

PST를 기반으로 하는 멀티미디어 프리젠테이션 모델

최 해 길[†] · 최 속 영^{††} · 유 관 종^{†††}

요 약

멀티미디어 프리젠테이션에서 미디어의 함축된 의미를 효과적으로 전달하기 위해서는 미디어 데이터간에 존재하는 시간 및 공간적인 정보를 효과적으로 표현하고 처리할 수 있는 기법이 필요하다. 특히 이들 시간 및 공간 정보는 상호간 긴밀히 연결되어 있기 때문에 종합적인 표현 방법이 요구된다. 따라서 본 논문에서는 시공간의 동기화 정보를 종합적으로 표현하는 프리젠테이션 동기화 트리(PST)를 제안하고 있으며, 이 프리젠테이션 동기화 트리를 이용하여 효과적으로 프리젠테이션을 하는 수행 모델을 제안하고 있다. 이 프리젠테이션 모델은 실행 시간에 사용자의 입력에 따른 프리젠테이션의 동적인 변경에 쉽게 대처하여 수행할 수 있다. 특히, 본 모델에서는 사용자가 프리젠테이션을 위한 시나리오를 작성하기 위하여 특정한 사용자 인터페이스나 스크립트 언어를 익힐 필요 없이 Allen의 시간 모델에 기초하여 시간 관계를 간단하게 작성하기만 하면 되며, 실제 프리젠테이션을 위한 나머지 작업들로서 시간 관계의 일관성 검증 단계, PST 생성 단계, PST 수행 단계 등을 거쳐 자동적으로 프리젠테이션 되도록 하였다.

A Multimedia Presentation Model Based On PST

Hae-Gill Choi[†] · Sook-Young Choi^{††} · Kwan-Jong Yoo^{†††}

ABSTRACT

To facilitate presentations of various media and transfer their implicit meaning efficiently, needed are effective development techniques for representing and manipulating temporal and spatial informations between multimedia data. Especially, these temporal and spatial informations are so deeply related with each other that it is required to have the integrated method for their representation. This paper suggests the Presentation Synchronization Tree(PST) which represents their temporal and spatial informations and the efficient presentation model as traversing this PST. In this model, without knowing the specific user's interface or any script languages, the user can write the presentation scenario only based on Allen's temporal specification. This model has a characteristic of allowing user's inputs that may dynamically modify the presentation sequence at presentation time. As through the remaining steps of consistency checking for their temporal relations, PST generation, and PST traversing step, this model provides users with simple and effective presentation of multimedia data by having them be presented automatically.

* 본 논문은 1998년도 우석대학교 학술연구 조성비에 의하여 부분적으로 연구되었음.

† 정 회 원 : 한남대학교 전자계산교육원 전임교수

†† 정 회 원 : 우석대학교 컴퓨터과학과 교수

††† 정 회 원 : 충남대학교 컴퓨터과학과 교수

논문접수 : 1998년 7월 6일, 심사완료 : 1998년 9월 29일

1. 서 론

여러 가지 다양한 미디어를 통합 처리하는 멀티미디어 시스템의 발전은 컴퓨터 대중화에 의한 교육, 국방, 정보 처리 관련 등 여러 분야에 적용되기 시작하였다. 이에 따라 멀티미디어 정보 시스템에 대한 요구가 사회 전반 분야에서 확대되어 가고 있고 여러 관련 분야에서 멀티미디어를 사용하여 프리젠테이션(presentation)하는 시스템들이 연구되고 개발되었다[1,2,3,4,14,18,19].

멀티미디어 프리젠테이션은 실시간에 생성되거나 저장 장치에 저장된 멀티미디어 데이터를 처리하여 사람이 인식할 수 있는 형태로 보여주는 것을 의미한다. 멀티미디어 데이터에 함축된 의미를 사람에게 전달하기 위해서는 미디어 데이터간의 시간(temporal) 및 공간(spatial)적인 정보를 효율적으로 표현하고 처리할 수 있는 기법이 필요하다[5]. 시간 정보는 여러 미디어간의 시간 관계 또는 시간 동기화(synchronization) 정보를 의미하며 이것을 모델링하기 위한 여러 기법들이 연구되어 왔다[6]. 그중 가장 대표적인 방법으로서 Allen[7]이 제안한 모델이 있으며, 여러 시간 동기화 명시 기법에서 이를 기초하고 있다. 공간 정보는 멀티미디어 데이터간의 공간 관계 또는 디스플레이 공간 상에서의 레이아웃(layout) 구조에 대한 정보를 의미하며 대표적으로 ODA에서 제안한 표준안[8]이 있다. 그러나, 지금까지의 멀티미디어 동기화에 관련된 연구는 기초적인 명시방법에서부터 저작 시스템에 이르기까지 시간적인 동기화만을 주로 다루어 왔으며, 공간적 동기화에 관해서는 많은 연구가 진행되지 않은 상태이다.

멀티미디어 프리젠테이션을 살펴보면 객체들간에는 시간적 동기화 뿐만 아니라 공간적인 동기화도 이루어져야 한다. 특히 이들 시간 및 공간적인 동기화는 상호 긴밀히 연관되어 있기 때문에 종합적으로 명시하고 운행되어야 한다[16,17].

따라서, 본 논문은 멀티미디어 데이터를 프리젠테이션 하는데 필요한 미디어들간의 시간 정보와 공간 정보를 효과적으로 표현하는 프리젠테이션 동기화 트리 PST(presentation synchronization tree)를 제안하고 있다. 또한 프리젠테이션을 하기 위해 PST를 운행(traverse)하고, 실행 시간에 사용자의 입력을 받아들여 동적인 프리젠테이션을 효과적으로 처리하는 프리젠테이션 모델을 제안하고 있다. 본 모델에서는 사용자가 프

리젠테이션을 쉽게 수행할 수 있도록 프리젠테이션을 위한 시나리오를 Allen의 시간 관계 모델에 기초하여 작성하도록 한다. 이 작성된 시나리오의 시간 관계는 본 모델을 위한 시스템의 일관성 유지 검사기에 의해 검증된 후, PST 생성기에 의해 시간 관계 명세로부터 PST가 자동적으로 구성된다. PST 프리젠테이션 엔진에서 이 PST를 운행함으로써 프리젠테이션이 된다. 따라서, 사용자는 멀티미디어 프리젠테이션을 위해 특정한 사용자 인터페이스나 스크립트 언어를 익힐 필요 없이 미디어들 간의 시간 관계를 간단하게 기술해주기만 하면 되며, 나머지 작업들을 시스템에서 자동적으로 수행되기 때문에 보다 쉽고 효과적인 프리젠테이션을 수행할 수 있다.

본 논문의 구성은 다음과 같다. 2장에서 관련 연구로서 멀티미디어 동기화 참조 모델과 기존의 시.공간 동기화 방법들을 고찰해보고, 3장에서 본 연구에서 제안하는 PST에 대해 정의한다. 4장에서 프리젠테이션 모델로서 프리젠테이션을 위한 PST 관리기, 사용자 상호작용을 처리하는 이벤트 처리기와 스케줄러 등을 설명하고, 본 모델을 지원하는 시스템의 구현 및 기능을 5장에서 설명한다. 끝으로, 6장에서 결론 및 향후 과제를 논의한다.

2. 관련 연구

2.1 멀티미디어 동기화 참조 모델

멀티미디어 시스템에서 동기화란 미디어 객체간에 존재하는 시간 관계를 나타낸다. 멀티미디어를 구성하는 각 미디어들이 시간적으로 동기화되어 프리젠테이션 되지 않으면 시나리오 작성자의 의도를 위반하여 원하는 효과를 얻을 수 없다. 이러한 멀티미디어의 동기화를 위해서는 미디어 객체간의 동기화를 효과적으로 표현하는 방법들과 동기화의 실제 실행을 지원하는 실행 시간 메커니즘들을 구성하는 작업이 요구된다[9,10,11,15].

Blakow[6]는 이러한 멀티미디어 동기화에 대해 네 계층의 참조 모델(four-layer reference model)로 구분하고 있다.

첫째, 미디어 계층(media layer)으로 가장 하위(low-level)에 속하고 단일 연속 미디어 스트림을 운영하며 파일과 장치들을 액세스한다. 둘째, 스트림 계층(stream layer)으로 intra stream 동기화를 보장하며 LDU

(Logical Data Unit)처리를 스케줄링한다. 셋째, 객체 계층(object layer)으로 가장 상위 계층으로부터의 작성된 동기화 명세를 입력으로 받아들여 전체 프리젠테이션을 위한 스케줄링을 하며, 프리젠테이션하기 위해 필요한 행위(action)들을 시작한다. 넷째, 표현 계층(specification layer)으로 가장 상위(high-level)에 해당하며, 원하는 프리젠테이션을 위해 미디어간의 동기화를 기술하는 계층이다.

본 연구에서 제안하고 있는 모델은 위에서 언급한 동기화 참조 모델의 네 가지 계층 중에 객체 계층과 표현 계층에 관련되어 있다.

즉, 사용자가 작성한 시간 관계 명세를 입력으로 받아들여 시간 관계 및 공간 관계를 표현하는 프리젠테이션 동기화 트리를 구성하며, 프리젠테이션을 수행하는데 필요한 동기화를 위해 Playing-Firer를 운영하고 있다.

2.2 시간 동기화 표현 방법

멀티미디어 시간적 동기화 표현 방법들은 다음과 같이 분류한다[5.6.15].

2.2.1 시간 축 모델

시간 축(time line)모델은 모든 미디어의 시간 관계를 시간 축 상에 설정하는 것이다. 이 방식은 절대적인 시간을 시간 축 상에 표시하기 때문에 어느 한 미디어의 시간 정보에 변화가 생기면 그 미디어와 관계된 모든 미디어의 시간 정보에도 영향을 미치게 된다.

2.2.2 구간 기반 모델

시간 간격(time interval)을 기본 단위로 표현하는 방법인 구간 기반(interval-based)모델은 각 미디어 객체마다 프리젠테이션 지속 시간(duration)을 가지고 있어서 이들 구간간의 시간 관계를 표현하게된다. Allen은 시간 간격을 기준으로 하여 서로 다른 시간 간격 사이에 존재할 수 있는 13가지의 시간 관계를 정의하였다[7].

2.2.3 제어 흐름 기반 모델

제어 흐름에 기초한(control flow-based) 표현 모델로써 프리젠테이션이 미리 정해진 시점들에서 동기화되는 방법이다. 이에 관련된 것들로서 계층적(hierarchical) 표현 모델과 Timed Petri Net 모델이 있다. 계

층적 모델은 시간 관계를 쉽게 표현할 수 있다는 장점이 있는 반면, 모든 동기화 조건을 표현할 수 없다는 단점이 존재한다. Timed Petri Net 모델은 시간 관계를 자연스럽게 표현할 수 있지만, 표현 방법이 복잡해질 수 있다는 단점이 존재한다.

2.3 공간 동기화 표현 방법

공간적 동기화란 어떤 이미지와 텍스트를 서로 중첩한다거나 미디어를 확대시키는 것과 같이 미디어를 시간의 흐름에 따라 디스플레이 공간에 서로 결합시키는 것이다. 그러나 이러한 공간적 동기화에 관한 연구들은 활발히 이루어지지 않는 상황이며, 몇 가지 관련 연구들을 살펴보면 다음과 같다.

먼저, 화면상의 좌표로 공간적인 정보를 명시한다. 좌표를 지정하는 방법으로 절대 좌표와 상대 좌표의 두 가지 방식이 존재하며, 절대 좌표는 원점을 기준으로 지정된 좌표이며, 상대 좌표는 현재 위치로부터의 거리를 나타낸 좌표이다. 뿐만 아니라 회전 및 확대 등을 지원하고 있다[12].

상대좌표나 절대 좌표 외에 방향과 거리를 사용하기도 하며, 이는 현재의 위치로부터 아래, 위, 왼쪽, 오른쪽으로 얼마나 떨어졌는지를 표시하는 방법[10]이다.

Fuhrs[8]는 상대적인 공간 관계로써 contain, overlap, connect 등을 정의하고 있다. 본 연구에서는 절대 좌표와 미디어들간의 상대적인 공간 관계를 지원하고 있다.

3. 프리젠테이션 동기화 트리

(Presentation Synchronization Tree: PST)

본 장에서는 본 연구에서 제안하고 있는 프리젠테이션 동기화 트리 모델을 정의하고, 동기화를 제어하는 Playing-Firer의 기능과 수행 규칙을 기술한다.

3.1 모델 정의

프리젠테이션 동기화 트리 PST는 $\{N, L, PF\}$ 로 정의된다.

N : PST 노드들의 집합

$N \subseteq \{PO \cup CO \cup DO\}$

PO : 원형 미디어 객체(primitive media object)들의 집합

CO : 복합 미디어 객체(composite media object)들의 집합

DO : 지연 객체(delay object)들의 집합

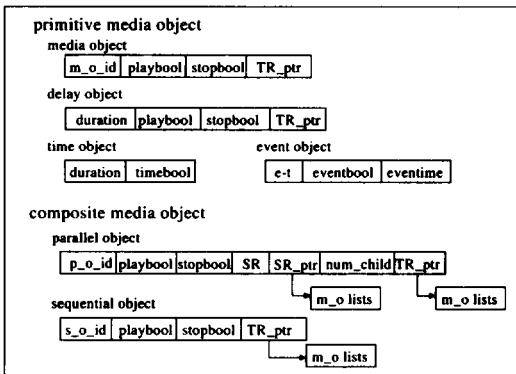
L : 노드 연결 링크의 집합

$L \subseteq \{PO \times CO\} \cup \{CO \times PO\} \cup \{CO \times CO\} \cup \{PO \times PO\} \cup \{PO \times DO\} \cup \{DO \times PO\} \cup \{CO \times DO\} \cup \{DO \times CO\}$

PF : Playing-Firer

PST를 구성하는 노드 N은 원형 미디어 객체 PO와 복합 미디어 객체 CO가 된다. L은 노드들을 연결하는 링크이며, PF는 Playing-Firer로써 프리젠테이션 수행 시 동기화를 맞추면서 노드들을 운행하게 된다.

PST에서 정의하고 있는 객체들의 종류와 그 객체들을 구성하는 필드의 구성은 (그림 1)과 같으며 내용은 다음과 같다.



(그림 1) 미디어 객체들의 구성 형태
(Fig. 1) Components of media objects

3.1.1 원형 미디어 객체(Primitive Media Object)

원형 미디어 객체는 프리젠테이션 될 각 미디어들을 나타내며, 프리젠테이션을 위한 시간 및 공간 정보를 가지고 있으며 다음과 같은 필드로 구성된다. m_o_id는 프리젠테이션 될 미디어의 이름을 나타내며, TR_ptr은 다른 객체에 대한 포인터로서 포인터형 필드이다. 객체의 프리젠테이션 상태는 playbool과 stopbool 필드로서 나타내는데 프리젠테이션 이전에는 두 필드 모두 false로, 프리젠테이션 중이면 playbool은 true, stopbool은 false로, 프리젠테이션 완료 후면 모두 true, 사용자 입력 처리로 일시중지 상태이면 playbool은 false로, stopbool은 true로 세트된다.

3.1.2 지연 객체(Delay Object)

지연객체는 프리젠테이션 동기화를 위해 필요한 객체로서 다음과 같은 필드로 구성된다.

duration은 지연 객체의 프리젠테이션 시간을 나타내며 실수형 필드이다. TR_ptr 필드 및 지연 객체의 프리젠테이션 상태를 나타내는 boolean 필드는 위에서 정의된 것과 동일하다.

3.1.3 시간 객체(Time object)

임의의 객체가 주어진 시간동안 프리젠테이션 되려면 시스템 타이머가 구동되어야 한다. 시간객체는 프리젠테이션 될 객체의 지속시간을 전달받아 외부장치인 시스템 타이머와의 링크를 수행하기 위한 객체로서 다음과 같은 필드로 구성된다. duration은 프리젠테이션 될 객체의 프리젠테이션 시간을 나타내며, 타이머 작동이 시작되면 timebool은 true로, 타이머가 멈추게 되면 false로 세트된다.

3.1.4 사건 객체(Event Object)

사건객체는 프리젠테이션 상태를 변경하는 사용자 입력을 처리하기 위한 목적으로 사용되며 다음과 같은 필드로 구성된다. e_t는 사용자 입력 형태로 임시정지(FZ), 재수행(RS), 전진방향 스킵(F_SK), 후진방향 스킵(B_SK), 프리젠테이션 속도 조절(PS) 등이 있다. 포인터는 사용자 입력을 처리하기 위해 인터럽트 되어야 할 객체에 대한 포인터로서 포인터형 필드이다. eventime은 사용자 입력을 수행하기 위해 소요되는 인터럽트 시간을 나타내며 사용자 상호작용(user interaction)이 발생하면 사건객체의 eventime이 true로 세트되고 사건처리 루틴이 활성화된다.

3.1.5 복합 미디어 객체(Composite Media Object)

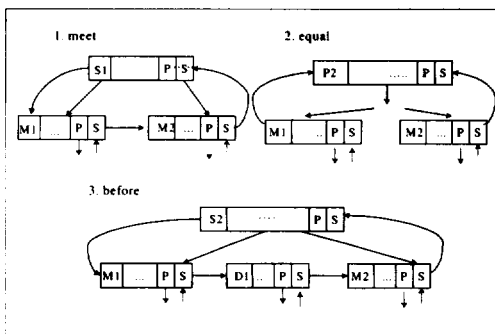
복합 미디어 객체는 미디어 객체간의 시간관계를 표현하기 위한 객체로서 병렬객체(Parallel Object)와 순차객체(Sequential Object)로 분류된다. 순차객체의 경우, 순차 객체에 연결된 하위 미디어 객체들은 링크를 따라 순차적으로 수행되며, 병렬 객체는 각 미디어 객체들이 동시에 프리젠테이션을 수행하게 된다. p_o_id와 s_o_id는 각 복합 객체의 이름을 나타내며 이들 복합 객체의 프리젠테이션 상태 필드인 playbool과 stopbool은 위에서 정의된 바와 같다. TR_ptr은 시간 관계에 있는 미디어 객체들이나 복합

객체들의 리스트들을 가리킨다. SR은 미디어간의 공간 관계를 나타내며, 공간 관계에는 M(meet), D(disconnect), C(contain), O(overlap) 등이 정의되었다. SR_ptr는 공간 관계에 있는 미디어 객체들의 리스트를 가리킨다.

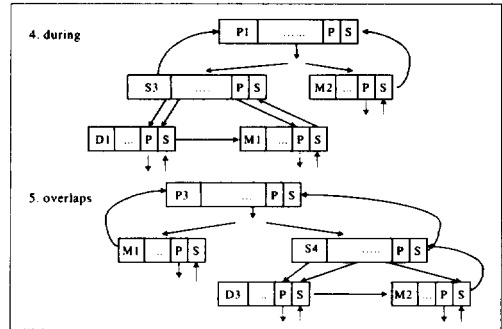
순차객체는 마지막으로 연결된 미디어 객체가 프리젠테이션을 마칠 때 수행을 끝마치게 된다. 병렬객체는 동시에 수행되는 미디어 객체들이 다 마쳐져야만 수행이 끝나기 때문에 이를 위해서 두 가지 정수형 필드를 포함하고 있다. num_child는 병렬객체를 구성하고 있는 미디어 객체의 수이고 counter는 수행이 끝난 자식 객체의 수를 나타낸다. counter필드 값이 num_child필드 값과 같으면 병렬 객체의 수행을 끝내게 된다.

3.2 PST를 이용한 시간 관계 명세

임의의 두 미디어 객체간에는 Allen이 제안한 13가지 시간관계가 존재한다[7]. 이들 관계 중에서 equal을 제외한 다른 모든 시간관계는 역관계를 가진다. 모든 역관계는 정상적인 시간관계로 변환될 수 있기 때문에 기본적인 시간관계는 before, equal, meets, overlaps, during, starts, finishes라고 말할 수 있다. 이들 기본적인 7가지 시간관계를 PST로 나타내면 (그림 2a, 2b)와 같이 5가지 형태로 축약될 수 있다. 그림에서 알 수 있듯이 before와 meet는 순차적으로 수행되어야 하는 미디어의 시간 관계를 나타내므로 PST에서 순차객체를 사용하여 표현할 수 있으며, 나머지 시간 관계는 병렬 적으로 수행되는 미디어 시간 관계를 나타내므로 병렬 객체로 표현할 수 있다.



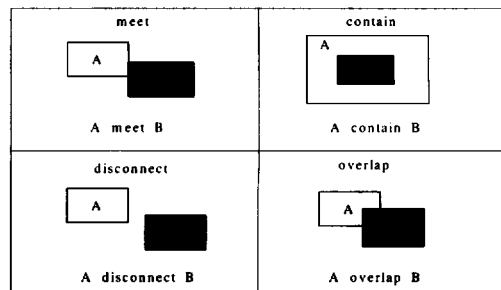
(그림 2a) ALLEN의 시간 관계에 대한 PST
(Fig. 2a) PST for the ALLEN's temporal relationship



(그림 2b) ALLEN의 시간 관계에 대한 PST
(Fig. 2b) PST for ALLEN's temporal relationship

3.3 PST에서의 공간 관계 명세

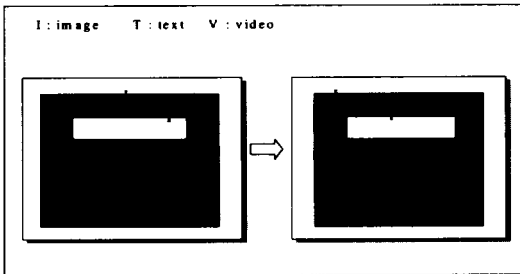
PST 모델에서는 (그림 3)과 같은 4가지의 공간 관계를 정의하고 있다. 이러한 미디어들간의 공간 관계 중 contain과 overlap은 미디어들이 동시에 수행되면서 한 화면에 보여질 때 중요한 의미를 갖는다. 즉, 어느 시간에 병렬로 미디어들이 디스플레이 될 때, 어떠한 미디어가 배경으로 보여지는지 혹은 앞부분에 보여지는지를 결정해야 되기 때문이다. (그림 3)에서 볼 수 있는 바와 같이, contain과 overlap의 경우 A가 바탕 화면이 되고 B가 앞부분에 보여진다.



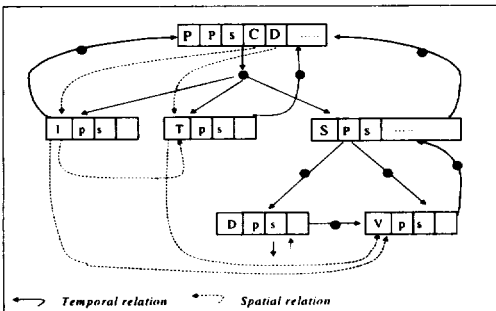
(그림 3) 공간 관계
(Fig. 3) The spatial relationships

PST에서는 복합 미디어 객체인 병렬 객체가 미디어들간에 존재하는 공간 관계 정보 및 그와 관련된 미디어들의 리스트들을 유지한다. 따라서, 프리젠테이션 될 때 이러한 공간 정보를 참조함으로써 미디어들간에 공간 동기화를 얻게 된다. 한 예로서 (그림 5)는 (그림 4)에서 보여지는 미디어들간의 시간 및 공간 정보를

PST로 표현한 것이다. 그림에서 볼 수 있는 바와 같이 병렬 객체 P에서 미디어들간의 공간 관계 C(contains), D(disconnect)를 유지하고 있으며, 점선은 그 공간 관계에 있는 미디어들을 포인트 한다. I(image)와 T(text)는 동시에 보여지며, I(image)는 배경 화면으로 띄워지고, 그 위에 T(text)가 나타나는 것을 의미하게 된다.



(그림 4) 미디어들의 시간 및 공간 정보
(Fig. 4) The information of the temporal and spatial relations between media



(그림 5) (그림 4)에 대한 PST
(Fig. 5) PST for the (fig.4)

3.4 Playing-Firer를 이용한 동기화

프리젠테이션시 객체들 간의 동기화를 위해 PST는 Playing-Firer를 운영한다. Playing-Firer는 PST의 각 노드들을 방문하면서 각 노드의 playbool를 세트 시킴으로써 미디어를 플레이시킨다. 순차객체에 연결된 미디어 객체들을 플레이(play) 하는 경우에는 한 미디어 객체가 플레이를 마친 후 stopbool이 세트(set) 될 때 다음 미디어 객체를 방문하게 된다. 이들 Playing-firer는 객체들 간의 동기화를 제어하는데 아주 중요한 역할을 수행하게 된다. (그림 6)는 각 객체에 대한 playing-Firer의 수행 규칙(rules)들을 기술하고 있다.

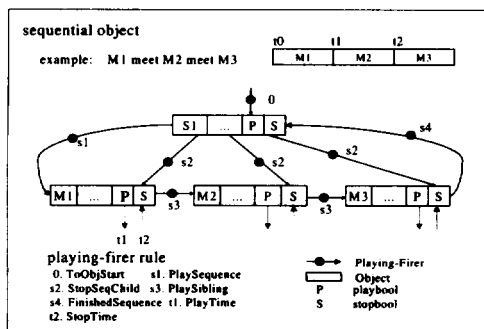
- 0. *ToObjStart()*
set playbool of start object to T
- to sequential object
- s1. *PlaySequence()*
if playbool(sequential) = T then
set playbool(1st media object) to T
endif
- s2. *StopSeqChild()*
if playbool(sequential) = F then
set stopbool(all media objects) to T
endif
- s3. *PlaySibling()*
if stopbool(media object) = T and not end of child then
set playbool(next media object) to T
endif
- s4. *FinishedSequence()*
if stopbool(last media object) = T then
set stopbool(sequential) to T
endif
- to parallel object
- p1. *PlayParallel()*
if playbool(parallel) = T then
set counter to 0
set playbool(all media objects) to T
endif
- p2. *StopParChild()*
if playbool(parallel) = F then
set stopbool(all media objects) to T
endif
- p3. *FinishedParallel()*
if stopbool(media object) = T then
increment counter by 1
if counter = numchild then
set stopbool(parallel) to T
endif
endif
- to time object
- t1. *PlayTime()*
if playbool(media) or playbool(delay object) = T then
start system clock
set timebool to T
endif
- t2. *StopTime()*
if duration time is elapsed then
stop system clock
set timebool to F
if timebool = F then
set stopbool(media object) or stopbool(delay) to T
endif
endif
- to delay object

d1. *PlayDelay()*

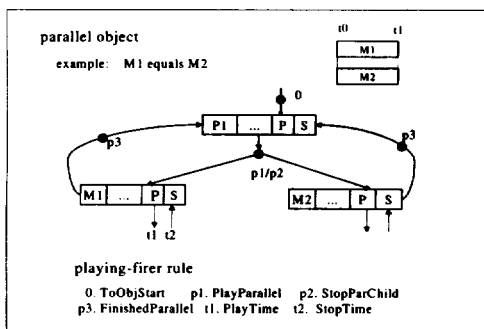
set playbool(delay object) to T
 ask the system clock counting second
 set stopbool(delay object) to T when seconds
 are passed

(그림 6) Playing-Firer의 수행 규칙
 (Fig. 6) The rules for the Playing-Firer

(그림 7a, 7b)는 복합 미디어 객체의 PST에서 Playing-Firer를 운행하면서 프리젠테이션하는 것을 보여주고 있다. 먼저 순차 객체로써 미디어 객체 M1, M2, M3가 순차적으로 프리젠테이션되는 것을 보여주고 있으며, 다음 병렬 객체로서 미디어 M1, M2가 동시에 프리젠테이션을 시작하여 동시에 마치는 것을 보여주고 있다. 여기서, 아크 번호는 프리젠테이션시 동기화에 관련된 Playing-Firer의 수행 규칙들로서 (그림 6)에서 기술되었다.



(그림 7a) 복합 객체에 대한 PST
 (Fig. 7a) The PST for the composite object



(그림 7b) 복합 객체에 대한 PST
 (Fig. 7b) The PST for the composite object

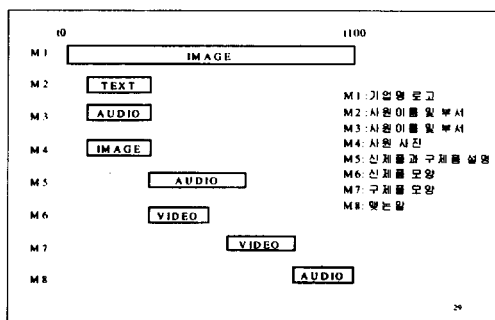
3.5 예제 시나리오

본 절에서는 하나의 예제 시나리오에 대해 위에서 설명한 PST를 보여준다.

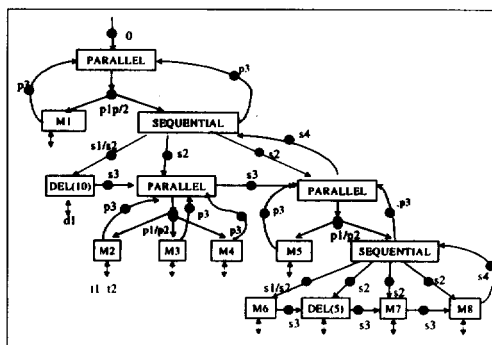
예제 시나리오는 다음과 같다.

어느 기업체의 한 사원이 자신이 개발한 새로운 제품에 대해 프리젠테이션하는 광경이다. 프리젠테이션이 시작되면 기업을 상징하는 마크와 기업 명(M1)이 배경 화면으로 등장한다. 그리고 나서 자신의 이름 및 부서 등을 소개하는 텍스트(M2)가 보여지고, 동일한 내용을 소개하는 음성(M3)이 들려진다. 동시에 화면의 한 편에는 자신의 사진(M4)이 보여진다. 자신의 소개가 끝나면 신제품과 구제품에 대한 설명(M5)이 이어지고, 이 때 화면에는 각 제품들(M6, M7)이 차례대로 보여진다. 제품 설명이 끝난 후에는 결론을 이야기하는 음성(M8)이 들리고 그 음성이 끝남과 함께 프리젠테이션을 마친다.

(그림 8)은 이 시나리오에 대한 미디어들 간의 시간 관계를 나타내고 (그림 9)는 이에 대한 PST를 표현한다.



(그림 8) 예제 시나리오에 대한 시간 관계
 (Fig. 8) The time relations for the example scenario



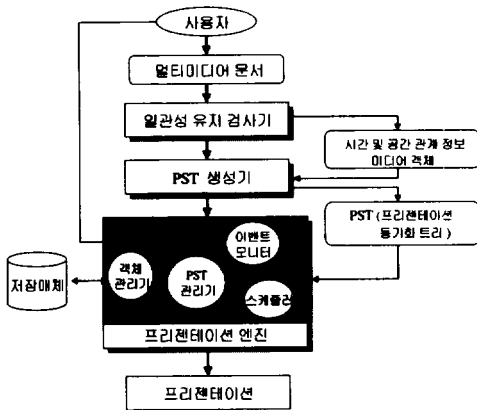
(그림 9) 예제 시나리오에 대한 PST
 (Fig. 9) The PST for the example scenario

4. 프리젠테이션 모델

4.1 모델의 구성

본 연구에서 제안한 프리젠테이션 모델의 구성은 다음과 같다.

(그림 10)에서 볼 수 있는 바와 같이 시스템의 구성 모듈들은 일관성 유지 검사기, PST 생성기, 프리젠테이션 엔진으로 구분된다. 멀티미디어 문서 부분은 사용자 인터페이스를 통하여 사용자가 프리젠테이션을 위한 시나리오를 저장하는 부분으로 미디어 객체간의 시간 관계를 표현한다. 이렇게 하여 입력된 멀티미디어 문서는 일관성 유지 검사기를 거쳐 미디어들의 시간 관계에 대한 일관성을 검사한 다음, 미디어 객체와 미디어 객체간의 시간 관계 및 공간 정보를 추출하게 된다. 이러한 정보들을 이용하여 PST 생성기는 프리젠테이션을 위한 PST를 구성한다. 프리젠테이션 엔진에서는 PST를 운행하면서 프리젠테이션하며, 프리젠테이션시 사용자 상호작용이 발생되는 경우 이를 처리한다.



(그림 10) 모델의 구성도
(Fig. 10) Model architecture

4.2 일관성 유지 검사기

Allen의 시간 관계를 미디어들간의 결합에 적용하는데 있어 문제점으로 지적되는 것이 비일관된 시간관계의 명시가 가능하다는 것이다. 즉, 이미 한 시간관계에 의하여 결합되어 있는 두 시간 간격간에 의미가 일치하지 않는 또 다른 시간 관계가 명시될 수도 있다. 이러한 문제의 해결을 위하여 사용자와 일관성 유지 검

사기 사이에 피드백 구조를 취하여 비 일관된 시간 관계의 입력을 완전히 배제시킨다.

일관성 유지 검사기에서 행해지는 검사과정은 두 가지로 구분될 수 있는데 첫째, 입력된 시간 관계 자체의 정확성 조사로 지속 시간과 지연 시간 등을 고려하여 조사하게 된다. 순차적인 시간관계의 경우에는 미디어 객체의 프리젠테이션 지속 시간과는 상관없이 지연시간만을 조사하면 된다. 그러나, 병렬적인 시간관계의 경우에는 각 시간관계가 프리젠테이션 지속시간과 지연시간에 따라 그 의미를 상실할 수도 있으므로 이들 값을 고려한 조사가 필요하다.

둘째, 중복된 시간관계의 일관성 조사과정으로 이미 사용자의 입력이나 파생시간 관계 생성기에 의해 직접적 혹은 간접적으로 시간관계가 성립하고 있는 두 미디어 객체 사이에 또 다른 시간 관계가 입력되는지의 여부를 조사한다. 만약 이 전의 어떤 시간관계에 의해 이미 두 미디어 객체간에 간접적인 시간관계가 성립하고 있다면 현재 입력된 두 미디어간의 시간관계는 기존의 간접시간 관계와 일치해야 한다.

(그림 11)은 일관성 유지 검사기에서 시간 관계에 따른 검사 규칙을 보여준다.

Consistency_Check_Rule

```

/* let ptime(m) be presentation duration for media
   object m
   let dtime(r) be a delay time for the temporal
   relation r
   let m1, m2 be the first and second media object,
   respectively */

```

```

begin
case meet
dtime(r) == 0
case before
dtime(r) > 0
case overlap
(ptime(m1) > dtime(r)) AND
((ptime(m1) - dtime(r)) < ptime(m2))
case during or start or finish
(ptime(m1) < ptime(m2)) AND
(dtime(r) < (ptime(m2) - ptime(m1)))
case equal
(ptime(m1) == ptime(m2))
end

```

(그림 11) 일관성 검사를 위한 규칙
(Fig. 11) A rule for the consistency check

4.3 PST 생성기

PST 생성기는 위의 일관성 유지 검사기를 거쳐 생성된 미디어 객체들과 미디어 객체들의 시간 및 공간 정보를 입력받아 프리젠테이션을 위한 PST를 생성한다. 즉, 입력된 시간 및 공간 정보와 미디어 객체들의 프리젠테이션 지속 시간을 이용하여 미디어 객체들의 시간 순서에 따라 객체들 간의 링크 및 외부 객체와의 링크를 구성함으로써 PST를 생성하게 된다. (그림 12)는 이에 대한 PST 생성 알고리즘이다.

Algorithm for PST generation

```
/*let m_obj be a primitive media object, p_obj be a parallel object,
s_obj be a sequential object, d_obj be a delay object:
let tr(ij) be a temporal relation between media object i and j:
let tr(ij)_p be a previous time relation: */
```

Input: Media_Object_Table
Output: PST

Procedure Get_token()

```
begin
  read the next token
  call Make_presentation_tree()
end
```

Procedure Make_presentation_tree()

```
begin
  while(token is not null)
    if the token is m_obj then
      put token to the buffer
      call Get_token()
    else if tr(ij) = tr(ij)_p then
      get the token from the buffer
      connect the token to the last created composite object
      call Get_token()
    else if tr(ij) is during or overlap or end then
      call Create_p_obj_child()
      create s_obj and connect it to p_obj as a child node
      connect d_obj to s_obj as a child node
      call Get_token()
    else if tr(ij) is start or equal then
      call Create_p_obj_child()
      call Get_token()
    else if tr(ij) is after then
      call Create_s_obj_child()
      connect the d_obj to s_obj as a child node
      call Get_token()
    else
      call Create_s_obj_child()
      call Get_token()
    endif
  endwhile
end
```

Procedure Create_p_obj_child() /* process which creates parallel object */

```
input: object
output: sub PST
begin
  create p_obj as a parent node
  connect p_obj to the previous composite object
```

```
get the token from the buffer
connect the token to p_obj as a child node
end
```

Procedure Create_s_obj_child() /* process which creates sequential object */

```
input: object
output: sub PST
begin
  create s_obj as a parent node
  connect s_obj to the previous composite object
  get the token from the buffer
  connect the token to s_obj as a child node
end
```

(그림 12) PST 생성 알고리즘

(Fig. 12) An algorithm for PST generation

4.4 프리젠테이션 엔진

프리젠테이션 엔진은 PST 관리기, 객체 관리기, 스케줄러, 이벤트 처리기로 구성되어 프리젠테이션과 사용자 상호작용을 효과적으로 수행한다. PST 관리기는 프리젠테이션을 수행하기 위해 PST의 미디어 객체들의 시간 및 공간 정보를 참조하면서 PST를 운행한다. 객체 관리기는 각 미디어 객체의 프리젠테이션에 관련된 정보들을 관리하며, 이벤트 처리기는 사용자 상호작용을 감지하여 그 해당 사건에 대한 처리를 수행하기 위해 객체의 실행상태 및 객체 수행시간 등을 관리한다.

4.4.1 PST 관리기

PST 관리기는 프리젠테이션을 위한 미디어 객체들의 시간 및 공간 정보들을 유지하고 있는 PST를 운행한다. 이 PST의 각 노드들을 Playing-Firer가 활성화 시킴으로써 동기화를 제어하면서 프리젠테이션한다. (그림 13)의 PST 수행 알고리즘에서 이탤릭체로 기술된 부분이 Playing-Firer의 동기화를 위한 수행 규칙을 호출하고 있는 부분이다.

Algorithm for processing the presentation of PST

Input: Presentation Synchronization Tree

Procedure Present_PST()

```
begin
  ToObjStart()
  call Proc_composite
end
```

Procedure Proc_composite()

```
begin
  if parallel object then call Processpar()
  else call Processseq()
```

```

endif
end

Procedure Processpar() /* process for the parallel
object */
begin
  PlayParallel()
  get child object
  while (not end of child)
    if primitive object then
      if delay object then PlayDelay()
      else PlayTime()
    endif
    else call Proc_composite()
  endif
  FinishedParallel()
  get next child
endwhile
end

Procedure Processseq() /* process for the sequential
object */
begin
  PlaySequence()
  get child
  while (not end of child)
    if primitive object then
      if delay object then PlayDelay()
      else PlayTime()
    endif
    else call Proc_composite()
  endif
  PlaySibling()
  endif
  FinishedSequence()
endwhile
end

```

(그림 13) PST 수행 알고리즘
(Fig. 13) An algorithm for traversing of the PST

4.4.2 객체 관리기

프리젠테이션을 위한 미디어 객체들과 그들에 관한 정보를 관리한다. 이들 정보들은 미디어 객체 타입, 사용자가 명시한 프리젠테이션 지속 시간과 초기 프리젠테이션 상태 등의 기본 정보들과 파일 시스템에 존재하는 미디어 객체에 대한 경로 정보 및 크기, 그 밖의 미디어 속성 등의 미디어 특성에 관한 것들이다. 객체 관리기는 디스플레이 위치 등의 정적 정보를 유지하며 PST 트리를 운행함에 따라 변화되는 미디어 객체의 프리젠테이션 상태와, 사용자 상호작용에 의한 동적 변화에 따른 가장 최근의 정보를 유지한다. 각 미디어 객체의 상태는 프리젠테이션 이전 상태인 Ready, 실행 상태인 Running, 객체의 수행 완료 상태인 Complete,

그리고 사용자 상호 작용에 따른 일시 정지를 위한 Pause 등의 4가지 상태로 정의된다.

4.4.3 스케줄러

입력된 시간 관계 및 미디어 객체의 프리젠테이션 지속 시간을 이용하여 두 미디어 간의 프리젠테이션 시작 시간을 결정한다. 구해진 시작 시간은 객체 관리기에 의해 미디어 객체 테이블에 저장되어 관리된다. 시간 관계가 의미하는 바에 따라 각 미디어의 시작 시간을 결정하는 기본 방법은 (그림 14)와 같고, 변수들은 일관성 유지 검사기에서 정의된 바와 같다.

Algorithm for scheduling of media presentation

Input: presentation duration, temporal relations

Output: presentation start time for the media object

/* let stime(m) be presentation start time for media object m */

```

Procedure Scheduling_Proc()
begin
  check temporal relation
  case before or meet
    stime(m2) = stime(m1) + ptime(m1) + dtime(r)
  case overlap
    stime(m2) = stime(m1) + dtime(r)
  case during
    stime(m2) = stime(m1) - dtime(r)
  case start or equal
    stime(m2) = stime(m1)
  case finish
    stime(m2) = stime(m1) - (ptime(m2) - ptime(m1))
end

```

(그림 14) 프리젠테이션 시작 시간 결정 알고리즘
(Fig. 14) An algorithm for determination of start time for the presentation

두 객체 모두의 시작 시간이 결정되어 있지 않은 경우에는 먼저 프리젠테이션 되어야 할 미디어 객체의 시작점을 0으로 하고, 그 둘 간의 시간 관계에 따라 나머지 한 미디어 객체의 시작 시간을 결정한다. 이미 존재하던 미디어 객체와 시간 관계를 갖게 되면 그 미디어 객체의 프리젠테이션 시작 시간은 이미 결정되어 있으므로 그 것을 이용하여 새로운 미디어 객체의 프리젠테이션 시작 시간을 결정할 수 있다. 두 객체 모두 이미 결정된 시작 시간을 가지고 있는 경우는 문서를 구성되 서로 독립적으로 존재하던 즉, 어떠한 시간 관계에 의해서도 연결되지 않고 있던 두 컴포넌트(component)를 연결하는 시간 관계가 입력된 것이다.

그러므로 두 번째 미디어 객체의 프리젠테이션 시작 시간을 첫 번째 미디어 객체의 시작 시간을 기준으로 삼아 재조정하고 두 번째 미디어 객체가 속한 컴포넌트의 모든 미디어 객체들의 시작 시간도 재조정하여 준다.

4.4.4 이벤트 처리기

이벤트 처리기는 사용자 상호 작용에 의해 프리젠테이션의 순서를 동적으로 변경하는 경우에 각 미디어 객체의 상태를 관리하고, 이를 효과적으로 처리해주는 모듈이다.

PST 모델에서 지원하고 있는 사용자 상호작용에는 일시정지(freeze) 및 재수행(resume), 전진방향 스킵(forward skip), 후진방향 스킵(backward skip), 프리젠테이션 속도조절(scaling)이 있다. 사용자 상호작용에 의해 사건이 발생되면, 이벤트 처리기는 사건 객체의 상태를 활성화시키고 각 사건에 따른 처리 루틴을 수행한다. (그림 15)는 각 사건에 따른 처리 루틴을 나타낸다.

(1) 일시정지 및 재수행 (freeze and resume)

사용자가 프리젠테이션의 수행을 일시 정지하는 명령을 입력하면 현재 프리젠테이션 중인 모든 미디어의 프리젠테이션을 즉각 정지한다. 이 명령을 수행하기 위해서는 현재 playbool이 true인 모든 미디어 객체를 선택하여 프리젠테이션 수행을 중지한 후 그것들의 stopbool을 true, playbool을 false로 세트하고 잔여시간으로 duration을 고정시킨다. 사용자가 일정 시간이 경과한 후에 다시 재수행 명령을 입력하면 현재 일시정지 상태에 있는 모든 객체의 프리젠테이션 상태를 나타내는 필드인 stopbool을 false, playbool을 true로 세트하여 수행을 다시 재개한다.

(2) 전진방향 스킵(forward skip)

사용자가 현재 프리젠테이션 중인 미디어의 수행을 전진방향 스킵하라는 명령을 입력하면 현재 수행중인 모든 미디어 객체들의 playbool을 false로, stopbool을 true로 세트시키고 스킵시간이 지난 다음에 프리젠테이션 되어야 할 미디어 객체들의 playbool을 true로 세트시키고 각 미디어들의 duration을 남은 시간으로 갱신한다.

(3) 후진방향 스킵(backward skip)

사용자가 현재 프리젠테이션 중인 미디어의 수행을

후진방향 스킵하라는 명령을 입력하면 현재 수행중인 모든 미디어 객체들의 playbool을 false로, stopbool을 true로 세트시키고 스킵시간이 지난 다음에 다시 프리젠테이션 되어야 할 이전 미디어 객체들의 stopbool을 false로 세트시키고 각 미디어들의 duration을 남은 시간으로 갱신한다.

(4) 프리젠테이션 속도 조절(scaling the presentation speed)

사용자가 프리젠테이션 수행 중인 미디어의 프리젠테이션 속도를 늦추거나 빨리하는 프리젠테이션 속도 조절 명령을 입력하면 현재 프리젠테이션 중인 미디어 데이터 뿐 아니라 playbool이 true인 모든 미디어의 속도를 변화시킨다. 또한, sequential object에 의해서 순차적으로 수행될 예정인 미디어의 속도는 변경되지 않지만 동기화를 위해 프리젠테이션 속도의 조절비에 따라 프리젠테이션 시간을 연장 또는 단축시켜야 한다.

이 후에 수행될 다른 미디어들의 정적 프리젠테이션 시간은 변경하지 않더라도 Playing-Firer에 의해 수행되면서 자동적으로 속도 조절에 따른 프리젠테이션 변화가 반영된다.

Algorithm for processing of user's interaction
Input: PST, information of user's interaction

Procedure Event_process()

```
/* let Et be a current time at the point of event occurred:
let event occurred against a media object i:
let object k be any related objects with object i:
let mST(i) be a presentation start time of i:
let mFT(i) be a presentation finish time of i:
let Sk_time be a time after skip:
let V be a speed of presentation:
let ΔV be a scaling factor of the presentation speed:
let Rt(i) be a remaining presentation time for media object i:
let Rt'(i) be a remaining presentation time for media object i as the result of event SS:
let Pts be a presentation time gap as scaling the speed:
let Rt(k), Rt'(k) be a duration and a remaining presentation time of the other related objects, respectively: */
```

begin

```
case event-type : FZ
set stopbool=T, playbool=F for all object which have playbool of T
Rt(i) = duration(i) - Et /* reset duration with remaining time */
endcase
```

case event-type : RS

```

    reset Rt for all object which have playbool of F
    and stopbool of T
    set playbool to T, stopbool to F
endcase

case event-type : F_SK
    set playbool to F, stopbool to T for object i
    if (mST(i) < Sk_time) and (mFT(i) > Sk_time)
    then
        if (mST(i) < Et) then
            Rt(i) := duration(i) - (lmST(i) + Et) + skip
            time )
        endif
        if (mST(i) > Et) then
            Rt(i) = duration(i) - (skip time - lmST(i) - Et)
        endif
    endif
endcase

case event-type : B_SK
    set playbool to F, stopbool to T for object i
    set stopbool of object k to F
    if (mST(i) < Sk_time) and (mFT(i) > Sk_time)
    then
        if (mFT(i) > Et) then
            Rt(i) := skip time + (duration(i) - (|Et -
            mST(i)|))
        endif
        if (mFT(i) < Et) then
            Rt(i) = duration(i) - ((Et - skip time) -
            lmST(i))
        endif
    endif
endcase

case event-type : SS
    reset duration for all object which have playbool of
    T as follows:
    Rt'(i) = Rt(i) * V * ΔV
    Ptq = Rt(i) - Rt'(i)
    Rt'(k) = Rt(k) + Ptq
endcase
end

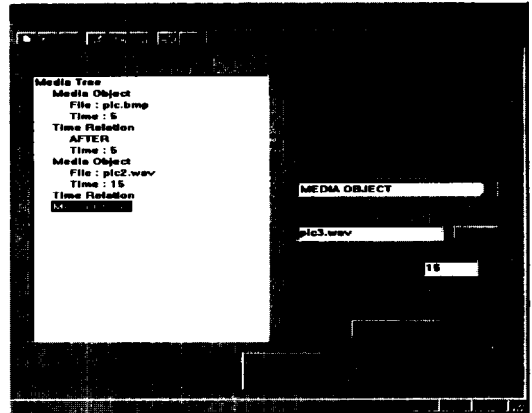
```

(그림 15) 이벤트 처리기의 사용자 상호작용에 대한 수행 알고리즘
 (Fig.15) An algorithm for execution of the event processor

5. 구현 및 실험

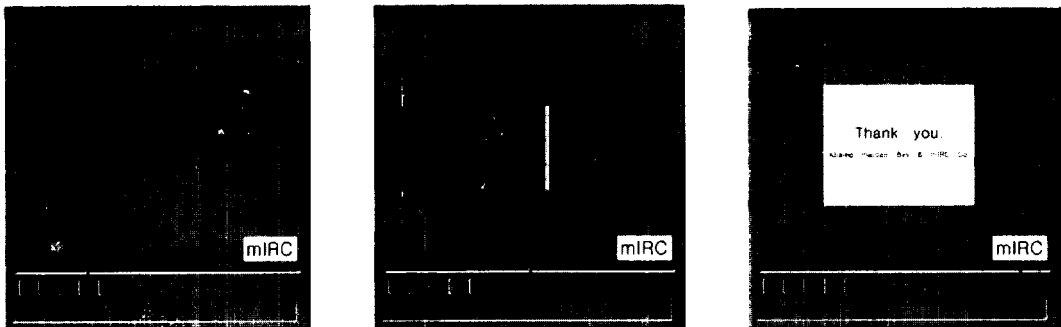
본 모델의 구현 환경은 비주얼 C++ 4.2와 5.0을 바탕으로 MFC 라이브러리를 사용하여 이미지, 텍스트 미디어 파일들을 처리하고, Win32.Api의 MCI 함수를 사용하여 비디오, 오디오 미디어 파일들을 처리한다.

(그림 16)은 사용자가 사용자 인터페이스를 통해 Allen의 시간 관계에 따라 미디어간의 시간 관계를 입력하면 PST 생성기에 의해 생성되는 PST를 보여준다. 생성된 PST는 프리젠테이션 엔진에 의해 자동적으로 프리젠테이션 된다.



(그림 16) 입력을 위한 사용자 인터페이스
 (Fig. 16) User interface for input

다음은 3.5절에서 언급한 예제 시나리오에 대하여 프리젠테이션된 결과를 보여준다.



(그림 17) 예제 시나리오의 수행 과정
 (Fig.17) Procedure of the example scenario

6. 결 론

본 논문에서 제안한 PST는 멀티미디어 문서를 구성하는 미디어들 간의 시간 관계와 공간 관계를 효과적으로 표현할 수 있고, 트리를 운행함으로써 그 정보를 이용하여 손쉽게 프리젠테이션할 수 있다. 또한 사용자 상호작용에 따른 프리젠테이션의 동적인 변경에 쉽게 대처할 수 있다. PST 관리기는 프리젠테이션의 시간적 동기화를 제어하기 위해 playing-firer를 지원하고 있으며, PST의 노드들을 따라 해당 playing-firer에 관한 수행 규칙을 실행함으로써 프리젠테이션이 진행된다. 특히, 프리젠테이션시 어느 시점에서 미디어들이 병렬로 수행되는 경우, PST에 표현된 미디어들간의 공간 관계 정보를 이용하여 공간의 동기화를 효과적으로 제어할 수 있다.

본 모델에서는 사용자가 멀티미디어 프리젠테이션을 위해 미디어간의 시간 관계를 Allen의 모델에 기초하여 간략하게 명시하기만 하면 되며, 실제 프리젠테이션을 수행하기 위한 나머지 작업들은 시스템에서 자동적으로 수행하도록 하고 있다. 따라서, 본 모델은 프리젠테이션을 위한 자동 생성(automatic generation)에 관한 연구에 기초가 될 수 있다. 또한, PST구성에 의한 본 모델은 다양한 멀티미디어 환경에서 여러 응용들을 쉽게 이식할 수 있도록 하며, 멀티미디어 통합 환경의 구축에 기초가 될 수 있을 것이다. 향후 과제으로써 본 모델에 하이퍼링크(hyperlink) 기능을 추가하는 것과, 분산 환경으로의 확장이 요구된다.

참 고 문 헌

- [1] K.Fujikawa and et al., "Multimedia Presentation system Harmony with Temporal and Active Media," Proc. of the 1991 USENIX Conference, pp.75-93, June 1991.
- [2] J. M. Bacon and et al., Storage and Presentation Support for Multimedia Application in Distributed, ATM Network Environment, Technical Report 295, Univ. of Cambridge Computer Lab, 1993.
- [3] M. Bordegoni, "Multimedia in Views," CR Categories H.5.1, CWI, Amsterdam, The Netherlands, 1991.
- [4] J. Bates, Presentation for Distributed Multimedia Application, Ph D Dissertation, Cambridge Univ.
- [5] R. Steinmetz, "Synchronization Properties in Multimedia Systems," IEEE Journal on Selected Areas in Commu, 8(3), pp.401-412, Apr. 1990.
- [6] G. Blakow and R. Steinmetz, "A Media synchronization in Survey," IEEE Journal on Selected Areas in Communication Vol.14, No.1, 1996.
- [7] J. F. Allen, "Maintaining Knowledge about Temporal Intervals," Comm. of the ACM, Vol.26, No. 11, pp.832-843, Nov. 1983.
- [8] B. Fuhr, "Multimedia Systems: An Overview," IEEE Multimedia, Vol.1, No.1, pp.47-59, Spring 1994.
- [9] Thomas D. C. Little and A. Ghafoor, "Spatio-Temporal Composition of Distributed Multimedia Objects for Value-Added Networks," IEEE Computer, 1991.
- [10] B. Prabhakara and S. V. Raghavan, "Synchronization Models for Multimedia Presentation with User Participations," Proc. of the First ACM Int'l Conference on Multimedia, pp.157-163, Aug. 1993.
- [11] A. Duda and et al, "Structured Temporal Composition of Multimedia Data," Intl. Workshop on Multimedia Database, pp.136-142, Aug. 1995.
- [12] M. C. Buchanan and P. T. Zellweger, "Automatic Temporal Layout Mechanisms," Proc. of ACM Multimedia 93, pp.341-350, Jun. 1993.
- [13] T. D. C. Little and A. Ghafoor, "Synchronization and Storage Models for Multimedia Objects," IEEE Journal On Selected Area, Vol.8, No.3, pp. 413-427, Apr. 1990.
- [14] R. Price, "MHEG: An Introduction to the Future International Standard for Hypermedia Object Interchange," ACM Multimedia 93, pp.121-126, Aug. 1993.
- [15] John F. Koegel Buford, Multimedia Systems, Chap. 7, 11, 12, ACM Press, Addison-Wesley New York, 1994.
- [16] H. Shin, Y. Won, W. Yoo and K. Yoo, "Presentation Jockey: Multimedia Authoring System for

Automatic Presentation," IASTED Applied Informatics '97, pp.259-262, Feb. 1997.

[17] H. Choi, S. Choi and K. Yoo, "A Study on Internal Representation Model for Efficient Presentation in Various Multimedia Environment," 13th International Conference on Computers and Their Applications, pp.230-233, Mar. 1998.

[18] K. Fujikawa and et al., "Multimedia Presentation System "Harmony" with Temporal and Active Media," USENIX, pp.75-93, Summer, 1991.

[19] J. Schnepf and et al., "Doing FLIPS: Flexible Interactive Presentation Synchronization," IEEE Journal On Selected Areas In Communications, Vol.14, No.1, pp.114-125, Jan. 1996.



최혜길

e-mail : hgchoi@nspplab.chungnam.ac.kr

1978년 연세대학교 생화학과(이학사)

1980년 연세대학교 생화학과(이학석사)

1982년 북일리노이 주립대학 생화학과(M.S)

1986년 북일리노이 주립대학 전산학과(M.S)

1996년 충남대학교 박사과정 수료

1991년~현재 한남대학교 전자계산교육원 전임교수

관심분야 : 멀티미디어 시스템, 병렬처리, 컴퓨터 모델링



최숙영

e-mail : sychoi@core.woosuk.ac.kr

1988년 전북대학교 전산통계학과(이학사)

1991년 전북대학교 전산학과(이학석사)

1996년 충남대학교 전산학과(이학박사)

1996년~현재 우석대학교 컴퓨터교육과 조교수

관심분야 : 멀티미디어 시스템, 병렬처리, 원격 교육



유관종

e-mail : kjyoo@mssplab.chungnam.ac.kr

1976년 서울대학교 계산통계학과(이학사)

1978년 서울대학교 계산통계학과(이학석사)

1984년 서울대학교 계산통계학과 이학박사

1989년~1990년 캘리포니아 대학(IRVINE) 방문 교수

1979년~현재 충남대학교 컴퓨터과학과 교수

관심분야 : 멀티미디어 응용, 멀티미디어 통신, 프로그래밍 언어, 병렬처리