

분산 멀티미디어 환경에서의 공동 저작

성 미 영

인천대학교 전자계산학과

Joint Authoring in Distributed Multimedia Environments

Mee Young SUNG

Department of Computer Science, University of Incheon

요 약

본 논문에서는 분산 환경에서 멀티미디어 문서를 공동 저작하는 시스템을 설계하는 데 있어서 고려해야 할 요구 사항들을 분석하였다. 또한 이들 요구 사항을 만족시키는 문서 구조 DMDA (Distributed Multimedia Document Architecture)의 기본 개념을 소개하였다. DMDA는 논리 구조 지향적이다. 표현 객체와 상대 공간 합성의 도움으로 동적인 재배치가 가능하다. 또한 사용자 관점에 따른 문서 출력이 가능하고 버전 관리가 쉬운 구조로 되어 있다. 공동 작업 중에 문서 위에 행해진 표시나 주석은 표시 객체에 담겨 분산된다. 분산 멀티미디어 환경에서 문서를 편집 또는 공동 저작하는 시스템은 분산 객체들을 조합하여 하나의 문서를 만들 수 있기 위하여 논리 구조 지향적이고 구조 편집 지향적이어야 한다.

1. 서론

워크스테이션과 개인용 컴퓨터들로 구성된 네트워크를 기반으로 하는 현재의 컴퓨팅 환경에서는 공간적으로 떨어져 있는 사람들의 공동 작업(CSCW; Computer Supported Cooperative Work)이 훨씬 쉬워졌다. 최근 들어 급속도로 발전된 멀티미디어 처리 기술은 분산 환경에서의 멀티미디어 공동 작업이 가능하게 하였다. 미래의 기본적인 사무용 소프트웨어는 바로 책상 위에 있는 워크스테이션이나 개인용 컴퓨터에서 멀티미디어 통신을 통해 문서나 응용 소프트웨어들을 공유하는 그런 형태가 될 것이다.

분산 멀티미디어 환경에서 컴퓨터를 이용하여 할 수 있는 공동 저작으로는 공동 문서 저작, 공동 영화 제작, 공동 설계, 공동 작곡 등을 들 수 있다. 이 중 문서 공동 저작은 신문, 기술 보고서, 매뉴얼, 편지 등을 공동으로 저작하는 것을 포함하며 그 수요가 가장 많다. 오늘날 문서는 단순히 종이를 전자화한 것이 아니고 동적이고 모듈화되어 있고 멀티미디어적인 대상이다. 문서에 소리와 비디오 자료를 집어넣는 일이 가능한 지금의 전자 문서는 의사소통을 위한 모든 정보(미디어) 형태를 포함하고 있다. 멀티미디어 문서는 분산 시스템에서의 의사 전달의 수단이 될 뿐 아니라 공동 작업에서의 공유 작업장으로도 이용될 수 있다.

본 논문에서는 주로 멀티미디어 문서의 공동 저작 시스템을 설계하는 데 있어서 고려해야 할 사항들에 대해 논의하고자 한다. 이들 고려 사항들은 공동 영화 제작, 공동 설계, 공동 작곡 등의 문제에 있어서도 많은 부분이 공통적으로 적용될 수 있을 것이다. 다음절로부터 차례로 공동 작업, 문서 모델과 문서 관리, 공동 저작 시스템 설계를 위한 요구 사항, 분산 멀티미디어 문서 구조, 멀티미디어 문서 편집을 살펴보고 결론을 맺겠다.

2. 공동 작업(Collaborative Work)

분산 시스템은 서로 다른 종류의 컴퓨터들이 LAN 또는 WAN에 연결되어 서로 소통하는 시스템이다. 비디오 화면을 통해 대화하는 동시에 같은 문서나 도면을 화면상에서 공유하며 문서의 특정 부분을 지적 또는 수정하며 토론하는 상황을 우리는 분산 공동 작업이라 말할 수 있다. 공동 작업은 동기적 공동 작업(synchronous CSCW)과 비동기적 공동 작업(asynchronous CSCW)으로 나뉘어진다. 동기적 공동 작업은 공간적으로는 분산되어 있더라도 시간적으로는 동시에 실시간(real-time)으로 작업하는 경우이고 비동기적인 공동 작업은 시간적으로 분산되어 작업하는 경우이다. 공동 작업의 모드는 시간과 공간이 같고 다름에 따라 동기(synchronous), 분산 동기(distributed synchronous), 비동기(asynchronous), 분산 비동기(distributed asynchronous)의 네 가지 시공 모드로 구분된다[Palmer 94].

분산 멀티미디어 환경에서의 공동 작업 도구로(groupwares)는 비디오 회의(video conference), 공동 저작(joint authoring), 의사결정 지원(decision support), 전자 우편(electronic mail), 전자 게시판(computer bulletin board) 등을 들 수 있다. 이들 공동 작업 도구들에서 주로 사용되는 시공 모드를 분석해 보면 표 1과 같다[Groudin 94, Palmer 94].

비디오 회의(video conferencing) 도구는 다른 공동 작업 도구와 병행하여 사용될 수 있는 중요한 도구로서 기본적으로 비디오 자료와 소

리 자료를 교환하고 있다. 몇몇 비디오 회의 시스템에서는 화면의 일부가 모든 사용자에게 복사되는 가상 공유 작업 장(shared virtual work space)[Furht 94]으로 문서 또는 화이트 보드를 이용하고 있다[Beadle 95, Gidney 94].

	동기	분산 동기	비동기	분산 비동기
그룹의사결정 지원 시스템	x	x		
비디오 회의		x		
공동 저작	x	x	x	x
전자 우편				x
전자 게시판				x

표 1. 공동 작업 도구들의 시공 모드

분산 공동 작업 시스템에서는 자연스럽게 사용이 편리한 사용자 인터페이스가 절실히 요구되고 있다. 모든 조직의 의사 소통은 문서를 통해 이루어지고 있다. 그러므로 분산 시스템에서의 공동 작업을 위한 가장 자연스러운 사용자 인터페이스는 기존의 종이 문서를 기반으로 하며 음성과 비디오 정보가 추가된 멀티미디어 문서의 형태라 할 수 있다.

최근 들어 MMConf(Diamond System), RTCal(Real Time Calculator, MIT), Rapport(AT&T), MIAS(Multipoint Interactive AudioVisual System, European collaborative project), Person-to-Person(IBM), ShareVision(Creative) 등의 많은 비디오 회의 시스템들이 만들어지고 있으나 이들은 단지 비디오와 소리 자료만을 전용선을 통해 전송하거나 문서를 전송하더라도 문서를 페이지 단위의 이미지로 전송하고 있기 때문에 문서를 텍스트로 전달하는 것보다 비용이 더 많이 든다.

공동 저작(joint authoring) 도구는 모든 시공 모드에서 이용될 수 있는 중요한 도구이다. 공동 저작 작업은 기본적으로 편집 가능한 상태의 문서를 공유할 수 있어야 한다. 그리고 비디오 회의 도구와 결합되어 비디오와 소리 자료로도 생각을 전달할 수 있어야 하겠다.

3. 문서 모델과 문서 관리

멀티미디어 문서는 공간 개념만이 존재하는 문자, 사진, 또는 그래픽 자료들로 구성된 문서를 기본 골격으로 하고 이 문서에 시간 개념이 있는 소리 자료나 시간과 공간 개념이 공존하는 애니메이션 또는 비디오 자료를 포함(inclusion)시킨 것이다. 멀티미디어 문서 모델의 예로는 ODA(Open Document Architecture) [ODA 93] 모델에서 출발한 MINOS [Christodoulakis 86], Multos [Bertino 86] 그리고 Muse [Gibbs 87] 등과 MAEStro [Drapeau 91]를 들 수 있으며 최근들어 정의되고 있는 HyperODA [ODA 93], HyTime [Newcomb 91] 등은 하이퍼텍스트 기능이 추가된 모델이다.

미래의 문서들은 분산 시스템 상에서 서로 교환되어야 할 필요가 있으며 문서들은 기본적으로 서로 다른 문서 시스템에서 만들어진다고 전제해야 한다. 그러므로 분산 시스템에서의 공동 작업을 가능하게 하려면 우선 그 문서 구조는 국제 표준과 호환성이 있어야 한다. 문서 구조의 국제 표준은 ISO에서 제정한 ODA(Open Document Architecture, ISO/IEC 8613-1989)와 SGML(Standard Generalized Markup Language, ISO/IEC 8879-1986)이 있다.

ODA의 가장 큰 특징은 문서의 내용(content)과 구조(structure)를 완전히 분리한 것이다. 또한 한 문서 대해 두개의 구조를 가지는 것도 중요한 특징이다. ODA의 두 구조중 하나는 문서를 논리적 관점(logical view), 즉 장(chaper), 절(section), 문단(paragraph) 등으로 분석한 논리 구조(logical structure)이고 다른 하나는 배치 관점(layout view), 즉 종이 등 출력 매체에 어떻게 문서 내용들을 배치할 것인지 분석한 배치 구조(layout structure)이다. 여기서 배치 구조는 문서의 논리적인 내용을 페이지 단위로 보여 주기 위한 것이다.

또한 ODA는 원형 구조(generic structure)와 특정 구조(specific structure)를 구분하고 있으며 원형 구조는 기본적으로 계층적인 트리 구조이며 SEQ(sequence), AGG(aggregate), CHO(choice) 등의 구조 생성자(construction operator)를 이용해 특정 구조 생성을 제어한

다. 이들 연산자들은 OPT(optional), REP(repeatitive), OPT REP와 같은 생성 인자(construction factor)들과 결합될 수 있다.

SGML도 역시 문서의 내용과 구조를 분리하고 있다. 문서 구조는 TeX이나 troff 등에서처럼 태그(tag)로 표현되는 마크업 언어(markup language)로 명세된다. SGML 문서의 묘사 자체는 내용과 구조가 혼합되어 있는 형태이다. 문서 구조는 명세가 파서에 의해 해석되었을 때 분리된다. SGML에서도 태그를 이용하여 문서 생성의 원형 구조를 선언할 수 있으나 배치 구조는 없고 논리 구조만이 존재한다. 배치는 배치기(formatter)가 태그를 해석하여 포맷된 문서를 만들어 낸다.

여러 문서 모델들을 논리 구조와 배치 구조가 있고 없음에 따라 구분해 보면 다음과 같다.

- 논리 구조와 배치 구조 모두 있는 모델
ODA와 [Karmoush 90], [Christodoulakis 86], [Gibbs 87] 등의 모델이 있다.
- 논리 구조만 있는 모델
SGML과 [Woelk 86]의 모델을 들 수 있다.
- 배치 구조만 있는 모델
ODA의 전신인 CCITT (International Telegraph and Telephone Consultative Committee)의 T73 모델 등 과거의 많은 경우에 해당한다.

한편, 문서 모델의 결정은 문서 관리의 형태에 크게 영향을 받는다. 문서 관리는 우선 동적인 문서 관리(dynamic document management)와 이미지 문서 관리(Image document management)로 구분될 수 있다 [Reinhardt 94]. 동적인 문서 관리는 동적인 문서 즉, 편집 가능하고 내용 찾기(content search)를 할 수 있는 문서들을 등록하고 자신의 컴퓨터 또는 다른 서버에 저장되어 있는 동적인 문서를 찾아오는 일에 초점을 맞추고 있다. 이미지 문서 관리는 주로 다른 시스템에서 만들어진 문서들의 비트 맵 이미지를 입력하고 그 이미지에 꼬리표를 붙여 저장하며 그 이미지 문서를 찾아오는 일을 관리한다.

또 하나의 새로운 유형의 관리로는 문서와 데이터 베이스를 결합한 형태의 문서들을 관리하는 문서 조합 관리(document assembly management)를 들 수 있다. 문서 안의 표 내용이 스프레드 시트 또는 데이터 베이스의 내용으로부터 결정되는, 문서가 만들어진 상태로 고정되는 것이 아니라 거대한 정보 베이스의 순간 순간의 스냅 사진(snap-shot)으로서 존재하는 문서를 대상으로 한다. 예를 들면 문서 안의 도표의 값이 순간 순간 바뀌는 그런 문서들을 말하며 본인은 계산 자료(calculable data)를 가진 문서라 부른다[Sung 90].

우리의 관심은 저작이므로 문서 모델 설계에 있어서 편집 가능한 문서를 대상으로 하는 동적 문서 관리를 잘 할 수 있도록 배려해야 한다. 또한 분산 데이터 베이스의 내용을 동적으로 반영할 수 있도록 문서 조합 관리도 고려하여야 한다.

미래에는 문서 관리라는 말이 화일 관리, 데이터 접근, 데이터 프레젠테이션이라는 말들과 동의어가 될 것이다. 문서 관리기 자체는 바로 문서 관리의 사용자 인터페이스가 된다. 말하자면 사용자가 처음 만나게 되는 것은 문서 관리기이고 편집할 문서 화일을 선택한 다음 문서 편집기를 수행시키게 된다. 또한 문서 화일을 저장시키면 저자, 제목, 주제, 키워드, 요약문, 생성 날짜, 버전과 전문 탐색(full-text retrieval)을 위한 인덱스 등이 profile에 수동 또는 자동으로 저장되어야 한다.

4. 공동 저작 시스템 설계를 위한 요구 사항

본 절에서는 분산 시스템에서의 멀티미디어 문서 공동 저작 시스템 설계를 위한 요구 사항을 살펴본다. 본 논문에서 논의하는 멀티미디어 문서는 기본적으로 편집이 가능한 문서이다. 또한 하이퍼미디어 문서의 기본 골격이 될 뿐만 아니라 공동 작업을 잘 지원하기 위한 기법들이 포함되어야 할 것이다. 공동 작업을 위해서는 여러 가지 기능들이 요구된다. 지적 또

는 표시(pointing or marking)를 할 수 있어야 하며 주석을 다는 기능이 있어야 할 것이다. 주석을 단 문서는 그 문서의 나 자신의 새로운 버전이 된다. 이는 곧 문서 관리자가 문서의 원본과 그 문서를 언제 누가 수정했는가를 알고 있어야 함을 뜻한다.

분산 멀티미디어 문서 구조는 앞에서 언급한 기본적인 기능을 잘 지원할 뿐 아니라 사용자 자신의 버전을 만들고 유지하기 쉽도록 만들어져야 하며 사용자 자신의 고유한 관점으로 문서를 출력(presentation)해 볼 수 있는 구조가 되어야 할 것이다. 뿐만 아니라, 정보가 일정한 체계를 갖추고 저장되어 있는 일반 데이터 베이스와는 달리 문서는 그 자체가 하나의 정보 덩어리이다. 이러한 문서에서 정보를 얻어내기 위해서는 전 내용 탐색(full-content search)이 필수적이며 이를 쉽게 할 수 있도록 해주는 구조가 되어야 한다.

분산 멀티미디어 문서의 편집/공동 편집/공유/저장/검색/관리를 잘 지원하는 시스템을 설계하는 데 있어서 고려해야 할 기술적 요구 사항들을 아래에 정리한다.

- 개별 미디어 편집 및 통합 편집이 가능하여야 한다.
- 문서에 포함된 계산 자료(calculable data)의 편집 및 처리가 가능하여야 한다.
- 문서 안에 분산 데이터 베이스 객체들을 포함할 수 있어야 한다.
- 편집된 내용은 네트워크를 통해 분배되어야 한다.
- 국제 표준과의 호환성이 있어야 한다.
- 원격 지적(tele-pointing), 원격 표시(tele-marking), 원격 편집(tele-editing)등의 공동 작업을 위한 기법들이 지원되어야 한다.
- 전 내용 탐색(full-content search)이 가능하여야 한다.
- 문서 베이스(document base) 구축이 가능하여야 한다.
- 사용자 관점(user views)에 따른 문서 표현

(presentation)이 가능하여야 한다.

- 버전 관리(version control)가 가능하여야 한다.
- 접근 권한에 따라 문서 접근을 제어(access control)할 수 있어야 한다.

5. 분산 멀티미디어 문서 구조

본 절은 새로운 문서 모델 분산 멀티미디어 문서 구조 DMDA(Distributed Multimedia Document Architecture)에 대한 기본 개념을 소개하는 장으로서 DMDA의 특징과 문서 처리 모델, 그리고 문서 공유 기법에 대해 알아 본다.

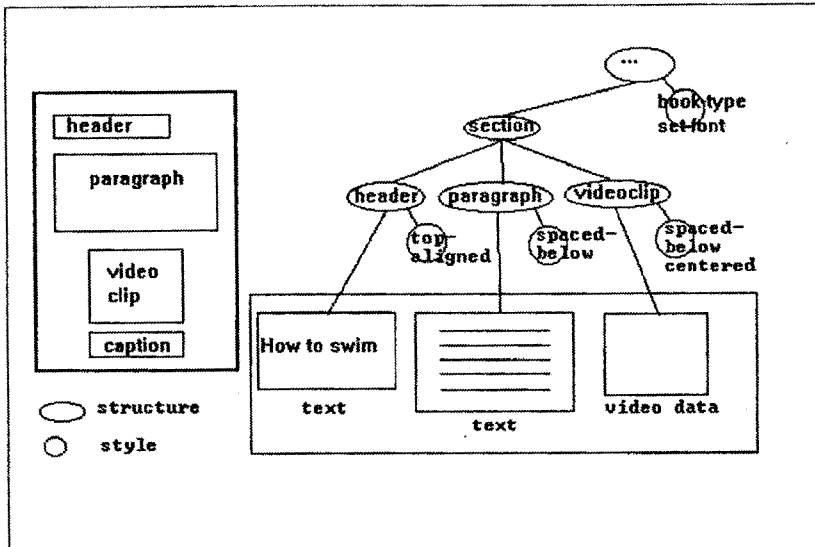
5.1. DMDA의 특징

분산 멀티미디어 문서 구조 DMDA의 특징을 다음에 요약해 본다.

- ODA와 호환
앞 절에서 살펴 본 바와 같이 분산 환경에서의 문서 구조는 국제 표준과 호환성이 있어야 한다. 본 연구는 두 가지 문서 구조 국제 표준 ODA와 SGML 중 ODA를 채택 하였다. 이유는 ODA에서는 의미적으로나

표현 상으로나 구조와 내용을 완벽하게 분리하고 있고 문서 생성에 있어서 필요한 구조 생성 연산자 SEQ, AGG, CHO, OPT, REP, OPT REP들을 지원하고 있기 때문이다. 이 연산자들은 클래스로부터 인스턴스들을 생성할 때 이것이 없는 SGML에서 보다 훨씬 강력하다.

- 논리 구조와 표현 스타일 객체 (presentation style object)
그림 1에서 보는 것처럼 DMDA는 논리 구조와 배치 구조를 구분하지 않는다. 단지 논리 구조만 있고 배치에 관한 정보는 표현 스타일 객체에 담아 표현 처리 과정에서 이용한다. 표현 스타일 객체들은 논리 구조의 계층 구조에서 준하여 속성을 계승받는다.
- 동적 배치(dynamic layout)
각각의 논리 구조 객체에 연결된 표현 스타일 객체안의 정의가 변화함에 따라 한 문서 내용을 여러 모양으로 표현(presentation)할 수 있다. 표현 스타일에는 페이지 단위의 출력, 섹션 단위의 출력, 전 문서의 스크롤 등의 배치 모드(layout mode)와 신문, 책, 논문, 보고서, 편지, 사진, 공문, 시나리오 등의 문서 타입이 정의될 수 있으며 각 문서 타입 고유의 표현 객체 속성에 따른 배치가 일어난다. 이를 위해서는 6.3 절에서



논의할 상대 공간 합성(relative spatial composition)이 필요하다. 만일 책 모양(book design) 데이터 베이스를 유지할 수 있다면 표현 객체에 정의된 속성은 책 모양 데이터 베이스와 연계되어 문서 전체를 동적으로 재배치하도록 운영할 수 있을 것이다.

- 표시 객체(mark object)
문서 위에 어떤 표시를 하거나 간단한 글 또는 그림을 그린 내용은 표시 객체에 담겨진다. 표시 객체는 이를 다른 공유자에게 전송되기도 하고 나중에 다시 참조할 수 있기 위해 표시가 행해진 논리 객체에 연결된다.
- 프로파일(profile)
화일을 저장시키면 저자, 제목, 주제, 키워드, 요약문, 생성 날짜, 버전과 전문 탐색(full-content retrieval)을 위한 인덱스 등이 프로파일에 수동 또는 자동으로 저장된다.
- 사용자 관점
문서의 논리 구조는 사용자가 원하는 섹션만 읽거나, 그림들만 보거나, 제목에 특정 단어가 들어 있는 부분만 살펴보는 것을 가능하게 한다.
- 버전 관리(version control)
문서의 논리 구조는 버전 관리를 쉽게 해준다. 문서 내용의 일부가 수정되었을 경우 문서 전체의 새로운 복사본을 만들 것이 아니라 단지 바뀐 부분에 해당하는 논리 객체의 내용으로의 연결만을 새것으로 바꾸면 된다.
- 접근 제어(access control)
분산 시스템에서 문서를 공유할 때 가장 문제가 되는 것이 보안 문제이다. 문서의 논리 구조는 문서 전체 뿐 아니라 각 논리 객체 단위로 접근 권한을 부여할 수 있게 한다. 또한 논리 객체 단위로 저자 이름을 부여하여 저작권(authority)도 제어할 수 있다.

5.2. 문서 처리 모델

본 논문에서 논의하는 전자 문서의 처리 과정은 크게 아래의 세 단계로 나눌 수 있다.

- 편집 처리(editing process)
문서의 내용, 구조, 그리고 구성 요소들의 관계를 기록하고 수정하는 과정이다.
- 저장과 검색 처리(store and retrieval process)
분산 환경에서 문서나 문서 구성 요소들을 저장(storing), 분배(distribution), 교환(exchange), 그리고 검색(retrieval) 하는 과정이다.
- 표현 처리(presentation process)
편집된 문서의 내용을 사람이 인식할 수 있는 매체(컴퓨터 화면, 종이, MIDI 합성기 등)로 보여주는 과정이다.

위의 처리 모델은 멀티미디어 문서 편집기, WYSIWIS(What You See Is What I See) 문서 공동 저작 도구(collaborative document authoring tool), 문서 베이스 서버(document base server), 하이퍼미디어 검색기(browser) 등의 도구를 필요로 한다.

5.3. 문서 공유 기법

분산 시스템에서 문서 등의 객체 또는 편집기 등의 응용 프로그램을 공유하는 데는 아래에서 보는 바와 같이 네 가지 방법이 있다[Hodges 93].

- 프로토콜 복사(protocol replication)
주 공유자만이 프로그램을 수행시키며 프로그램 수행 과정에서의 결과들만이 결과 분배기(output distributor)에 의해 다른 공유자들에게로 복사된다.
- 이벤트 복사(event replication)
공유자들 각자가 같은 프로그램을 수행시키고 마우스 클릭이나 키보드 입력 등의 입력 이벤트만을 입력 분배기(input distributor)에 의해 모든 공유자에게 복사한다.
- 이미지 복사(image replication)
주 공유자만이 프로그램을 수행시키며 주 공유자의 스크린 이미지가 이미지 분배기

(image distributor)에 의해 다른 공유자들에게 연속적으로 보내진다.

• 객체 공유(shared objects)

문서가 바로 분산 객체 데이터 베이스(distributed object database) 안의 객체들을 조합하여 만들어지는 하나의 복합 객체인 그런 시스템에서 가능한 문서 공유 기법이다. 각 공유자는 화면의 공유 문서 안의 객체들을 편집하며 편집된 내용은 분산 객체 데이터 베이스에서 갱신되는 대로 다른 공유자에게 보여지게 된다.

우리가 추구하는 기법은 물론 객체 공유 기법이다. 이 기법을 구현하기 위해 가장 우선적으로 해결되어야 할 문제는 문서의 구성 요소들을 객체로 만드는 일이다. 그리고 한 문서는 객체들을 찾아가는 포인터들을 모아 놓은 복합 객체로 만들어져야 한다.

한 문서의 구성 객체들은 다른 시스템에 분산되어 있을 수 있다. 왜냐하면 수백만 바이트의 비디오 클립을 각 시스템에 복사할 이유가 없기 때문이다. 한 문서의 구성 객체들은 장, 절, 문단, 그림 등의 논리적인 단위이어야 하며 페이지 단위는 될 수 없다. 왜냐하면 하나의 문단이 두 페이지에 걸쳐 배치될 수 있고 그 경우 두 페이지를 접근하여 그 일부를 추출하는 불편을 겪어야 하기 때문이다.

6. 멀티미디어 문서 편집기

멀티미디어 문서를 만들기 위해서는 기본적으로 문자, 소리, 화상, 동화상, 영상, 비디오 등의 모노 미디어(mono-media)의 개별 편집이 가능하여야 한다. 그리고 우리가 생각하는 바를 효과적으로 표현하기 위하여 각 미디어 자료들이 적절하게 조합되어야 하며 이들이 저장되거나 출력되어야 할 필요가 있다. 여러 모노 미디어 자료들의 조합을 만들어 내는 편집이 가능할 때 비로소 진정한 의미의 멀티미디어 정보를 만들어 내는 멀티미디어 편집인 것이다.

기존의 문서 개념에서 탈피해서, 문서를 컴퓨터가 제공하는 다양한 정보들을 인간이 인식할 수 있는 모양으로 통합(integration) 또는 합성(composition)하여 보여주는 사용자 인터페이스라고 정의해 볼 수 있다. 개별 미디어 편집기에 의해 만들어진 미디어 자료들을 통합하기 위해서는 당연히 통합 편집기가 필요하다. 바로 통합의 실체가 문서이므로 멀티미디어 문서 편집기가 바로 통합 편집기가 된다. 멀티미디어 통합 편집기를 설계할 때 요구되는 사항에는 공간 합성(spatial composition), 시간 합성(temporal composition), 사용자와의 상호작용(user interaction), 쉽고 친절한 사용자 인터페이스(easy and friendly user interface), 확장성(extensibility) 등이 있다[성 94].

본 논문에서 논의하는 문서 구조는 문서를 내용 위주로 편집하는 기존의 방식과는 달리 구조 지향적 편집(structure-oriented editing)을 하기를 요구하고 있다. 즉, 우선 문서의 구조를 편집하고 나서 구조의 각 구성 요소들에 대한 편집을 한다. 문서 타입 별 문서 구조가 미리 마련되어 있다면 이들 문서 구조를 이용하여 그 구조에 맞게 내용을 편집한다. 반대로 여러 사람들이 각자의 주제에 대한 내용을 편집하고 난 다음 이들을 종합하여 전체 문서를 통합하는 차원에서 문서 구조 편집을 할 수도 있다. 이러한 문서 편집 방식에서는 구조 편집기와 내용 편집기를 구분하여야 한다.

위의 사실로부터 멀티미디어 통합 편집 작업을 크게 구조 편집(structure editing), 내용 편집(content editing), 공간 합성 편집(spatial composition editing), 그리고 시간 합성 편집(temporal composition editing) 작업으로 나누어 볼 수 있다. 이들 작업들의 동작 메커니즘을 간단히 살펴본다.

6.1. 구조 편집

문서의 논리 구조는 그림 1에서 보는 바와 같이 트리 구조로 되어 있다. 구조 편집은 문서의 논리적 계층 구조가 잘 파악이 될 수 있도록 편집 중인 문서 구조의 트리를 방문하면서 내부 트리 구조의 상태를 그대로 보여 주어

야 한다. 트리 구조는 6.4 절에서 소개할 MediaNet 편집과 유사하게 그래픽 도구를 이용하여 만들 수 있다. 또한 만들어진 트리 구조 그림은 그대로 그 트리의 내부 표현으로 바뀔 수 있다. 표현 스타일 객체도 같은 방법으로 편집되어 문서 구조 트리에 연결시킨다.

6.2. 내용 편집

개별 미디어 내용은 별도의 윈도우에서 개별 미디어 편집기를 수행시켜 만들어진다. 만들어진 내용은 문서 트리 구조의 각 객체에 연결시킨다.

6.3. 공간 합성 편집(Spatial Composition Editor)

구조와 내용으로 구성된 멀티미디어 문서는 최종적으로 사용자가 인식할 수 있는 형태로 만들어 주어야 한다. 이를 위해서는 문서 틀에서의 공간 합성 편집 기능이 필요하다. 공간 합성을 하는 방법에는 상대 공간 합성(relative spatial composition)과 절대 공간 합성(absolute spatial composition)의 두 가지가 있다.

- 상대 합성
객체들을 right-of, left-of, top-aligned,

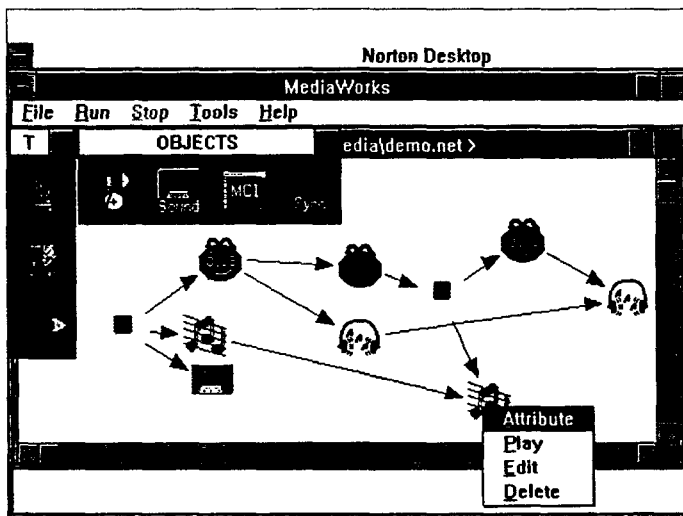
spaced-below, centered 등과 같이 다른 객체에 대해 상대적인 배치 속성을 정의함으로써 공간 합성을 할 수 있다. 이들 합성을 위한 속성들은 그림 1에서와 같이 표현 스타일 객체에 정의되며 문서 표현(presentation)을 요구한 시점에서 그 때의 배치 모드 등 다른 속성들과 결합되어 공간에 배치된다.

- 절대 합성
각 객체의 물리적인 좌표 값을 정의함으로써 공간 합성을 한다.

5.1 절에서 논의한 동적인 배치가 가능하려면 상대 합성이 가능하여야 한다. 상대적 위치 정보가 있어야만 이를 이용하여 객 모양 데이터 베이스의 원형(template)에 따라 해당 논리 객체들을 배치할 수 있다.

6.4. 시간 합성 편집(Temporal Composition Editor)

시간 합성을 시각화 해주는 사용자 인터페이스로는 그림 2에서 보는 바와 같이 비순환 방향 그래프(Directed Acyclic Graph: DAG)에 기반을 둔 작업 네트워크 형태의 MediaNet을 이용한다[성 94]. MediaNet 편집기에서는 각각의 미디어 자료들을 아이콘으로 형상화하여 각 아이콘들을 서로 의미 있고 적절하게 연결



함으로써 여러 가지 미디어 자료들의 논리적인 출력 순서를 지정할 수 있게 해 준다. MediaNet 편집기는 소리나 동화상 같은 개별적인 미디어들의 표현 순서를 관리하며 동시에 두개 이상의 미디어를 표현해 주는 기능과 이 표현을 해석하여 수행시켜 주는 기능이 갖추어져 있다.

사용자 인터페이스로서의 MediaNet의 가장 중요한 장점은 사용자가 머리 속에 구상한 미디어 자료들의 논리적인 출력 순서가 그대로 MediaNet에 표현되는 것이다. 그리고 이 표현은 다른 내부 표현으로의 변화 과정 없이 그대로 해석되어 실행되어진다. MediaNet은 멀티미디어 문서의 시간 합성 기법일 뿐 아니라 멀티미디어 프레젠테이션을 위한 시각 프로그래밍(visual programming) 도구로도 이용될 수 있다.

7. 결론

현재, 문서를 기반으로 하는 좋은 공동 작업 도구가 없는 시점에서 필요한 것은 분산 멀티미디어 시스템에서 공동 작업장으로 이용될 멀티미디어 문서를 편집, 공동 편집, 저장, 검색, 그리고 관리하는 시스템이다. 본 논문에서는 분산 환경에서 멀티미디어 문서를 공동 저작하는 시스템을 설계하는 데 있어서 고려해야 할 요구 사항들을 분석하였다. 또한 이들 요구 사항을 만족시키는 문서 구조 DMDA(Distributed Multimedia Document Architecture)의 기본 개념을 소개하였다.

DMDA는 ODA와 호환성이 있고, 논리 구조 지향적이다. 표현 객체와 상대 공간 합성의 도움으로 동적인 재배치가 가능하다. 또한 사용자 관점에서의 문서 출력이 가능하고 버전 관리가 쉬운 구조로 되어 있다. 공동 작업 중에 문서 위에 행해진 표시나 주석은 표시 객체에 담겨 분산된다. DMDA를 기반으로 하는 시스템의 가장 큰 특징은 분산 멀티미디어 객체들을 조합할 수 있기 위하여 논리 구조 지향적이고 구조 편집 지향적이라는 점이다.

앞으로의 연구는 이 논문에서 소개한 내용에 대한 프로토타입을 완성하고 실험을 통하여 그 유용성을 증명하는 일에 집중될 것이다. 또한 상대 합성을 쉽고 효율적으로 시각화해 주는 메커니즘과 사용자 인터페이스에 대한 연구가 필요하다. 프로토타입은 Sun SparcStation과 PC 사이트들로 구성된 LAN 상에서 클라이언트/서버 기법을 이용하여 구현될 것이다.

본 연구가 완성되면 멀티미디어 신문 등의 공동 저작 뿐 아니라 비디오 회의, 그룹 의사 결정 지원 등의 그룹웨어에서의 공유 작업장으로 이용될 수 있다. 또한 분산 비동기적인 응용인 멀티미디어 전자 우편에도 이용될 수 있다.

References

- [성 94] 성미영, 윤자천, "멀티미디어 프레젠테이션을 위한 시·공간 합성의 시각화", HCI '95 학술대회 발표논문집, 한국정보과학회 인간과 컴퓨터 상호작용 연구회지 4권 1호, 1995.2.17, pp152-161.
- [Beadle 95] P.Beadle, "Experiments in Multipoint Multimedia Telecommunication", IEEE Multimedia, Vol.2, No.2, summer 1995, pp30-40.
- [Bertino 86] E. Bertino, S. Gibbs, F. Rabitti, C. Thanos and D. Tschritzis, "A Multimedia Document Server", Proceedings of Advanced Database Symposium, Tokyo, Japan, Aug.29-30, Information Processing Society of Japan, Tokyo, 1986, pp.123-133.
- [Christodoulakis 86] S. Christodoulakis, M. Thodorodou, F. Ho, M. Papa and A. Patria, "Multimedia Document Presentation, Information Extraction, and Document Formation in MINOS: A Model and a System", ACM Transactions on Office Information Systems, Vol.4, No.4, October 1986, pp345-383.
- [Drapeau 91] G.D. Drapeau, H. Greenfield, "MAEstro - A Distributed Multimedia Authoring Environment", Proceedings on USENIX-Summer'91-Nashville, TN, 1991, pp315-328.
- [Furht 94] B. Furht, "Multimedia Systems:

- An overview", IEEE Multimedia, Vol.1, No.1, Spring 1994, pp47-59, .
- [Gibbs 87] S. Gibbs, D. Tsichritzis, A. Fitas, D. Konstantas, and Y. Yeorgaroudakis, "Muse: A Multimedia Filing System", IEEE Software, March 1987, pp.4-15.
- [Gidney 94] E.Gidney, "CSCW for Film and TV Preparation", IEEE Multimedia, Vol.1, No.2, summer 1994, pp16-26.
- [Groudin 94] J. Groudin, "Computer-Supported Cooperative Work: History and Focus", IEEE Computer, Vol.27, No.5, May 1994, pp19-26.
- [Hodges 93] M. E. Hodges, R. M. Sasnett, "Multimedia Computing, Case Studies from MIT Project Athena", Addison-Wesley, 1993, 302 pages.
- [Karmoush 90] A. Karmoush, L. Orozco-Barbosa, N. D. Georganas, M. Goldberg, "A Multimedia Medical Communications System", IEEE Journal on Selected Areas in Communications, Vol.8, No.3, April 1990, pp325-339.
- [Newcomb 91] S.R. Newcomb, N.A. Kipp, and V.T. Newcomb, "HyTime - Hypermedia / Time-based Document Structuring Language", Communication of the ACM, Vol.34, No.1, November 1991. pp.67-83.
- [ODA 93] International Organization for Standardization, "Information technology - Open Document Architecture(ODA) and Interchange Format - Temporal relationships and non-linear structures(ISO/IEC DIS 8613-14:1993), Geneva, 1993.
- [Palmer 94] J.D. Palmer, "Computer-Supported Cooperative Work", IEEE Computer, Vol.27, No.5, May 1994, pp15-17.
- [Reinhardt 94] A.Reinhardt, "Managing the New Document", Byte, August 1994, pp91-104.
- [Sung 90] M.Y. Sung, "Modelization of Multimedia Documents for Tables, Calculable Date and Mathematicla Formulae", Thesis of Doctorat: INSA (Institut National des Sciences Appliquees) de Lyon, July 10, 1990, 221 pages (in French)
- [Woelk 86] D. Woelk, W. Kim, W. Luther, "An Object-Oriented Approach to Multimedia Databases", Proceedings on ACM SIGMOD Conference on the Management of Data, Washington D.C., May 1986, pp311-324.