지 않을 때 이를 이용하여 어떤 프레임은 읽고, 어떤 프레임은 건너뛰어야 하는지를 결정한다.

미디어 소스는 클라이언트로 전송할 클립의 리스트를 갖고 있는 클립 리스트 슬롯, 클립 리스트와 관련 있는 LTS 오브젝트의 이름을 저장하는 LTS 슬롯, 그리고 CMS와 CMX 사이에 네트워크 연결을 나타내는 다양한 네트워크 슬롯의 세 슬롯을 갖고 있다. 클립 리스트 슬롯은 미디어를 읽은 CMS에 의하여 제공되는 불연속 미디어 데이터의 부분을 나타내고 있다. 이는 스트림의 클립 리스트에 각 호스트가 지정되어 있고 그 호스트가 제공하는 클립의 집합을 제거하기 위하여 하나의 클립 리스트가 생성되므로 애플리케이션과 CMX 사이에 분산되어 있는 스트림 오브젝트를 이용하여 계산할 수 있다. 빈 클립은 다른 CMS에 의하여 제공되어 삽입된다.

미디어 소스는 재생 전에 로컬 디스크로부터 짧은 시간(0.5초)에 프레임은 읽는다. 각각의 프레임은 조각들로 쪼개어진 후 이를 전송하기 위해 패킷 소스로 보낸다. 패킷 소스는 컴퓨터 네트워크를 통하여 패킷 목적지로 조각을 보내고, 패킷 목적지는 프레임은 다시 조립하고 분실된 패킷의 재전송을 요구한다. 프레임이 완성되면 프레임의 재생을 위해 미디어 목적지로 보내어진다. 그러나, 재생을 위하여 적절한 시간에 프레임은 전송하는 것이 가능하지 않을 수도 있다.

패킷 목적지가 완전히 조립된 프레임은 미디어 목적지로 보내면, 미디어 목적지는 재생을 위하여 필요한 사용자 프로세스를 실행하고 프레임은 스케줄링 한다. 이러한 처리의 한 예로 소프트웨어로 압축을 복원하는 것을 들 수 있다. 여기서 소프트웨어로 압축을 복원할 때까지 기다리면 동기화 문제

〈표 2〉 미디어 소스 슬롯들

슬롯들	내용
LTS	스트림을 제어하는 LTS의 이름 저장
클립리스트	미디어 소스에 의해 공급될 스트림의 위치가 지정된 클립리스트
네트워크	CMS와 CMX 사이의 연결을 나타내는 다양한 구조

〈표 3〉 미디어 소스 메소드

처리 방법	설명
LtsChanged	제어하는 LTS 때의 Speed나 Offset이 변화되었을 때 생성된다.
Create	미디어 소스 생성
Destroy	미디어 소스 파괴
Configure	슬롯들을 위한 읽기/쓰기 방법

가 발생할 수도 있다. 이러한 처리는 미디어적인 특성을 갖고 있기 때문에 미디어 목적지가 다르면 각각의 미디어 타입마다 필요하게 된다. 미디어 목적지는 도착하는 모든 불연속과 연속 미디어 데이터에 있는 불충분한 자원 때문에 생기는 스케줄링 충돌을 지적으로 해결할 수 있고, 완전히 전송되지 못한 데이터(즉, CMS와 CMX간의 대역폭이 감소할 때 생기는 완전하지 않은 데이터)를 찾아내어 CMS로 알려준다. 이러한 상세한 처리내용은 어디서든지 구할 수가 있다.

재생을 위한 프레임의 스케줄링은 프레임의 스트림과 관련이 있는 LTS를 이용하여 프레임의 논리적인 시작 시간과 끝나는 시간을 시스템 시간으로 환산하여 수행하며 at-event의 높은 우선 순위가 프레임은 재생하기 위하여 만들어진다.

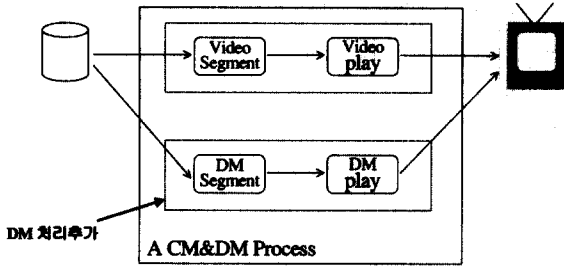
프레임은 재생할 시간이 되면, at-event는 미디어 목적지가 프레임은 재생하기 위한 방법인 미디어 자원을 호출하면 스케줄링 된다. 재생은 미디어 목적지와 프레임의 재생의 조화를 이루기 위하여 하드웨어의 압축을 푸는 상태를 다시 정의한다. 따라서 미디어 자원은 여러 미디어 목적지 사이에서 소프트웨어의 압축을 복원하기 위한 CPU를 포함하여 하드웨어 정보를 보내는데 이의 내용도 미디어에 따라 많이 다르다. 이것의 내부적인 상태는 미디어에 높은 의존성을 가지고 있다.

불연속 미디어(DM)를 저장장치에서 출력장치로 보낼 때 고려해야 할 일은 미디어가 로컬에서 처리되는 경우와 분산환경에서 처리되는 경우로 나누어 생각해야 한다.

3.5.2 로컬 환경에서 불연속 미디어 처리

다음(그림 5)은 로컬 환경에서 불연속 미디어 오브젝트를 추가하여 설계한 것을 간략하게 나타낸 것이다.

1. 첫 번째 cdmwish(연속불연속미디어 서버 데몬)가 cdm-player.tcl(연속불연속플레이어)을 실행한다.
2. 첫 번째 cdmwish가 RPCServer가 되기 위해 포트 \$gTempPort를 오픈한다.
3. 첫 번째 cdmwish가 exec로 child.tcl을 실행하는 두 번째 cdmwish를 child process로 실행한다.

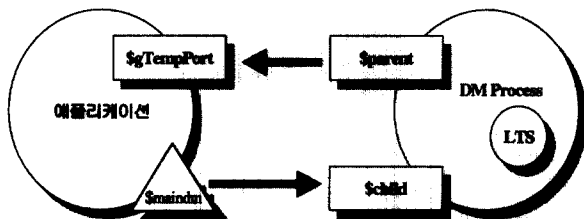


(그림 5) 로컬 환경에서 불연속 미디어(DM) 처리

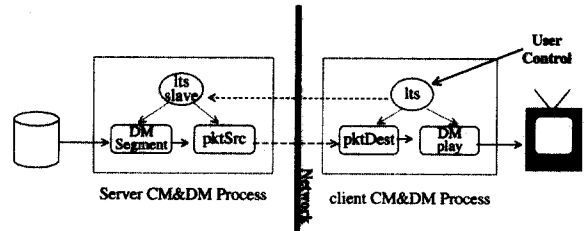
4. 두 번째 cdmwish가 첫 번째 cdmwish(RPCServer)에 대한 RPCClient가 되기 위해 명령을 보내줄 포트 \$parent를 오픈한다.
5. 두 번째 cdmwish가 RPCServer가 되기 위해 포트 \$child를 오픈한다.
6. 두 번째 cdmwish가 첫 번째 cdmwish에게 dp_RDO "Makeconnection \$child" 명령을 전달한다.
7. 첫 번째 cdmwish는 "Makeconnection \$child"를 받아 interpret하고 cdmpayer.tcl상에서 실행한다.
8. MakeConnection 명령은 \$maincdm cdmt object를 생성한다.
9. cdmpayer.tcl은 \$maincdm을 통하여 child CDM process로 들어가는 포트 \$child connection을 설정한다.
10. cdmpayer.tcl은 \$maincdm을 통하여 child CDM process상에 lts object를 생성

3.5.3 분산 환경에서 불연속 미디어 처리

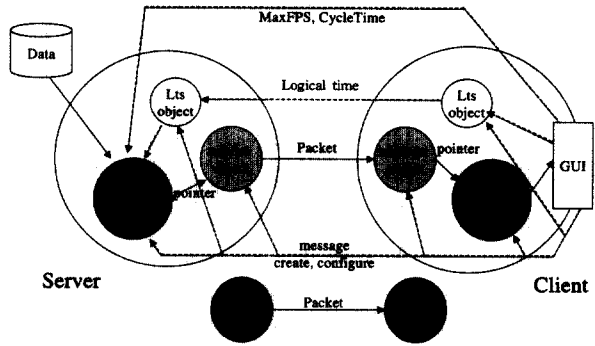
네트워크를 통하여 각 미디어들이 전송/표현되기 때문에 각 미디어간의 특성이 미치는 영향은 상당히 크지만 불연속 미디어가 소용량이기 때문에 트래픽에 큰 영향은 없다. 그러나 연속 미디어와 불연속 미디어가 혼합되어서 사용되는 통신 응용에는 신뢰성이 높은 동기화가 필요하다. 분산 환경에서 서버에서 클라이언트까지 미디어를 전송할 때 각 미디어간에 동기화를 하면서 불연속 미디어를 표현하는데 모든 8개의 오브젝트가 필요하고, 각 불연속 미디어 관련 오브젝트를 생성하고, 연결하여 작동하기까지 총 15 단계의 처리과정이 필요하다. 상세한 처리 과정과 불연속 미디어 프로세스와 오브젝트의 관계도는 다음 (그림 8)과 같다.



(그림 6) 로컬 환경에서의 불연속 미디어 프로세스와 오브젝트의 관계도



(그림 7) 분산 환경에서 DM 처리



(그림 8) 분산환경에서의 불연속미디어 프로세스를 추가 설계한 관계도

4. 확장-CMT 및 원격 강의 시스템 구현

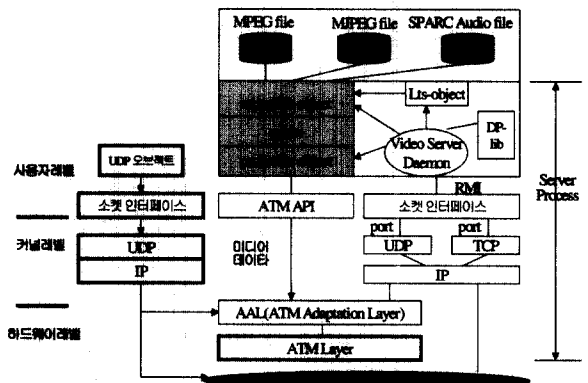
4.1 구현 환경

시스템은 Windows NT 환경의 Pentium PC 2대(Server 1대, Client 1대)이고 프로그램은 객체지향 프로그래밍 기법을 적용하여 C 언어로 구현하였고 이를 위해 Tcl-Tk 라이브러리, Tcl-DP 라이브러리 등을 특별히 사용하였다.

4.2 확장-CMT 생성 절차

CMT를 확장하는데 알아야 할 것은 먼저 CMT 오브젝트에 관한 사용법과 Tcl-Tk 언어를 사용할 수 있는 기술이 필요하고, 전체 소스의 트리 구조 파악, 소스 트리에 관한 make 파일을 이해해야 한다.

4.3 확장-CMT기반의 서버 구조



(그림 9) 확장-CMT를 사용한 미디어서버 구조

확장-CMT를 기반으로 기존에 구성할 수 없었던 불연속 미디어에 관한 처리를 할 수 있도록 불연속미디어 파일 오픈 및 버퍼링부분을 개선하였으며, 불연속미디어와 연속미디어간의 동기화를 위한 LTS 오브젝트를 개선 설계하였으며, 서버의 구성은 (그림 9)와 같다.

4.3.1 서버상 오브젝트

서버 프로세스는 비디오 파일서버 시스템 상에 실행되고 있는 비디오 서비스 태스크의 기본 단위이며, 비디오 및 오디오 스트림을 디스크로부터 읽어서 클라이언트로 전송하기 위하여 여러 오브젝트를 사용하고 있다.

- mpegClipfile : MPEG-1 비디오 파일로부터 주문된 부분만큼만의 비디오 데이터를 읽고, 버퍼링해서 그 포인터를 다음 오브젝트에게 넘겨준다.
- mjpegClipfile : Motion JPEG 비디오 파일로부터 주문된 부분만큼만의 비디오 데이터를 읽고, 버퍼링해서 그 포인터를 다음 오브젝트에게 넘겨준다.
- dmjpegClipfile : jpeg 타입으로 불연속미디어와 연속미디어간의 시간적 관계성에 따라 만들어진 불연속미디어 파일로부터 주문된 부분만큼의 불연속 미디어 데이터를 읽고, 버퍼링해서 그 포인터를 다음 오브젝트에게 넘겨준다.
- audioFile : PCM 방식으로 샘플링된 오디오 데이터를 파일로부터 읽고, 버퍼링해서 그 포인터를 다른 오브젝트에게 넘겨 준다.
- Its : Logical Time System 오브젝트로 플레이백 메커니즘과 VCR 조정 메커니즘을 제공한다. 서버의 Its는 클라이언트의 Its의 명령을 받아 데이터의 해당 부분만큼 읽어오게 한다. Vaule와 Speed 값을 파라미터로 받는다.
- pktSrc : 사이클-UDP 프로토콜에 의하여 파일 오브젝트로부터 넘겨 받은 데이터를 클라이언트에게 전달한다. 재전송 매카니즘이 있다.
- mpegPriority : MPEG 프레임(I,P,B)들을 우선순위에 따라 재정렬하여 현재의 네트워크 대역폭 수준에 맞게 전송 비트율을 적용시킨다.

4.3.2 클라이언트상의 오브젝트

- pktDest : 사이클-UDP 프로토콜을 사용하여 서버의 pktSrc 오브젝트로부터 데이터를 전송받아 버퍼링한다.
- Its : 디스플레이 오브젝트 및 서버의 Its 오브젝트를 조정하여 동기화 프리젠테이션 및 FF, REWIND, PAUSE 등의 VCR 조정을 한다.
- mpegPlay : 버퍼로부터 전송받은 MPEG-1 비디오 데

이터를 디코딩하고 지정 윈도우 상에 재생한다.

- mjpegPlay : 버퍼로부터 전송받은 MJPEG 비디오 데이터를 디코딩하고 지정 윈도우 상에 재생한다.
- dmjpegPlay : 버퍼로부터 전송받은 JPEG 불연속 미디어 데이터를 디코딩하고 지정 윈도우 상에 재생한다.
- auPlay : 버퍼로부터 전송받은 오디오 데이터를 스피커 상에 재생시킨다.

4.3.3 새로운 오브젝트 dmjpegSegment의 정의

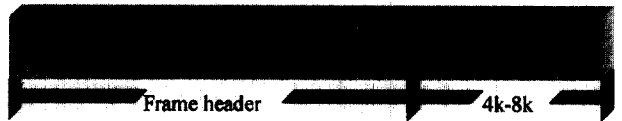
- 데이터 구조체 : dmjpegSegment object

4.3.4 새로운 오브젝트 dmjpegPlay의 정의

dmjpegPlay 오브젝트는 프레임은 받아 플레이할 수 있는 프레임의 큐를 관리하며 jpeg 디코딩 디바이스를 사용하여 각 프레임을 디코드하고 디스플레이한다.

4.4 불연속 미디어 클립파일의 생성 결과

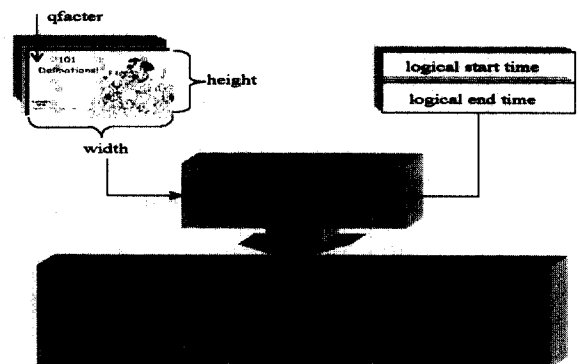
불연속 미디어는 특성상 연속적으로 균등하게 재생되는 것이 아니라, 연속 미디어와의 의미적인 관계에 따라서 동기화 된 후 재생되므로 각 프레임당 어느 정도의 표현시간이 설정되는지 알 수 없다. 이러한 문제점으로 인하여 자동적인 클립파일의 생성이 어렵다. 따라서 불연속 미디어는 파일 생성자가 어떠한 타이틀을 작성할 때에 연속 미디어와 비교해 가면서 편집을 하고, 이렇게 편집된 파일이 클립파일로 생성되도록 해야 한다.



ls : logical start time, le : logical end time
w : width of frame, h : height of frame

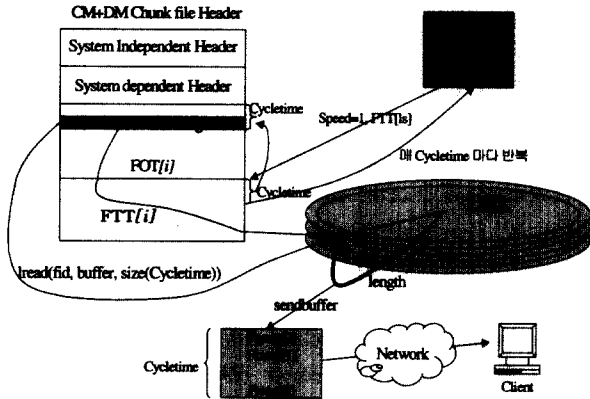
(그림 10) jpeg 불연속 미디어 프레임 포맷

다음 (그림 11)은 불연속 미디어 클립파일이 생성되는 전체 절차를 표현한 것이다.



(그림 11) 편집기의 역할

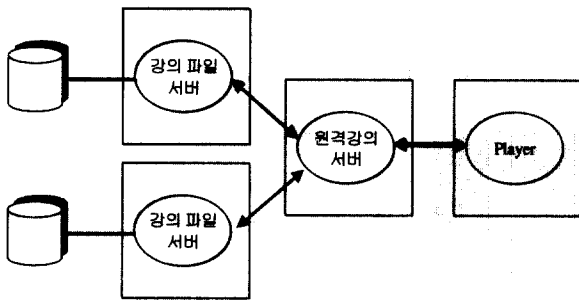
4.5 확장-CMT기반의 LTS 구조



(그림 12) LTS동작 구조

4.6 원격 강의의 서비스 실현

시스템 아키텍처는 강의 파일서버와 클라이언트, LOD 서버 등으로 구성되어 있다.



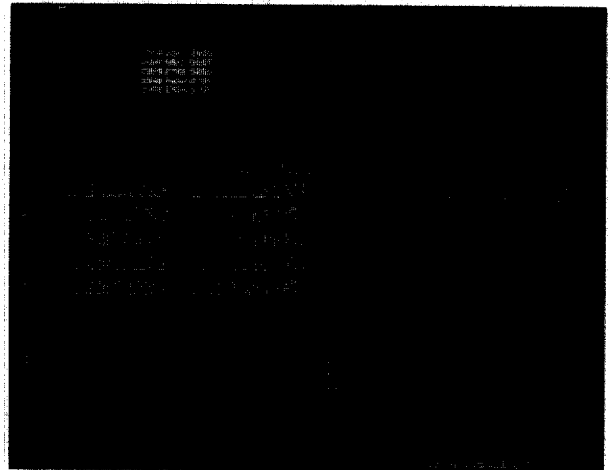
(그림 13) 시스템 구성도

본 논문에서 구현하고 있는 원격강의 시스템은 진행 논리시간이 변경될 때 적합한 시간과 데이터의 위치를 계산하는 모듈, 음성의 소리 조절 모듈, LTS, 패킷 소스/목적지에 관한 모듈, 파일 처리 모듈 등으로 구성되어 있다.

본 시스템의 VOD 서버는 등록된 클라이언트의 정보를 관리하며, 분산 저장되어 있는 강의 파일서버의 상태를 파악하여 클라이언트로부터의 서비스 요청 및 새로운 타이틀의 전송시 적당한 강의 파일서버와의 연결을 통해 원활한 서비스가 이루어지도록 보장하고 있다.

그리고, 강의 파일서버는 이미 저장되어 있는 강의 타이틀과 새로 등록된 강의 타이틀을 보관하며 조회 횟수가 많은 강의 타이틀의 경우 여러 강의 파일서버에 분산 저장/보관하는 기능을 수행한다.

한편, LOD 시스템의 기본 구조는 클라이언트-서버를 기본으로 하며 그 구성 요소로는 강의 타이틀을 제공하며 검색할 수 있는 클라이언트와 클라이언트에 의하여 제공되거나 이미 만들어져 있는 강의 타이틀을 보관하고 있는 강의 파일서버, 그리고 여러 강의 파일서버의 상태를 파악하여



(그림 16) 확장-CMT기반으로 구현된 원격강의 시스템의 예

적정 수준의 서비스를 제공시켜 주는 LOD 서버로 구성되어 있다.

WWW에서 프로그래밍을 하는 방법에는 C를 통한 CGI 프로그래밍과 Perl 스크립트를 이용한 CGI 프로그래밍, 그리고 Java를 통한 프로그래밍이 있다.

첫 번째로 C를 통한 CGI 프로그래밍은 그 방법에서 일반적으로 개발자들에게 익숙한 프로그램 언어라는 장점이 있으며, 두 번째로 Perl 스크립트를 이용한 CGI 프로그래밍은 스크립트 언어의 장점을 살려서 C로 구현하는 것 보다 쉽게 구현할 수 있는 장점이 있다. 세 번째로 인터넷상에서 자바를 통한 프로그래밍은 객체 지향을 기본으로 하고 있는 언어로써, 자바의 많은 장점으로 많이 사용하고 있다.

5. 결 론

기존의 상용 주문형 멀티미디어 서비스의 제공을 위한 소프트웨어는 분산된 컴퓨터 네트워크를 고려하고 있지 않지만 버클리 VOD시스템은 분산된 유닉스 컴퓨터 네트워크 환경하에서 TCP/IP와 UDP/IP를 사용하고 스토리 보드 개념과 LTS을 활용해 동기화 및 비디오 제어 기능을 제공하며, 분산된 여러 형태의 비디오 클립파일들과 오디오 클립 파일들로 편집하여 한편의 VOD서비스를 제공할 수 있는 고도의 기능을 갖고 있는 시스템이다.

특히 본 논문에서 기존의 비디오와 오디오만을 다룰 수 있는 시스템 개발 도구인 CMT를 불연속 미디어인 이미지, 텍스트, 그래픽 등을 다룰 수 있도록 Windows NT 기반으로 CMT를 확장하였으며, 결과적으로 보다 효과적인 주문형 멀티미디어 서비스를 개발할 수 있는 기틀을 마련하였다<표 4>.

CM Toolkit을 확장하여 다음과 같은 실험을 하여서 기존의 성능과 차이점이 없음을 확인하였다. 첫째로 기존의 CMT에서 동영상과 음성을 로컬 및 분산환경에서의 재생할 때에 동기화 및 미디어의 질의 수준과 불연속 미디어

〈표 4〉 확장 기능 정리

비교 항목	기존 CMT	확장 CMT
미디어의 범위	동영상, 오디오에 국한	동영상, 오디오, 이미지, 텍스트, 3D 이미지로의 확장
시스템	유닉스 기반	유닉스, 윈도우, 리눅스 기반으로의 확장
미디어 동기화	동영상과 오디오의 동기화	동영상, 오디오, 이미지, 텍스트, 3D 이미지로의 확장
미디어 송수신	동영상, 오디오 송수신 모듈	동영상, 오디오, 이미지, 텍스트 송수신 확장
네트워크 기술	동영상의 프레임 조절	연속적인 미디어와 불연속적인 미디어의 QoS 설정 기능 확장
데이터 타입	동영상, 오디오 타입 지원	동영상, 오디오, 이미지, 텍스트, 그래픽 관련 타입 지원
동기화	동영상과 오디오 동기화 지원	동영상, 오디오, 이미지, 텍스트, 그래픽 동기연진 설계 및 구현

처리의 로컬 및 분산처리를 평가하였다. 두 채로 기존의 방법과 연속 미디어(음성, 동영상)와 불연속 미디어(이미지)를 동시에 재생할 때의 동기화 및 미디어의 질의 수준을 평가한 결과 동영상의 화상의 약간의 끊김현상이 일어나는 것 이외에 기존의 수준을 유지하였다.

확장된 CMT를 기존망 뿐만 아니라 차세대 망에서도 서비스할 수 있도록 연구가 요망되며, 현재는 소프트웨어로 되어 있는 비디오 압축과 복원의 부분을 MPEG4등의 보다 효과적인 방법으로 개선되어야 할 것이다.

참 고 문 헌

[1] J. Escobr, D. Deutsch, C. Partridge, "Flow Synchronization Protocol," Proc. of the IEEE Globecom. Vol.3, pp.1381-1387, 1992.
 [2] T. D. C. Little, F. Kao, "An Intermedia Skew Control System for Multimedia Data Presentation," Proc. of the 3rd Int. Workshop on Network and Operating System Support for Digital Audio and Video, 1992.
 [3] D. P. Anderson and G. Homsy, "A Continuous Media I/O Server and Its Synchronization Mechanism," IEEE Computer, Vol.32, No.10, pp.51-57, Oct. 1991.

[4] Brian C. Smith, "Implementaion Techniques for Continuous Media System and Application," UC Berkeley, 1994.
 [5] Rowerance A. Rowe, "Introduction to the CM Toolkit," UC Berkeley, 1994.
 [6] W. B. Pennebaker, JPEG still image data compression standard, Van Nostrand, New York, 1992.
 [7] G. K. Wallace, The JPEG Still Picture Compression Standard, Communications of the ACM, Vol.24, Num, pp.30-44, April 1991.
 [8] B. W. Kernighan, D. M. Ritchis. The C Programming Language. Englewood Cliffs, N.J., Prentice-Hell, 1978.
 [9] L. A. Rowe., "Continuous Media Applications" Multipoint Workshop held in conjunction with ACM Multimedia 1994, San Francisco, CA, November 1994.
 [10] L. A. Rowe and B. C. Smith, "A Continuous Media Player," Proc. 3rd Int Workshop on Network and Operating System Support for Digital Audio and video, San Diego, CA, November 1992.
 [11] Tony King, "Pandora : An Experiment in Distributed Multimedia," proceedings of Eurographics '92.
 [12] 박순은, 윤효상, 박세준, 이근왕, 오해석, "ATM-LAN 환경하에 서 주문형 강의 시스템 구현", '97 한국정보처리학회 춘계 학술발표 논문집, 제4권 제1호, pp.771-774.



김 용 대

e-mail : ydkim@omega.sunmoon.ac.kr

1970년 서울대학교 공과대학 응용수학과 (공학사)

1988년 서울대학교 자연과학대학 계산통계학과 전산과학 전공(이학석사)

1993년 서울대학교 자연과학대학 계산통계학과 전산과학 전공(이학박사)

1973년~1976년 Fujitsu Korea(주) OS부

1977년~1984년 현대건설(주) 전산자료실

1984년~1991년 현대전자산업(주) 정보시스템 사업부 상무

1991년~1994년 선경정보시스템(주) SI 사업부 전무

1995년~현재 선문대학교 자연과학대학 컴퓨터정보학부 교수

관심분야 : 병렬처리, 멀티미디어, 네트워크 관리