

멀티미디어 문서의 프리젠테이션 품질 유지를 위한 이벤트 모델링 기법

An Event Modelling Technique for Maintaining the Presentation Quality of a Multimedia Document

이규남, 허기택*, 양현호**, 나인호**

동신대학교 산학협력단 디지털콘텐츠협동연구센터
전임연구원

동신대학교 디지털콘텐츠학과*
군산대학교 전자정보공학부**

Lee Kyu-Nam, Hur Gi-Taek*, Yang Hyun-Ho**,
Ra In-Ho**

Digital Contents Cooperative Research
Center, Dongshin Univ.

Department of Digital Contents, Dongshin
Univ.*

School of Electronic & Information
Engineering, Kunsan National Univ.**

요약

본 논문에서는 멀티미디어 콘텐츠의 프리젠테이션 품질을 유지하기 위해 멀티미디어 콘텐츠를 처리할 때 발생하는 여러 사건들을 이벤트(event)로 정의하고, 각각의 특징에 따라 여러 개의 클래스로 분류하였다. 그리고 이벤트의 처리와 관리에 의해 프리젠테이션 동기화를 처리할 수 있는 모델을 제안하였다. 또한, 쇼(show) 진행에 관련된 실행모듈, 스레드(thread) 등이 이벤트를 이용하여 상호 통신함으로써 프리젠테이션 서비스를 유지하는 방안에 대해 제안하였다. 특히, 제안된 모델에서는 사용자 인터페이스를 통해 사용자가 프리젠테이션 수행에 직접적으로 개입할 수 있을 뿐만 아니라 서비스 과정에서 발생하는 예측 불가능한 상태 변화에 대하여 능동적으로 대처함으로써 멀티미디어 문서의 프리젠테이션 품질을 유지할 수 있도록 하였다.

Abstract

In this paper, for maintaining the presentation quality of multimedia contents, we define the several events that can happen in showing when handle the multimedia contents. Also we classified these events in 4 types based on each characteristic. Furthermore we design the event model which can manage synchronization of presentation by control and managing the events. Also we propose an idea that maintain the quality of presentation service by interaction between each module and threads related in progress of show. Especially, in this research, we focus the active process for the user's participant and the unpredictable state variation of IMPSSs(Interactive Multimedia Presentation Systems).

I. 서론

최근 들어 네트워크 및 컴퓨터 관련 기술의 눈부신

발전에 따라 멀티미디어 응용의 사용 범위는 급속도로 확산되고 있으며, 최근에는 인터넷을 통한 대화형 멀티미디어 서비스 분야가 각광을 받고 있다[3]. 이에

따라, 멀티미디어를 구성하는 모노 미디어의 종류와 포맷도 매우 다양해지고 있다.

과거 멀티미디어 콘텐츠의 제작 및 프리젠테이션 기법에 대한 연구는 주로 텍스트, 그래픽, 사운드, 비디오 등의 4가지 모노 미디어들을 중심으로 시공간적 조합과, 표현에만 중점을 두고 있으나 본 논문에서는 추가적으로 디스플레이 공간에서 미디어 객체가 프리젠테이션될 때의 모양(shape)과 효과(effect)를 멀티미디어 문서에 적용할 수 있도록 하였다.

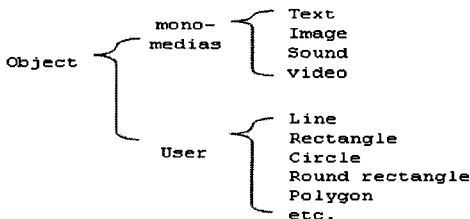
또한, 사용자의 프리젠테이션 개입, 처리 지연, 네트워크의 가변적 대역폭, 자원의 관리 등 다양한 요소를 동시에 종합적으로 고려하여 멀티미디어 프리젠테이션 품질을 유지하기 위한 'event' 모델을 설계하였다.

본 논문의 내용은 다음과 같다. 2장에서 'object', 'effect', 그리고 'actor'의 개념에 대하여 기술한다. 3장과 4장에서는 각각 제안된 'event' 모델링 기법의 특징 및 수행 방법과 간단한 시나리오를 통한 응용 예를 설명한다. 마지막으로 5장에서는 향후 연구방향과 결론에 대해 기술한다.

II. Object, Effect, 그리고 Actor의 개념

Object의 개념 : 멀티미디어 콘텐츠를 구성하는 4가지 유형의 모노 미디어와 디스플레이 공간에 대해 사용자에게 의해 정의되는 모양(user defined shape) 정보를 가진 프리젠테이션 대상을 의미한다.

Object = {(ID), ("network address": "local path"), "file name", "owner", etc}

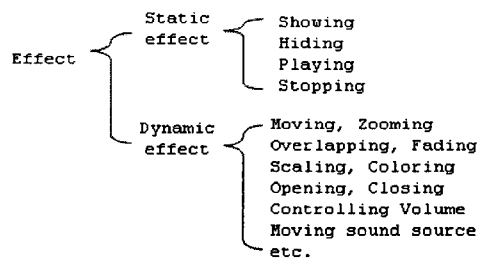


멀티미디어 응용은 시나리오 구성을 통해 각각의 개별적인 미디어에 미리 정의된 시공간적 동기화 정보를 엄격하게 준수하면서 사용자에게 보여 질 때 만족할 만한 서비스를 제공할 수 있다. 일반적으로, 텍스트, 이미지 및 동영상은 디스플레이 공간에 출력되고 오디오는 스피커를 통해 출력되며, 이러한 출력은 정의된 시간에 따라 순차적 또는 병렬적으로 동시에 이루어진다.

이와 같이 각 미디어 객체의 출력에 대한 기본 서비스 이외에도 멀티미디어 프리젠테이션을 수행할 때에는 이미지가 보여지면서 이동하거나 크기가 변화하는 등 각 미디어에 대한 다양한 표현 효과가 요구된다. 또한 확대와 겹치기, 초점이동 등 여러 표현 효과들을 동시에 수행하는 것이 필요한 경우도 있다. 특히 사운드에 대해서는 볼륨 조절, 잔향, 음원의 이동 등 좀 더 역동적인 효과를 필요로 하는 콘텐츠 제작이 요구되기도 한다. 이러한 표현 효과들은 정적(static) 표현과 동적(dynamic) 표현으로 나누어진다.

본 논문에서는 정적 및 동적 표현 효과를 표현할 수 있도록 이것들을 "Effect"로 정의하였다.

Effect의 개념 : 'effect'는 콘텐츠를 구성하는 각 'object'에 대해 요구된 정적 및 동적 표현 효과와 출력을 위한 시공간 정보를 기반으로 사용자에게 내용을 전달하고자 하는 표현 방법을 정의한다.



Effect = {(ID), (pos(x1, y1, x2, y2)), (start time, duration time), (effect content), etc}

여기서 'object'와 'effect'가 결합하면, 'object'가

가지고 있는 정보를 'effect'를 이용하여 표현할 수 있으며, 본 논문에서는 이것을 다시 'actor'로 정의하였다.

Actor의 개념 : 'actor'란 콘텐츠를 구성하고 있는 'object'를 시공간 정보를 지니고 있는 'effect'와 결합하여 실질적으로 사용자에게 보여줄 프리젠테이션 인스턴스(instance)를 의미한다.

$Actor = \{(Object), (Effect\ 1), (Effect\ 2), \dots (Effect\ n)\}$

여러 모노 미디어들을 조합하여 프리젠테이션 가능한 하나의 시나리오를 구성하였을 때에는 각 'object'에 적용할 수 있는 한 가지 이상의 표현 방법을 제공되어야 한다. 이를 위해 본 논문에서는 'actor'를 'object'에 'effect' 기능이 부가된 스레드(thread)로 모델링 하였다.

III. Event 모델링

1. Event의 개념

최근 들어 사용자 개입(user participation) 처리를 비롯하여 컴퓨터 내부에서의 시공간적 프리젠테이션 동기화, 네트워크 전송을 변화에 따른 전송지연 흡수를 위한 네트워크 동기화 등의 수행과 관련된 여러 사건(event)들을 고려하여 멀티미디어 콘텐츠의 서비스 품질을 유지하기 위한 기법에 대한 연구가 활발히 진행되고 있고 있으며, 주로 여러 가지 사건들은 독립적으로 관리하기 보다는 종합적으로 분석 및 처리하여 일정한 수준의 QoS를 제공하는 방법에 대한 연구에 초점을 두고 있다[5].

지금까지 진행된 많은 연구를 통해 'event'에 대한 다양한 정의가 이루어져 왔다[1],[2],[4],[6], [7], [8], [9]. 일부 논문에서는 event'를 단순히 프리젠테이션 과정에서 고려해야 하는 일시적인 사건으로 정의한 경우도 있다[6][7][8]. 즉, 이것은 이벤트를 프리젠테

이션이 진행되는 시간 중 어느 한 지점에서 발생한 일시적인 사건으로 정의하고, 이 사건은 개별적이거나 중속적일 수 있는 것으로 해석하였다. 또 다른 연구에서는 이벤트를 객체에 대한 시간적 조합으로 정의하였으며, 이를 프리젠테이션 객체에 대한 시간지연으로 표현하였다[10]. 그 외에도 멀티미디어 조합이나 프리젠테이션에 대하여 제안한 연구들에서 시공간적 인스턴스에 부합된 행위를 유발하는 것으로 정의하였으며, 이러한 이벤트는 사용자나 프로세스 자신이 발생시키는 것으로 기술하였다[1][2][4].

본 논문에서는 'event'를 IMPSSs(Interactive Multimedia Presentation Systems)의 QoS를 유지하기 위한 특별한 동기화 정보로 정의하였다.

Event의 개념 : IMPSSs의 QoS를 유지하기 위하여 필요한 동기화 정보이다. 이 정보는 사용자의 프리젠테이션 개입과 시스템 반응 과정, 조합된 객체(composed object), 시스템을 구성하고 있는 하부 모듈, 생명(life cycle)을 갖는 'actor' 등이 발생시키거나 이것들에 의해 처리될 수 있다.

2. Event의 종류

프리젠테이션의 안정된 성능을 보장하기 위해, 프리젠테이션 과정에서 고려해야 할 요소들을 4가지로 구분하였다. 이러한 요소는 동기화된 프리젠테이션 진행에 필요한 여러 사건을 발생시킬 수 있다. 각 요소는 'event'로 표현되며, 그것의 처리는 기능에 따라 정해진 수행모듈에서 처리되도록 설계하였다.

User Events : 이것은 IMPSSs와 상호작용하는 사용자 행위에 의해 발생하며, 사용자가 IMPSSs에 명령을 입력하면 시스템이 해당 명령에 대해 적절한 반응을 취함으로써 처리된다. 사용자 입력은 마우스, 키보드, 터치스크린과 같은 장치를 통해 발생된다.

Actor Events : 이것은 'Object Events'와 'Effect Events'로 구분된다. 'Object Event'는 'object'를 대상으로 하는 'event'로서 쇼가 진행 되는 동안에

IMPSs는 'object'를 구성하는 미디어 파일을 모니터 하면서 이와 관련된 'event'를 처리한다. 'Effect Events'는 프리젠테이션 효과를 나타내는 'event'이다. 여러 'object'는 다양한 표현 효과를 가질 수 있으며, 이를 'effect'로 정의하였다. 'actor'는 'effect'가 가지고 있는 시공간 정보 및 표현 방법에 의해 사용자에게 서비스 된다. 이 'event'는 'effect'의 정보를 실행시킨다.

System State Events : 이것을 이용하여 IMPSs는 컴퓨터 하드웨어 요소들의 개별적인 상태에 대한 정보를 공유한다. 연속미디어인 사운드와 비디오 데이터는 대부분 큰 메모리 용량을 차지하며, 특히 시간 지연에 민감한 특징이 있다.

Synchronization Events : 이것은 QoS 유지를 위한 IMPSs의 동적인 제어 'event'이다. 네트워크 전송률 변화를 감지한 'System State Events'는 IMPSs에 의해 다른 부분의 현재 상태를 고려한 동기화 'event'로 변화되어 처리한다. 현재 처리되고 있는 타 'actor'의 'effect'를 적절히 지연(delay)시키거나 건너뛰기(skip)하는 동기화 'event'가 처리될 수 있다.

3. Event 모델링

'event'는 다양한 발생 원인에 따라 그 내용이 달라지며, 또한 'event'를 처리할 대상 또한 그 내용에 따라 달라진다. 또한, 동시 다발적으로 발생한 'event'들은 그 내용에 따라 우선순위를 두고 구분하여 처리할 수 있다

$Event = \{(ID), "content", "source", "destination", "reference", "priority", \dots\}$

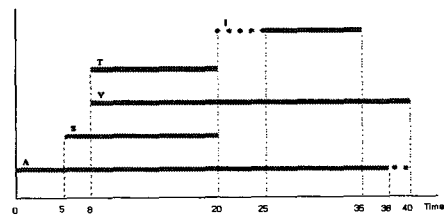
여기서 'content'는 'event'에 관련된 여러 가지 세부 항목을 나타낸다. 이 정보는 'event'가 어떤 종류인지와 목적지(destination)가 어떤 처리를 하여야 하는 것을 나타낸다. 'source'는 'event'를 유발한 근원지를 의미하며, 이 객체는 IMPSs 시스템, 시스템을 구성하는 모듈 혹은 스레드, 'actor' 등 여러 가지일 수 있다. 'destination'는 발생한 'event'를 처

리할 대상이며, 이 또한 IMPSs 시스템, 모듈, 스레드, 'actor' 등이 될 수 있다. 'reference'는 'event'를 처리하기 위한 세부정보를 포함한다. 이 속성은 'event'의 내용, 발생 객체와 처리 객체에 따라 크게 달라질 수 있다.

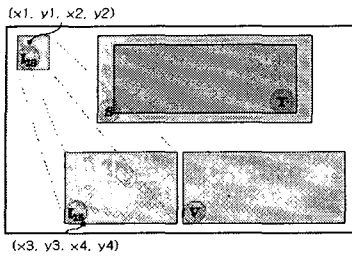
IV. 제안한 모델의 예

1. 프리젠테이션 시나리오

제안된 'event' 기반의 프리젠테이션 처리 모델의 특징과 적용방법을 기술하기 위해 다음과 같은 간단한 멀티미디어 프리젠테이션에 관한 예를 들었다. 그림 1과 2는 사용자에게 의해 정의된 시나리오 명세를 통해 각 미디어 객체가 프리젠테이션 될 때 필요한 시공간 정보를 나타낸 것이다. 사용자가 정의한 시나리오 명세는 다음과 같다 : 오디오 A와 함께 쇼가 시작된다. 오디오 A는 38초까지 재생되다가 2초 동안 소리가 점점 작아지면서 40초가 되면 재생을 중단하게 된다. A의 프리젠테이션이 시작된 후 5초 후에는 사용자가 정의한 도형 S가 화면에 나타난다. 텍스트 T는 8초에서부터 12초 동안 표현되고, S와 함께 사라진다. 네트워크의 원격지 저장장치에서 검색된 비디오 클립 V가 8초에 화면에 재생되기 시작하여 32초간 재생되게 된다. 이미지 I는 20초부터 35초까지 화면에 보여지게 된다. 이때, 이미지 I는 20초에 화면 좌표 (x1, y1, x2, y2)에 작은 크기로 나타나서(120) 5초동안 (x3, y3, x4, y4)로 위치를 이동하고(125), 남은 시간 10초 동안 화면에 보이다가 사라진다.



▶▶ 그림 1. 시나리오 시간 구성



▶▶ 그림 2. 시나리오 공간 구성

2. 주요 인스턴스 설정

'object'와 'effect', 그리고 'actor'는 아래와 같이 정의된다.

Definition of Object

```
Object A={(1),(null:"c:\media"),"Audio.wav","Author",...}
Object V={(2),("\\MDBMS":"c:\MDB"),"Video.mpg","Server",...}
Object T={(3),(null:"c:\media"),"Text.txt","Author",...}
Object I={(4),(null:"c:\Image"),"Image.bmp","Author",...}
Object S={(5),(null:null),null,"User",...}
```

Definition of Effect

```
Effect A(1.1)={(1.1),null,("00:00:00", "00:00:40"),("play"),...}
Effect A(1.2)={(1.2),null, ("00:00:38", "00:00:02"), ("volume
control"),(1.0, 0.0), ...}
Effect S(5.1)={(5.1), (sx1, sy1, sx2, sy2), ("00:00:05",
"00:00:15"), ("show"), ...}
```

Definition of Actor

```
Actor A={"Object A", ("Effect A(1.1)", "Effect(1.2)")}
Actor V={"Object V", ("Effect V(2.1)")}
Actor T={"Object T", ("Effect T(3.1)")}
Actor I={"Object I", ("Effect I(4.1)", "Effect I(4.2)")}
Actor S={"Object S", ("Effect S(5.1)")}

```

3. Event 스케줄

정의된 시나리오는 프리젠테이션이 진행되기 이전에 IMPSS의 내외부적 상태를 반영하여 스케줄을 작성하여야 한다. 본 논문에서는 정의된 시나리오의 시간적 순서를 기반으로 'event'의 스케줄을 작성하였다. 아래에 'actor'에 관한 'event'를 중심으로 프리

젠테이션 되기 직전의 스케줄된 내용을 일부 나타내었다.

The Scheduled Event List in Queue

```
{(P1), "Open object", ("Sync. Controller",
"Actor A"), "Object A", "Urgency", ...}
{(P2), "Play effect", ("Presentation Scheduler",
"Actor A"), "Effect A(1.1)", "Urgency", ...}
:
{(P10), "Stop effect", ("Presentation Scheduler",
"Actor S"), "Effect S(5.1)", "Urgency", ...}
{(P11), "Close object", ("Sync. Controller",
"Actor T"), "Object T", "unUrgency", ...}
:
{(P21), "Close object", ("Sync. Controller",
"Actor V"), "Object V", "unUrgency", ...}
{(P22), "Close object", ("Sync. Controller",
"Actor A"), null, "unUrgency", ...}
```

4. 프리젠테이션

프리젠테이션 도중에 네트워크의 이상으로 인하여 'object V'의 스트리밍 전송이 지연되었을 경우, 시간상에 관련된 다른 'actor'의 'effect'도 사용자가 허용할 수 있는 범위에서 지연 처리함으로써 서비스 품질 저하를 최소화 할 수 있다. 10초 시점에서 'object V'의 데이터 전송 지연이 발생하고, 그로 인해 약 1초간 다른 'actor'의 쇼도 지연시켜 진행해야 할 경우를 가정한 'effect events'를 보였다.

```
Effect speed(Sm.1)={(sm.1), null, ("00:00:15",
"00:00:01"), ("speed control"), ...}
{(Sm), "Speed Effect", ("Sync. Controller", "Actor
A"), "Effect Speed(Sm.1)", "Urgency", ...}
{(Sm+1), "Speed Effect", ("Sync. Controller",
"Actor V"), "Effect Speed(Sm.1)", "Urgency", ...}
{(Sm+2), "Speed Effect", ("Sync. Controller",
"Actor T"), "Effect Speed(Sm.1)", "Urgency", ...}
{(Sm+2), "Speed Effect", ("Sync. Controller",
"Actor S"), "Effect Speed(Sm.1)", "Urgency", ...}
```

만약 사용자가 28초에서 일시 정지 버튼을 선택하게 되면, 논리적 프리젠테이션 시간(the logical presentation timer)은 정지하게 되며, 사용자 'event Un'가 현재 스케줄러에 대기중인 모든 'actor'에게 일시정지(suspend) 'event'를 통보할 것을 나타낸다. 아래에 'event Un'의 내용을 보였으며, 'actor A', 'actor V', 그리고 'actor I'에게 전달될 'Pause effect event'를 보였다.

```
{(Un), "Pause Presentation", ("User Input",
"Sync. Controller"), "All Actors", "Urgency", ...}
{(Sn), "Pause Effect", ("Sync. Controller",
"Actor A"), null, "Urgency", ...}
{(Sn+1), "Pause Effect", ("Sync. Controller",
"Actor V"), null, "Urgency", ...}
{(Sn+2), "Pause Effect", ("Sync. Controller",
"Actor I"), null, "Urgency", ...}
```

사용자의 프리젠테이션 재개 명령은 일시정지되었던 'actor'에게 재개 'event'를 전달함으로써 가능해진다.

V. 결론

본 논문은 효율적인 대화형 멀티미디어 프리젠테이션을 수행하기 위하여 'event' 기반의 방안에 대하여 제안하였다. 'event'의 종류나 특성에 따라 여러 'event'로 구분하였으며, 동기화 정보로 활용하여 여러 모듈이 협력하여 처리할 수 있도록 하였다.

이를 위해서, 본 논문에서는 시나리오로 구성될 수 있는 미디어에 대한 정보를 두 가지로 구분하였다. 첫째는 순수하게 미디어 자체에 대하여 처리하는 'object'이며, 둘째는 'object'가 프리젠테이션 될 때 표현하는 방법과 시공간에 대하여 정의하는 'effect'이다. 또한 이 두 가지 중요한 정보 'actor'로 재구성하였다. 프리젠테이션 과정에는 IMPSS의 내 외부적인 예측 불가능한 사건으로 인하여 QoS 저하에 대한 문제가 발생할 수 있다. 그러한 사건발생에 대해

'user', 'actor', 'system', 그리고 'scheduler event'로 구분하였으며, 이러한 각 부분이 서로 협력할 수 있도록 동기화 정보를 포함하는 event로 통신하여 동시에 여러 상태 변화를 효과적으로 처리할 수 있도록 하는데 중점을 두었다.

본 논문에서 제안한 'event'기반의 프리젠테이션 모델은 현재 윈도우 XP 기반으로 Visual C++를 이용하여 구현중이며, 실시간 네트워크를 이용한 프리젠테이션 기능을 추가할 예정이다.

■ 참고 문헌 ■

- [1] M. Vazirgiannis and S. Boll, "Events in Interactive Multi-media Applications: Modeling and Implementation Design," Proc. of IEEE Int'l Conf. on Multimedia Computing and Systems (ICMCS 97), IEEE CS Press, Los Alamitos, Calif., June 1997, pp.244-251.
- [2] M. Vazirgiannis, T. Sellis. "Event and Action Representation and Composition for Multimedia Application Scenario Modelling". ERCIM Workshop on Interactive Distributed Multimedia Systems and Services, BERLIN, 3/1996.
- [3] Thomas Wahl, Stefan Wirag and Kurt Rothermel. "TIEMPO: Temporal Modeling and Authoring of Interactive Multimedia", In Proc. of IEEE 2nd Intl. Conference on Multimedia Computing and Systems, pp. 274-277, Washington DC, 5 1995
- [4] N. Hirzalla, B. Falchuk, and A. Karmouch. A Temporal Model for Interactive Multimedia Scenarios. IEEE Multimedia, 2(3):24--31, Fall 1995.
- [5] M. Jourdan, N. Layaida, C. Roisin, L. Sabry-Ismail, L. Tardif (1998). Madeus, an Authoring Environment for Interactive Multimedia Documents. In Proceedings: ACM Multimedia '98, Bristol UK.
- [6] S. Chakravarthy, D. Mishra. "Snoop: An Expressive Event Specification Language For Active Databases". Technical Report, UF-CIS-TR-93-00, University of Florida, 1993.

- [7] N.H. Gehani, H.V. Jagadish, O. Shmueli. "Composite Event Specification in an Active Databases: Model & Implementation". Proc. of VLDB Conference, 1992, pp.327-338.
- [8] N.H. Gehani, H.V. Jagadish, O. Shmueli. "Event Specification in an Active Object Oriented Database", In Proc. of the ACM/SIGMOD Conference, 1992.
- [9] G. Schloss, M. Wynblatt. "Providing Definition and Temporal Structure from Multimedia Data", Multimedia Systems Journal, Vol.3, 1995, pp.264-277.
- [4] N. Hirzalla, B. Falchuk, A. Karmouch. "A Temporal Model for Interactive Multimedia Scenarios". IEEE Multimedia, Fall 1995, pp.24-31.
- [10] G. Schloss, M. Wynblatt. "Providing Definition and Temporal Structure from Multimedia Data", Multimedia Systems Journal Vol.3, 1995, pp.264-277.