

# 분산 컴퓨팅환경에서 하이퍼 프리젠테이션을 위한 통합 동기화 기법

임 영 환<sup>†</sup> · 김 두 현<sup>††</sup> · 공 상 환<sup>††</sup>

## 요 약

멀티미디어의 연출을 순차적으로만 하는 것이 아니라 링크개념을 설정하여 특정지점에서 원하는 다음 정보를 보기 위하여 비순차적으로 연출할 수 있도록 하고 그 하위수준의 연출이 완료되면 다시 원래 프리젠테이션으로 되돌아 계속 연출하는 것이 하이퍼프리젠테이션의 개념이다. 이러한 순차적인 연출 시스템에 비순차적인 연출방법인 하이퍼 프리젠테이션 개념을 도입함에 있어서 하이퍼 프리젠테이션의 명시방법이나 연속미디어상의 하이퍼 링크를 위한 hot-spot 설정방법, 혼합된 프리젠테이션간의 일관성 검증방법 등의 문제점이 발생한다. 본 논문에서는 우선 기존의 동기화 명시방법을 살펴보고 하이퍼 프리젠테이션을 위한 새로운 동기화 명시방법과 하이퍼 링크 설정 방법을 제안하였다. 그리고 DCT변환방식을 이용하여 압축된 미디어에 하이퍼 링크를 실시간으로 설정하는 방법을 제안하였다. 또한 위에서 제시된 명시방법을 이용하여 나타낸 하이퍼 프리젠테이션 프로그램이 연출 가능한지 미리 검사할 수 있는 방법도 제안하였다. 마지막으로 지금까지 제안된 방법을 MuX(Multimedia Input/Output Server)에 구현하여 실제 연출될 수 있는 시나리오를 대상으로 실험하였다.

## An Integrated Synchronization Method for a Hyperpresentation in a Distributed Computing Environment

Younghwan Lim<sup>†</sup> · Doohyun Kim<sup>††</sup> · Sanghwan Kung<sup>††</sup>

### ABSTRACT

The concept of a hyperpresentation, as an extension of a hypermedia, is the presentation in which time-varying multimedia presentations are dynamically linked together and a hyperlink's context can be changed over time at any time during a continuous presentation. Problems caused by integrating the hyperpresentation into an existing multimedia system which handles a sequential presentation only are, how to describe the hyperpresentation, how to set up a hyperlink on a continuous media, and how to check the consistency of the synchronized presentations. In this paper, a new synchronization description method for the hyperpresentation and a method for setting a hyper link on a continuous media during a presentation are proposed after having survey of existing methods. The proposed method deals with only the DC value in a stream of a DCT based compressed data for checking a condition of the link. Finally, the method for checking the consistency of mixed presentations before actual play of the hyperpresentation is described. Proposed methods are implemented on MuX(Multimedia IO Server), where a sample scenario is tested.

† 종신회원 : 숭실대학교 부교수

†† 정 회 원 : 한국전자통신연구원 책임연구원

논문접수 : 1998년 2월 7일, 심사완료 : 1998년 4월 13일

## 1. 서론

하이퍼프리젠테이션 개념은 기존의 멀티미디어 연출에 하이퍼 미디어개념을 확장한 것이라고 보면 된다. 즉 멀티미디어의 연출을 순차적으로만 하는 것이 아니라 링크개념을 설정하여 특정지점에서 원하는 다음 정보를 보기 위하여 비순차적으로 연출할 수 있도록 하는 것이다. 그리고 하위수준의 연출이 완료되면 다시 원래 연출로 되돌아 와서 계속하게 된다.

지금까지의 멀티미디어 시스템은 미디어 스트림의 순차적인 연출을 위한 동기화방법에 관심을 기울여 왔다. 이러한 순차적인 연출 시스템에 비순차적인 연출방법인 하이퍼 프리젠테이션 개념을 도입함에 있어서 2장에서 설명한바와 같은 문제점이 발생한다.

첫째 본 논문에서는 기존의 동기화 명시방법을 살펴보고 하이퍼프리젠테이션을 위한 새로운 동기화 명시방법을 제안하였다. 하이퍼프리젠테이션을 명시하는데 있어 중요한 점은 프리젠테이션 자체를 하나의 객체로 보고 그들간의 동기를 명시한다는 점이다. 즉 하이퍼프리젠테이션 계층에서는 프리젠테이션을 구성하는 미디어 스트림이나 그들간의 동기화 문제는 보이지 않는다는 점이다. 그렇게 함으로 프리젠테이션 계층에서 사용했던 명시 방법을 하이퍼프리젠테이션 계층의 동기화를 위하여 적용할 수 있다. 그 외 하이퍼 링크를 설정함으로써 보다 자유로운 프리젠테이션의 수행이 가능해 지도록 확장하였다.

두 번째로 본 논문에서 하이퍼 프리젠테이션을 위하여 프리젠테이션 상에 하이퍼 링크를 설정하는 방법을 제안하였다. 분산 멀티미디어 프리젠테이션에 하이퍼 링크 설정을 한다는 것은 내부적으로 hot-spot을 지정한다는 의미와 같다. 지금까지는 hot-spot이 설정된 이미지나 텍스트로서의 하이퍼 텍스트 개념으로 많이 쓰였었다. 그러나 비디오나 오디오와 같은 연속 미디어 상에서의 hot-spot설정은 정적인 대상을 지정할 수 없기 때문에 그것과 전혀 다르다. 그리고 여러 미디어가 동기화된 프리젠테이션상의 hot-spot설정은 또 다른 방법을 요구한다. 더욱이 문제가 되는 것은 압축된 연속 미디어 상에서 실시간으로 하이퍼 링크의 조건 검사가 가능해야 한다는 점이다. 본 논문에서는 DCT변환 방식을 이용하여 압축된 미디어에 하이퍼 링크를 실시간으로 설정하는 방법을 제안하였다.

세 번째로 본 논문에서는 동기화 명시프로그램에서

혼합된 여러 프리젠테이션간에 발생할 수 있는 프로그래밍 에러를 검출할 수 있는 방안을 제안하였다. 하이퍼 프리젠테이션을 명시하는 것은 사용자의 의지를 표명한 것으로 실제 연출될 수 있으나 하는 것은 또 다른 문제이다. 여기서 실제 연출을 시작하기 이전에, 즉 컴파일 할 때, 명시 프로그램에서 흔히 발생할 수 있는 문제의 여러 경우를 살펴보고 그 문제를 해결하는 방법을 제시하였다.

마지막으로 지금까지 제안된 방법을 MuX(Multimedia Input/Output Server)에 구현하여 실제 연출될 수 있는 시나리오를 대상으로 실험하였다.

본 논문의 구성은 2장에서 하이퍼프리젠테이션의 개념 및 기존의 연구동향 그리고 문제점 및 필요성을 살펴보고 3장에서 정형화된 동기화 명시방법을 제안하였다. 그리고 4장에서 동기화 명시프로그램에서 발생하는 hot-spot설정하는 문제와 연출가능성을 검증하는 문제를 해결하는 방법을 제시하였다. 그리고 5장에서 실제 구현한 예를 보이고 6장에서 결론을 맺었다.

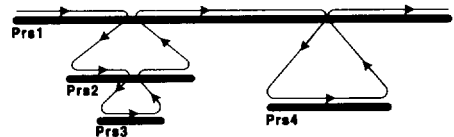
## 2. 하이퍼 프리젠테이션의 개념 및 문제점

### 2.1 하이퍼 프리젠테이션의 개념

하이퍼 프리젠테이션의 개념은 기존의 멀티미디어 연출(presentation)에 하이퍼 미디어 개념을 확장한 것이라고 보면 된다[8]. 즉, 멀티미디어의 연출을 순차적으로만 하는 것이 아니라 링크 개념을 설정하여 특정 지점(hot-spot)에서 원하는 다음 정보를 보기 위하여 비순차적으로 연출할 수 있도록 하는 것이다. 그러면서도 하위 수준의 연출이 완료되면 다시 원래 수준의 연출로 되돌아와 계속 수행한다[1].

### 2.2 하이퍼 프리젠테이션의 예

하이퍼 프리젠테이션의 간단한 예로 그림 1의 뉴스예제를 들어보겠다. 뉴스를 진행하는 앵커의 화면과 음



(그림 1) 뉴스 예제  
(Fig. 1) An example of news presentation

성, 배경 음악으로 구성되는 하나의 프리젠테이션(Prs1)이 주 흐름이 되며, 진행 도중 사건 보도를 담은 프리젠테이션들(Prs2, Prs3, Prs4)이 순서에 따라 연결이 된다.

앵커가 의원 선거에 대한 기사 예고를 하면, 후보연설 현장에 나와 있는 기자(Prs2)가 계속해서 진행을 한다. 잠시 후, 어느 한 후보와의 인터뷰 내용(Prs3)이 나오고 이어서 선거에 대한 보도(Prs2)가 끝나면 앵커가 다음 뉴스(Prs1)를 계속 진행한다. 곧이어 기상캐스터의 진행과 배경 음악, 배경 화면으로 구성되는 내일의 날씨(Prs4)가 계속되고, 앵커의 마감 멘트와 배경 음악(Prs1)으로 뉴스는 끝난다.

### 2.3 하이퍼 프리젠테이션의 문제점

이러한 하이퍼 프리젠테이션의 표현을 위해 고려되어야 하는 문제점은 다음과 같다.

- 하이퍼 프리젠테이션이 포함된 멀티미디어 동기화(synchronization)를 명시(description)하는 기법
- 연속 미디어를 포함한 프리젠테이션 상에 hot-spot을 지정하는 문제
- 여러 동기화 방법으로 혼합된 프리젠테이션간의 일치성을 검사문제

첫 번째 문제점은 하이퍼 프리젠테이션을 명시하는 정형적인 방법이 있어야 한다는 점이다. 멀티미디어 연출을 위한 시간적 동기화를 명시하는 방법으로는 연출할 미디어간의 관계(relation)를 가지고 연출의 시간적 우선 순위를 명시하는 간격 기반 명시방법(Interval-based Specification)[9] 과 광역시간대에 대한 미디어연출 시간을 명시하는 시간축 기반 명시방법(Time Axes-based Specification)[10], 전체 프리젠테이션의 구조를 명시하는 제어흐름 기반 명시방법(Control Flow-based Specification)[11], 페트리 넷(Petri Net)를 이용하여 미디어간의 사건(event)관계를 명시하는 사건 기반 명시방법(Event-based Specification)[12], 그리고 미디어간의 동기 시나리오를 문서적으로 기술하는 스크립(Script) 방법 등이 있다. 그러나 이러한 방법들은 미디어간의 시간적인 관계를 이론적으로 나타내는데는 충분하나 어느 한 동기화 명시방법이 실제 연출상황을 전부 명시할 수 없을 뿐만 아니라 어느 한 프리젠테이션을 연출하는 도중 다른 프리젠테이션을

연출하게 하는 하이퍼프리젠테이션을 명시하는 방법을 제공하지 못한다. 따라서 기존의 프리젠테이션의 동기화 명시방법을 통합하고 하이퍼프리젠테이션 기능까지 명시할 수 있는 새로운 하이퍼프리젠테이션 명시방법이 필요하다.

두 번째 문제점은 프리젠테이션 상에 하이퍼프리젠테이션을 위한 링크(hot-spot)를 설정하는 것이다. 대표적인 하이퍼미디어시스템은 텍스터 모델(Dexter Model)[13]과 암스테르담 모델(Amsterdam Model)[14] 등이 있다. 이러한 모델은 주로 정적인 링크를 사용하고 아주 제한적인 동적 링크를 제공하고 있어 프리젠테이션 상에서 하이퍼 링크를 설정하는 방법은 새롭게 연구되어야 한다. 최근에 비디오 상에 미리 설정된 움직이는 객체를 하이퍼 링크로 설정하여 그 객체가 이동하더라도 그에 맞게 하이퍼 링크도 움직이는 연구가 제한적으로 이루어지고 있다. 그러나 중요한 문제점은 어느 한 연속미디어 상에서 hot spot을 설정하는 것이 아니라 여러 미디어가 통합되어 있는 프리젠테이션 상에서 그것이 설정되어야 하고 또한 실시간으로 검출될 수 있어야 한다는 점이며 그리고 연속 미디어의 대부분이 압축된 데이터라는 점이다.

마지막 문제로 여러 프리젠테이션 미디어간의 관계를 다양한 명시방법을 통합하여 기술하였을 때 과연 그것이 전체적으로 연출 가능한지 아닌지를 미리 검증해야 한다는 것이다. 지금까지의 멀티미디어 스케줄링에 관한 연구는 ARTS(Advanced Real Time Technology Operating System)[15], YARTOS(Yet Another Real Time Operating System)[16], 스플릿 레벨 스케줄링(split level scheduling)[17] 등이 있거나 이것은 이미 연출이 가능하다고 보고 연출에 필요한 자원(resource)을 QoS(Quality of Service)에 맞도록 배분하는데 중점을 두고 있을 뿐 연출하기 전에 미리 연출가능 여부를 미리 검증하는 방법은 제공하지 않고 있다. 따라서 하이퍼프리젠테이션 명시방법에 의해 기술된 것을 실행하기 전에 미리 컴파일 하여 연출가능 여부를 판단하는 방법의 연구가 필요하다.

## 3. 하이퍼 프리젠테이션을 위한 새로운 동기화 명시기법

### 3.1 동기화 메카니즘 정의

프리젠테이션은 시간적 동기화가 설정된 미디어 스

트림들의 집합으로 볼 수 있다. 시간적 동기화를 명시하는 방법에는 앞장에서 살펴본 바와 같이 여러 가지가 있겠지만 본 논문에서는 시간축 기반 명시방법과 간격 기반 명시방법을 통합한 방법을 사용하였다. 즉 프리젠테이션 계층에서의 시간적 동기화는 논리적 시간 체계(LTS: Logical Time System)와 미디어간의 관계(relation)에 의해 이루어진다.

하이퍼프리젠테이션을 명시하는데 있어 중요한 점은 위에서 정의된 프리젠테이션 자체를 하나의 객체로 보고 그들간의 동기를 명시한다는 점이다.

3.1.1 논리적 시간 체계 (LTS)

LTS(Logical Time System)는 멀티미디어 통합 및 동기화 모델의 시간을 관장한다. LTS는 동기화의 기본 단위이며, 일정한 시간 간격마다 시간의 흐름을 알려준다. 이러한 LTS를 아래와 같이 일정한 비율로 증가하는 시간값의 배열로 볼 수 있다.

[정의3.1] LTS  $T = \langle t_i | i \text{는 정수} \rangle$

하이퍼 프리젠테이션을 구성하는 프리젠테이션마다 하나의 광역 LTS를 축으로 하여 시작 시간값과 끝나는 시간값을 지정함으로써 동기화를 이룰 수 있다.

[정의3.2] 한 개의 프리젠테이션  $prsP$ 가 어느 하나의 시간축  $T_i$ 에 동기화 되었다함은 그  $prsP$ 의 시작 시간과 끝나는 시간이 시간축  $T_i$ 에 있는 값으로 정해졌다는 것을 의미하며  $start\_time$ 과  $end\_time$ 이  $T_i$ 의 시간값이라고 할 때  $prsP(T_i, \{start\_time, end\_time\})$ 로 표기한다.

LTS로서 지원 가능한 기능은 시작, 끝, 정지, 재시작, 시간이동, 시간정지 등이다. LTS의 기간은 미리 정할 수도 있고 무한대로 할 수도 있다. 이 시간 체계는 논리적인 것으로서 실제 시간과 계산에 의해 맞춰 나갈 수 있고, 미리 정의된 시간 간격을 조정해 프리젠테이션의 연출 속도를 조정할 수 있다. [1][6]

하이퍼 프리젠테이션 계층에서의 시간적 동기화가 이루어지면, 하이퍼 프리젠테이션 계층의 하부 구조들, 즉 프리젠테이션 계층의 요소들(프리젠테이션, 채널, 트랙)과 스트림 계층의 요소들(source)의 실행 시작 시간까지 영향이 미친다. 각각의 요소들은 별도의 시간

축을 갖고 있기 때문이다. 예를 들어, 하이퍼 프리젠테이션의 시작 시간이 광역 LTS를 기준으로  $\alpha$ , 프리젠테이션의 시작 시간이  $\beta$ , 채널의 시작 시간이  $\gamma$ , 트랙의 시작 시간이  $\delta$ , 그리고 스트림의 시작 시간이  $\epsilon$ 라 할 때, 각 요소들의 시작 시간은 다음과 같다.

<표 1> 하이퍼 프리젠테이션 구성 요소들의 시작 시간  
<Table 1> Starting times of components in a hyperpresentation

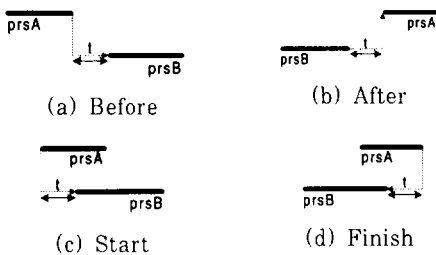
계    층	시  작  시  간
하이퍼 프리젠테이션	$\alpha$
프리젠테이션	$\alpha + \beta$
채  널	$\alpha + \beta + \gamma$
트  랙	$\alpha + \beta + \gamma + \delta$
스트림	$\alpha + \beta + \gamma + \delta + \epsilon$

3.1.2 큐 (Cue)

간격 기반 명시방법은 미디어들간의 시간적인 관계(temporal relationship)를 *before, meets, overlaps, during, starts, finishes, equal, before<sup>-1</sup>, meets<sup>-1</sup>, overlaps<sup>-1</sup>, during<sup>-1</sup>, starts<sup>-1</sup>, finishes<sup>-1</sup>* 등으로 정의하여 동기화를 맞추었다. 하이퍼프리젠테이션의 동기화를 명시하기 위하여 이 개념을 프리젠테이션 객체까지 확장하여 다음과 같은 큐 개념을 정의하였다.

두 개의 프리젠테이션간의 관계를 살펴보면 다음과 같다.

- ① A Before(t) B ⇔ 그림 2의 (a)  
A 프리젠테이션은 B 프리젠테이션의 시작보다 t만큼 전에 끝난다.
- ② A After(t) B ⇔ 그림 2의 (b)  
A 프리젠테이션은 B 프리젠테이션이 끝나면 t만큼 후에 시작한다.
- ③ A Start(t) B ⇔ 그림 2의 (c)  
A 프리젠테이션은 A가 시작된후 t만큼 뒤에 B 프리젠테이션이 시작되게 한다.
- ④ A Finish(t) B ⇔ 그림 2의 (d)  
A 프리젠테이션은 A가 끝나기 t만큼 전에 B 프리젠테이션이 끝나도록 한다.



(그림 2) 큐(Cue)의 Relation 속성  
(Fig 2) The relational characteristics of cue

위 그림 2에서 기준 프리젠테이션과 종속 프리젠테이션으로 분리함으로 각각의 역함수(inverse function),  $before^{-1}$ ,  $after^{-1}$ ,  $start^{-1}$ ,  $finish^{-1}$ 등을 별도로 정의하지 않더라도 기준과 종속의 교환으로 같은 효과를 얻을 수 있다.

[정의3.3] **큐(cue)**는 두 개의 프리젠테이션간의 관계로서 다음 중 하나의 관계값을 가지며 그 속성은 위와 같다.

*before, after, start, finish.*

[정의3.4] **두 개의 프리젠테이션이 큐에 의해 동기화 되었다함**은 두 개의 프리젠테이션이  $before(t)$ ,  $after(t)$ ,  $start(t)$ ,  $finish(t)$ 중 하나로 관계가 맺어졌다는 것을 의미하며 다음과 같이 나타낸다.

$$prsA \ Q(t) \ prsB$$

여기서  $Q(t)$ 의 앞에 있는 prsA가 관계  $Q$ 의 기준 프리젠테이션이고 prsB는 종속프리젠테이션이다.

지금까지 정의한 것은 두 개의 프리젠테이션만을 대상으로 하였는데 실제 이것은 두 개보다 많은 여러 프리젠테이션까지 확장할 수 있다. 그러면 프리젠테이션간 연출할 수 없는 문제까지 발생할 수 있는데 이러한 문제는 다음 장에서 다루기로 한다.

### 3.1.3 하이퍼 링크

하이퍼프리젠테이션은 어느 한 프리젠테이션을 연출하는 도중에 일시 중지하고 다른 프리젠테이션을 수행한 후 다시 원래의 프리젠테이션 연출을 계속한다는 것이다. 여기서 문제가 되는 점은 하나의 프리젠테이션

상에서 다른 프리젠테이션을 연출하기 위한 지점(하이퍼 링크점)을 어떻게 설정하는가 하는 것이다. 본 논문에서는 하이퍼 링크는 조건(condition)과 행동(action)의 순서쌍으로 정의하였다.

[정의3.5]  $C$ 를 프리젠테이션 상에 설정할 수 있는 조건들의 집합이라고 하고  $P$ 를 조건이 만족되면 연출할 프리젠테이션의 집합이라고 하자. 그러면 **하이퍼 링크**  $HL = \{(cond, prs) \mid cond \in C \text{ and } prs \in P\}$ 이다.

여기서 중요한 점은 조건을 어떻게 설정하는가 하는 문제이다. 예로서 가장 단순한 조건은 광역 시간축상의 시간적인 어느 한 지점(At)이나 기간(During)이 가능하다. 즉 프리젠테이션을 연출하는 도중 지정된 시간에 다른 프리젠테이션을 연출하고 돌아와서 원래의 프리젠테이션을 계속 수행한다는 개념이다. 이밖에 조건설정에 관한 좀더 상세한 연구는 다음 장에서 다루기로 한다. 또 다른 중요한 점은 행동(action)을 어떤 종류로 할 것인가 이다. 위 정의에서는 단순히 "주어진 프리젠테이션을 수행" 하는 행동으로만 한정되었는데 좀더 확장하면 "아래한글 수행"이나 "경고 메시지 발송"등과 같은 다양한 행동(action)을 설정할 수 있다.

[정의3.6] **프리젠테이션 prsA가 프리젠테이션 prsB를 하이퍼 링크로 연결하였다 함**은 prsA상에 prsB를 포함하는 하이퍼 링크가 prsA상에 적용할 조건이 설정되었음을 의미하여 다음과 같이 나타낸다.

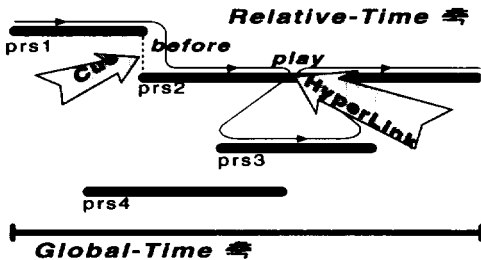
$$prsA \ HL(cond) \ prsB$$

### 3.2 하이퍼 프리젠테이션의 동기화 명시 방안

하이퍼 프리젠테이션 계층은 LTS와 큐(Cue), 그리고 하이퍼 링크에 의해 프리젠테이션들간의 동기화를 설정하므로써 표현된다. 기본적으로 하이퍼 프리젠테이션의 시간적 동기화는 LTS에 의한 광역시간축을 기반으로 이루어진다. 하이퍼 링크와 큐에 의해 연결된 프리젠테이션들은 광역시간축보다는 큐나 하이퍼링크의 관계성에 우선 순위를 두어 수행된다.

그림 3에서 보면, prs1을 "Before" 큐로 prs2에 설정한 경우 prs1은 관계성에 묶이게 된다. 여기서 광역

시간(global time)축을 기반으로 진행되는 프리젠테이션은 prs2와 prs4뿐이다. 즉 prs4는 광역시간축 상에 시작시간이 되면 연출을 시작하여 끝나는 시간에 연출을 종료하게 된다. 그러나 prs2는 prs1과 before로 큐되어 있기 때문에 prs2가 시작되는 시점으로부터 prs1의 연출시간만큼 먼저 연출을 시작한다. 그리고 prs1연출이 종료되면 prs2의 연출을 시작한다. 프리젠테이션 prs2는 prs1을 관계(before)에 맞게 실행시킬 뿐, 두 프리젠테이션은 서로 독립적으로 수행된다. 또한 prs2에 설정된 하이퍼 링크의 액션으로 수행되는 prs3은 설정된 조건(현재는 prs3에 적용된 시간축임)이 만족되면 실행되며, 실행이 끝나면 다시 prs2를 계속 수행한다.



(그림 3) 하이퍼 프리젠테이션의 동기화 Time 축  
(Fig. 3) The time axe for hyperpresentation synchronization

3.2.1 하이퍼프리젠테이션

윗 절에서 설명한바와 같이 하이퍼프리젠테이션은 다음과 같이 정의될 수 있다.

[정의3.7] 프리젠테이션들의 집합 P와 큐의 집합 Q, 그리고 하이퍼 링크의 집합HL이 주어져 있다면 하이퍼프리젠테이션은 다음같이 정의한다

1. P내에 있는 모든 프리젠테이션 hprsP 그 자체도 하이퍼프리젠테이션이다. 이것을 특히 단순 하이퍼프리젠테이션(simple hyperpresentation)이라고 부른다.
2. 만약 hprsA와 hprsB가 하이퍼프리젠테이션이라면 hprsA Q<sub>i</sub>(t) hprsB 또는 hprsA HL<sub>j</sub>(t) hprsB도 하이퍼프리젠테이션이다. 여기서 Q<sub>i</sub>는 어느 한 큐로서 Q의 원소이고 HL<sub>j</sub>는 하이퍼 링크로서 HL의 원소이다. 이렇게 만들어 진 것을

복합 하이퍼프리젠테이션(composite hyperpresentation)이라고 부른다.

3. 하이퍼프리젠테이션이 되기 위한 필요충분조건은 단순하이퍼프리젠테이션에 규칙2)를 유한 번(finite) 적용하여 얻어진 것이다.

위 정의와 같이 하이퍼프리젠테이션을 정의하면 너무 포괄적인 면도 있지만 본 논문에서는 하이퍼프리젠테이션계층에서 다루는 모든 메카니즘을 포함하여 동기화를 명시할 수 있도록 하기 위하여 확대 정의하였다.

[예3.1] 그림3에서 살펴보면 prs<sub>4</sub>와 prs<sub>1</sub> before(0) prs<sub>2</sub> HyperLink(0) prs<sub>3</sub> 은 서로 독립적인 하이퍼프리젠테이션이다. 물론 prs1과 prs2, prs3도 (단순) 하이퍼프리젠테이션이다.

위 정의에서 다음과 같은 성질을 쉽게 발견할 수 있다.

[성질3.1] 복합 하이퍼 프리젠테이션의 연출시간은 그것을 구성하고 있는 모든 프리젠테이션 연출시간의 합과 같다.

[성질3.2] 단순 하이퍼 프리젠테이션은 시작과 끝시간을 광역시간축에 대응시킬 수 있지만 복합 하이퍼 프리젠테이션은 끝나는 시간을 광역시간축으로 지정할 수 없다.

위의 성질을 근거로 하면 하이퍼 프리젠테이션의 시간적인 동기화를 위하여 광역시간축만을 이용한 동기화 명시방법이 불가능함을 알 수 있다. 따라서 하이퍼 프리젠테이션을 시간축에 동기화 시킬 때의 정의는 프리젠테이션을 시간축에 동기화 시킬 때의 정의와 달라지게 될 수밖에 없다.

[정의3.8] 한 개의 하이퍼 프리젠테이션 hprsP가 어느 하나의 시간축 T<sub>i</sub>에 동기화 되었다함은 그 hprsP의 시작 시간이 시간축 T<sub>i</sub>에 있는 값으로 대응되었다는 것을 의미하며 start\_time이 T<sub>i</sub>의 시간값이라고 할 때 hprsP(T<sub>i</sub>, start\_time)로 표기한다. 이때 hprsP가 복합하이퍼 프리젠테이션이면 start\_time은 실제 시작시간이 아닐 수 있다. 그리고 끝나는 시간은 hprsP를 구성하고 있는 프리젠테이션들의 연

출시각의 합과 같다.

3.2.2. 분산 하이퍼 프리젠테이션을 위한 통합 동기화 명시 방법

지금까지 정의한 동기화 메카니즘을 이용하여 하이퍼프리젠테이션의 동기화를 명시하는 방법을 정의해보면 다음과 같다.

[방법3.1] 하이퍼 프리젠테이션 동기화 명시방법

하이퍼프리젠테이션 동기 명시방법인 HPSS는 다음과 같이 4개의 구성원소를 갖는 것으로 정의되는데,

$$HPSS = (T, Q, L, HP).$$

단,

T 는 광역시간축으로 어느 한 LTS이고,

Q 는 Cue의 집합,

L 은 하이퍼 링크의 집합,

HP 는 시간축 T 에 시간적으로 동기화되어 있는 하이퍼프리젠테이션의 집합이다.

[예3.2] 위 그림2의 하이퍼프리젠테이션을 다음과 같이 나타낼 수 있다.

```
HPSS3 = (
Global_LTS,
{before(0)},
{time_at_40},
{
    prs4(Global_LTS, 10),
    (prs1 before(0) prs2 time_at_40(0)
    prs3)(Global_LTS,40)
}
)
```

위의 명시 방법 대로 연출하는 데 있어서 프리젠테이션의 시작을 결정하는 요소가 Global\_LTS와 큐 그리고 하이퍼 링크 등 3가지가 있다. 따라서 실제 연출의 우선 순위를 정해야 하는데 본 논문에서 제안하는 방식은 다음과 같다.

[방법3.2] 연출시간 우선 순위 조정 방법

1. 어느 한 프리젠테이션에 시간축, 큐, 하이퍼 링크 등이 동시에 적용될 때 시간축보다 큐나 하이퍼 링크가 우선 순위가 높다.

2. 같은 하이퍼링크에 2개이상의 프리젠테이션이 수행될 때 그들간의 순위는 명시방법에서 사용자가 선택하도록 한다

4. 연출상의 문제점 및 해결 방안

지금 까지 하이퍼 프리젠테이션의 동기를 명시하는 방법을 제안하였다. 이러한 명시방법은 복잡한 하이퍼 프리젠테이션 상황을 기술할 때 사용자가 의도하지 않았던 여러 가지 문제가 발생할 수 있다. 우선 가장 큰 문제는 정적인 미디어가 아닌 비디오나 오디오와 같은 동적인 미디어를 포함한 하이퍼 프리젠테이션 상에 어떻게 하이퍼링크를 설정할 것인가 하는 것이다. 그리고 다른 문제는 이렇게 명시된 복잡한 하이퍼프리젠테이션이 실제 연출 가능한 것인지 미리 검증하는 것이다. 이 장에서는 비디오 상에서 하이퍼 링크를 설정하는 문제와 명시된 하이퍼프리젠테이션의 일관성을 검사하는 방법에 대하여 중점적으로 제시하였다.

4.1 하이퍼 프리젠테이션을 위한 하이퍼 링크 설정 방법

분산 멀티미디어 프리젠테이션에 하이퍼 링크 설정을 한다는 것은 내부적으로 hot-spot을 지정한다는 의미와 같다. 지금까지는 hot-spot이 설정된 이미지나 텍스트로서의 하이퍼 텍스트 개념으로 많이 쓰였었다. 그러나 비디오나 오디오와 같은 연속 미디어 상에서의 hot-spot을 설정하는 것은 정적인 미디어상에서 설정하는 것과 다른 방법을 요구한다. 그리고 여러 미디어가 동기화된 프리젠테이션상의 hot-spot설정은 또 다른 방법을 요구한다. 프리젠테이션 상에 hot-spot은 시간축 상에 지점(At)이나 기간(During) 등과 같은 조건 설정이 가능하고 연속 미디어 상에는 다음과 같은 다양한 조건을 설정함으로써 hot-spot을 잡는 것이 가능하다.

- 연속미디어의 실행 중, 특정 시점 또는 일정 시간 간격으로 hot-Spot 지정 [7]
- 오디오에서의 hot-spot으로 소리의 시작과 끝, 남성, 여성 등의 특정 소리
- 비디오에서의 hot-spot으로 특정 물체의 등장과 사라짐, 움직임 감지, 특정색 감지
- 멀티미디어상의 hot-spot으로 비디오의 특정색과 동시에 나오는 폭발음

위와 같은 하이퍼링크를 설정할때 핵심이 되는 가정과 요구사항은 압축된 연속 미디어 상에서 실시간으로 하이퍼 링크의 조건 검사가 가능해야 한다는 점이다. 즉 일반적으로 하이퍼프리젠테이션 연출에 사용되는 오디오는 ADPCM(Adaptive DPCM)과 같은 방식으로 압축되어 있고 비디오는 H.261이나 H.263, MPEG, MJPEG 등으로 압축되어 있다

현재까지 연구된 방법들은 압축되어 있는 연속 미디어를 복원하여 히스토그램과 같은 여러 가지 음성이나 이미지 처리기법을 사용하였다. 이러한 방법은 복원하기 위하여 역DCT(Inverse DCT)를 계산해야하고 또 복원된 데이터 량이 너무 방대하기 때문에 실시간으로 하이퍼 링크의 조건을 검사한다는 것은 불가능하였다. 본 논문에서 제안하고자 하는 방법의 기본 아이디어는 복원하지 않고 조건을 검사하는 것이다. 우선 이것의 타당성을 확인하기 위하여 JPEG이나 MPEG, H.261등의 압축방법에 공통으로 사용되는 DCT변환을 거친 값들의 관계를 알아보고 다음과 같은 사실을 발견하였다.

[정리4.1] 하이퍼 링크의 조건을 검사하기 위하여 DCT를 이용하여 압축된 이미지 데이터의 DC값만 가지고 검사해도 충분하다.

우선 하이퍼 링크의 조건을 검사하기 위하여 필요한 특성은 이미지의 내용자체를 상세히 분석할 필요는 없다는 점이다. 즉 이미지내의 객체를 인식할 필요 없이 이미지 흐름의 변화만 가지고도 조건의 검사가 가능하다는 점이다.

그리고 압축된 데이터의 성질을 살펴보면 8x8 화소 블록에 대한 DCT로 변환된 값 중 DC성분은 블록내의 직류성분으로 8x8의 화소정보의 평균치이다. 따라서 한 개 이미지 전체를 해당되는 DC들을 한 점으로 보아 분석해도 하이퍼 링크의 조건을 검사하기에 충분한 정보를 가지고 있다.

위 사실을 실험적으로 분석하기 위하여 다음 방법을 사용하여 적용해 보았다.

[방법4.1] DCT 압축영상의 DC값 스케일링 알고리즘

1. 압축된 이미지에서 DC 값만 추출한다.
2. DC값을 회색계열(256단계 중 14개의 등급으로 나눔)에 대응시킨다

3. 그 값을 화면에 출력한다.

위 방식을 적용하여 화면에 출력한 3가지의 그림은 그림4와 같다.

그림 4에서 보는 바와 같이 오른쪽 DC값을 가지고도 이미지의 변화에 관한 분석을 하면 하이퍼 링크의 조건을 검사할 수 있음을 알 수 있다. 즉 압축된 각 영상의 DC값만 가지고도 첫 번째 두 개의 그림은 같은 영상의 흐름이라고 이해할 수 있고 마지막 3번째 그림은 전혀 다른 그림이라는 것을 판단할 수 있다. 실제 DC값만 가지고 패턴의 변화를 비교하는 방식은 히스토그램과 같이 이미 많이 연구되어 있다. 여기서 제안한 방식은 이미 패턴인식이나 연상처리분야에서 연구가 된 영상처리 방법을 원래의 복원된 영상이 아니라 압축된 데이터에 적용시킨다는 것이다. 그리고 이 방식은 원래의 이미지가 아니라 압축된 이미지이며 압축된 64개의 계수중 DC계수 1개만 사용하면 되기 때문에 조건 분석에 사용되는 데이터의 량을 획기적으로 줄일 수 있고 역DC계산을 하지 않아도 되기 때문에 실시간으로 분석이 가능하다. 이 방법의 단점은 원래의 이미지를 사용하지 않기 때문에 이미지상의 객체를 인식하는 정도의 내용자체의 분석은 곤란하다는 점이다.

[방법4.2] 영상데이터 스트림의 하이퍼 링크 조건 검사 방법

입력:

DCT방식을 이용하여 압축된 JPEG, H.261, MPEG등의 영상스트림  $\{I_i | i=1,2,3,\dots\}$ .

하이퍼링크의 조건집합  $COND = \{cond_i | 0 \leq i \leq n\}$

1. 영상 스트림  $I_i$ 와  $I_j$ 에서 DC값으로만 이루어진  $I'_i$ 와  $I'_j$ 를 구성
2. 히스토그램법에 의한 변화량  $M_{ij}$ 계산
3. 모든 조건에 대하여  $M_{ij}$ 가 그 조건의 변화량의 임계치  $T_{ij}(cond_k)$ 보다 크지 검사하여 만족되는 조건  $cond_k$ 검출
4. 검출된 조건  $cond_k$ 를 포함하고 있는 하이퍼 링크가 수행될 수 있도록 연락

4.2 하이퍼 프리젠테이션의 연출가능성 검사 방법

하이퍼 프리젠테이션을 명시하는 것은 사용자의 의지를 표명한 것으로 실제 연출될 수 있으나 하는 것은

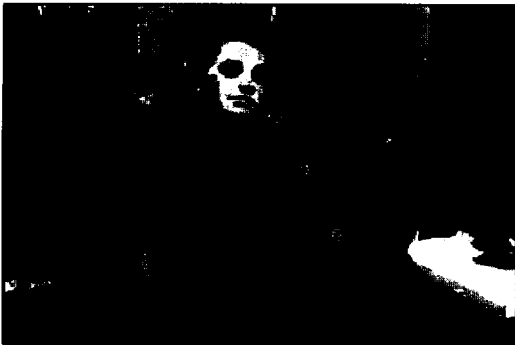


또 다른 문제이다. 즉 사용자가 원하더라도 연출되지 않을 수 있는 상황이 크게

- 하이퍼 프리젠테이션의 명시 프로그램 자체의 잘 못인 경우
- 시스템이나 네트워크의 구성요소의 성능이 사용자

QoS를 만족하지 못하는 경우

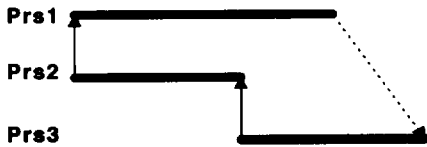
등 두 가지로 대별된다. 본 논문에서는 우선 첫 번째 문제를 해결하기 위하여 명시 프로그램에서 흔히 발생할 수 있는 경우를 살펴보고 그 문제를 해결하는 방법을 제시하였다.



(그림 4) DCT에서 DC 성분만을 추출, 비교  
(Fig. 4) A comparison of images based on DC values

5.2.1 사이클 발생의 문제점

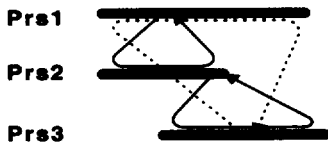
1) 큐 연결에서의 사이클이 발생하는 문제



(그림 5) 프리젠테이션 간의 큐의 연결  
(Fig. 5) Presentations connected by cue

위의 그림과 같이 prs3을 prs1에 finish속성으로 연결을 하려할 때 prs1은 prs3의 기준이 되며 따라서 prs1의 끝나는 시간에 맞춰 prs3이 끝나기 위해서 prs3의 시작 시간과 끝나는 시간을 변경해야 한다. 이때 prs3에 의존하는 prs2와 prs2에 의존하는 prs1에도 영향을 준다. 여기서 큐에 사이클이 생김을 알 수 있다. 이처럼 큐에 사이클이 생기거나 큐의 상대 프리젠테이션이 기준 프리젠테이션에 맞게 변환될 때 다른 큐에 영향을 받는 경우는 큐를 설정할 수 없도록 제한을 두어야 한다. 이를 위해 현재 연결하려는 기준 프리젠테이션이 상대 프리젠테이션과 한 그래프가 되면 사이클이 생김을 사용자에게 알리고 현재 연결하려는 큐의 상대 프리젠테이션이 또 다른 기준 프리젠테이션에 설정되어 있다면 다른 큐에 영향받음을 결정하도록 해야한다.

2) 하이퍼 링크 연결에서의 사이클 발생 문제



(그림 6) 프리젠테이션 간의 하이퍼 링크 설정  
(Fig. 6) Presentations connected by links

위와 같이 prs1에 prs2가 prs2에 prs3가 링크되어 있을 때 다시 prs3에 prs1을 점선으로 표시된 것과 같이 링크시키려 한다면 prs1->prs2->prs3을 수행하고 prs3에 prs1이 링크되어 있으면 링크 시점으로 되 돌아오지 못하고 다시 prs1->prs2->prs3을 또 prs1->

prs2->prs3을 반복하여 무한 loop에 빠지게 되는 결과를 가져온다. 이를 위해 자신의 링크를 직접 혹은 간접적이라도 가지고 있는 프리젠테이션을 링크할 수 없도록 제한을 두도록 알고리즘을 구현하였다.

5.2.2 하이퍼 프리젠테이션 명시프로그램의 일관성 검사방법

앞절에서 본바와 같은 사이클이 한 개 이상 존재하면 무한 루프에 빠져서 실제 연출을 할 수 없게되는 상황이 발생한다. 따라서 아무리 복잡한 명시 프로그램이라 할지라도 먼저 단순 사이클이나 복잡한 사이클을 발견하여 그것을 끊어주도록 해야한다. 본 논문에서 중요하게 생각하는 것은 우선 사이클을 찾는 문제라고 보고 그것을 해결하는 방법을 제안하였다.

[정의4.1] 하이퍼프리젠테이션 큐 연결 그래프

다음 과정을 반복하여 구성된 아래 방향성그래프

$$HPQG = (V, E)$$

를 **하이퍼프리젠테이션 큐 연결 그래프**라고 한다.

1. V는 하이퍼프리젠테이션을 구성하는 프리젠테이션들의 집합  $\{prs_i | 0 \leq i \leq n\}$ 이고,
2. 방향성 에지  $\langle prs_i, prs_j \rangle$ 가 에지의 집합 E에 속한다는 것은  $prs_i$ 가 기준 프리젠테이션으로  $prs_j$ 와 큐에 의해서 연결되어 있다는 것을 의미한다.

[정의4.2] 하이퍼프리젠테이션 링크 그래프

다음 과정을 반복하여 구성된 방향성 그래프

$$HPLG = (V, E)$$

를 **하이퍼 프리젠테이션 링크 그래프**라고 한다.

1. V는 하이퍼프리젠테이션을 구성하는 프리젠테이션들의 집합  $\{prs_i | 0 \leq i \leq n\}$ 이고,
2. 방향성 에지  $\langle prs_i, prs_j \rangle$ 가 에지의 집합 E에 속한다는 것은  $prs_i$ 상에 설정된 조건에 의해 프리젠테이션  $prs_j$ 가 연출되도록 명시되어 있다는 것을 의미한다.

하이퍼 프리젠테이션 그래프의 성질을 분석해 보면 다음과 같은 정리를 발견할 수 있다.

[정리4.2] 하이퍼 프리젠테이션 명시 프로그램에 사이클이 존재하면 하이퍼그래프에 그와 대응하는 사이클이 존재한다.

위 정리는 단순하지만 사이클을 찾는데 그래프이론

에서 개발되어 있는 사이클 찾는 방법을 활용할 수 있는 길을 열었다는 점에서 의미가 있다. 위의 정리를 근거로 하여 하이퍼 프리젠테이션 명시 프로그램의 일치성 검사하는 방법을 다음과 같이 제안하였다.

[방법4.3] 하이퍼 프리젠테이션 명시 프로그램의 일치성 검사 방법

입력: 하이퍼 프리젠테이션 명시 프로그램

1. 주어진 명시프로그램에 대응하는 하이퍼 프리젠테이션 큐 그래프 HPQG와 링크 그래프 HPLG를 구성한다.
2. 그래프이론의 사이클 찾는 방법을 이용하여 HPQG와 HPLG 내에 존재하는 사이클을 찾는다.

5. 구현 및 실험

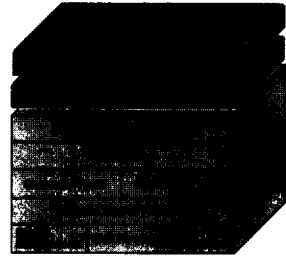
5.1 하이퍼 프리젠테이션 구현

기존의 MuX Server는 스트림 계층의 시간적 동기화를 통한 프리젠테이션 계층까지를 지원하고 있었다 [1][2][3]. 이제는 다양한 동기화 명시방법을 통해 프리젠테이션들을 제어함으로써 하이퍼 프리젠테이션의 표현이 가능해 졌다.

기존의 MuX서버는 프리젠테이션 계층이 구현되어 있는데 그위에 하이퍼 프리젠테이션을 새롭게 구현하였다. 그 계층에는 하이퍼 프리젠테이션을 명시하는데 필

MuX Client

MuX Server



(그림 6) MuX Server에서의 하이퍼 프리젠테이션 계층 (Fig. 6) The hyperpresentation layer in the MUX model

요한 메카니즘으로 큐나 하이퍼 링크 등을 제공하도록 했다. 그리고 그기능이 MuX Lib에 API로 제공되도록 확장하였다.

5.2 API(Application Programming Interface) 목록

앞에서 설명했던 하이퍼 프리젠테이션에서의 동기화 기법을 지원하기 위해 제공되는 API 목록이 <표 2>로 나와 있다.

5.3 하이퍼 프리젠테이션 실험

5.3.1 시나리오

<prs1>은 뉴스 앵커의 시작 멘트와 배경 음악으로 구성된 프리젠테이션이며, <prs2>는 의원 선거에 대한 보도이다. <prs3>과 <prs5>는 두 명의 의원후보 A와 B의 연설 장면을 담고 있다. 사용자는 동시에 화면으로 나타나는 두 연설 장면을 보며 그 중 하나를 선택하여 해당 후보의 자세한 정보를 얻을 수 있다. <prs4>와 <prs6>이 각각 후보 A와 B에 대한 자세한 소개를 담

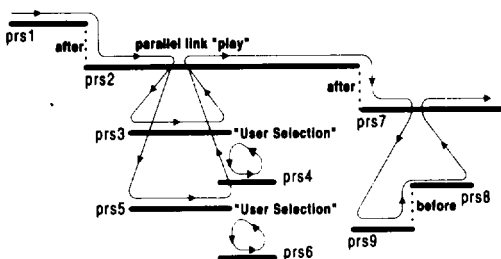
<표 2> 하이퍼 프리젠테이션의 API 목록  
<Table 2> A list of the hyperpresentation API

기능	A P I
하이퍼 프리젠테이션 객체의 생성과 소멸	HyperPresentation ::HyperPresentation(Core*) ::~~HyperPresentation(void)
프리젠테이션 객체의 접속	HyperPresentation ::AddPresentation(Presentation*)
하이퍼 링크 객체의 접속	HyperPresentation ::AddHyperLink(HyperLink*) ::DeleteHyperLink(HyperLink*)
큐 객체의 접속	HyperPresentation ::AddCue(Cue*) ::DeleteCue(Cue*)

하이퍼 프리젠테이션 객체의 제어	HyperPresentation ::GetCueType(Presentation*) ::GetCue(Presentation*) ::SetStartTime(LTime) ::SetEndTime(LTime) ::ReqCurrentTick(void) ::Play(void) ::Stop(void) ::Pause(void) ::Resume(void) ::Play(Presentation*) ::Stop(Presentation*) ::Pause(Presentation*) ::Resume(Presentation*) ::Back(Presentation*) ::Skip(Presentation*) ::ReturnToMain(Presentation*) ::Replay(Presentation*)
하이퍼 링크 객체의 생성과 소멸	HyperLink ::HyperLink(Core*, Presentation*, LTime, LTime) ::HyperLink(Core*, Presentation*, LTime, LTime, HyperLink_Type) ::~HyperLink()
하이퍼 링크 객체의 제어	HyperLink ::SetConditionandAction(ConditionType, HAction, Presentation*) ::SetConditionandAction(ConditionType, HAction, char*)
큐 객체의 생성과 소멸	Cue ::Cue(Core*, Presentation*, LTime, Cue_Type) ::~Cue()
큐 객체의 제어	Cue::SetCondion(ConditionType, Presentation*)

은 파일이다. 의원 선거에 대한 기사가 끝나면 계속해서 내일의 날씨를 진행한다. <prs7>은 기상 캐스터의 진행 멘트와 배경 음악, 그리고 화면으로 구성된 프리젠테이션이다. <prs8>은 각 지역별 날씨, <prs9>는 해상의 날씨로 구성되어 있으며, 이러한 프리젠테이션들을 하이퍼 링크와 큐로 연결하여 표현한 하이퍼 프리젠테이션이 그림 7에 나와 있다.

위의 시나리오를 구현하는 방법에는 두 가지가 있다.

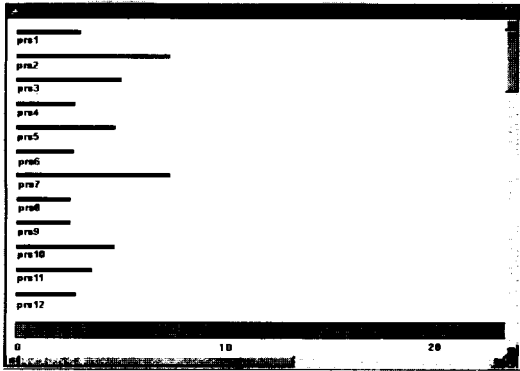


(그림 7) 뉴스진행 시나리오  
(Fig. 7) News scenario

첫 번째는 하이퍼 프리젠테이션을 지원하는 MuX의 API를 이용하여 구현하는 것이고 두 번째는 본 논문에서 제시하고자 하는 VIP(Visual Interface Player)를 이용하는 것이다. 먼저 API를 이용하는 경우를 먼저 살펴보자. API를 이용해 프로그래밍 하는 경우에 프로그래머는 일단 프로그래밍 언어(C/C++)와 API를 사용할 줄 알아야 한다. 또한 하이퍼 프리젠테이션을 이루는 각 객체에 대해 일관성을 유지하기 위해 많은 신경을 써야 한다. 이렇게 하여 만들어진 결과를 확인하기 위해서는 컴파일의 과정을 거쳐 실행파일로 만든 후 실행해야 한다. 이 같은 여러 불편한 점들을 개선하여 하이퍼 프리젠테이션을 더욱 편리하게 작성하는 방법이 바로 VIP를 이용하는 것이다. VIP에서는 각 객체를 나타내는 아이콘들이 있는데 프로그램은 단지 이들을 시나리오에 맞게 연결하고 시간 설정 등과 같은 몇 가지 추가적인 설정만 해주면 된다. 아이콘을 이용하므로 일관성 유지가 매우 쉽고 프로그래밍 언어를 모르는 사람도 구현 할 수 있는 등의 많은 장점이 있다.

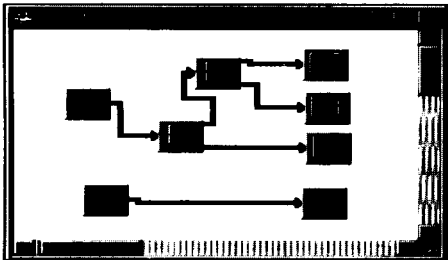
5.3.2 VIP를 이용한 뉴스 시나리오 프로그램

시나리오 설정을 위해 VIP의 초기 화면에서 하이퍼 프리젠테이션 창을 생성한 후, 툴바의 [Register Presentation] 버튼을 눌러 필요한 프리젠테이션들을 모두 하이퍼 프리젠테이션 창에 등록하면 다음과 같이 하이퍼 프리젠테이션 창에 등록된 프리젠테이션들의 이름과 라인을 볼 수 있다.



(그림 8) 하이퍼 프리젠테이션 창  
(Fig. 8) The window of a hyperpresentation

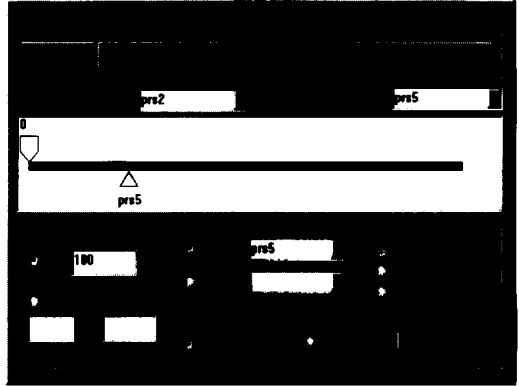
위 창에 막대모양으로 보이는 프리젠테이션은 실제 아래 그림에서 보는바와 같은 스트림 데이터들을 동기화 시켜 놓은 것이다.



(그림 9)아이콘 프로그램  
(Fig. 9) An iconic program

하이퍼프리젠테이션 창에서 큐와 하이퍼 링크를 설정할 수 있다. 전체 시나리오의 흐름을 고려하여 일단 아나운서의 멘트로 이루어진 <prs1>, <prs2>, <prs7>, <prs10>을 차례로 "before" 큐로 연결하고 <prs11>와 <prs12>를 "after" 큐로 연결한다. 큐의 연결은, 툴바의 connect 버튼으로 원하는 프리젠테이션들을 연결할

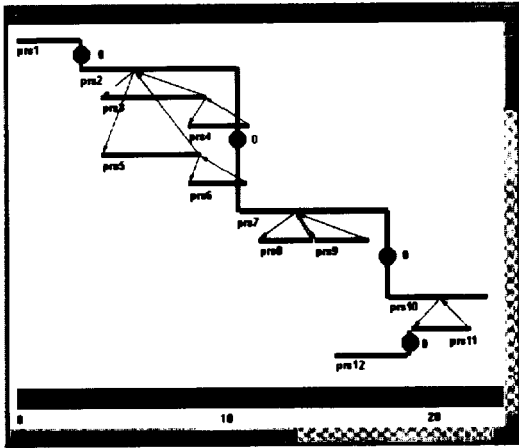
수도 있고, 툴바의 큐 버튼을 눌러 큐 설정을 할 수도 있다. 이제 하이퍼 링크를 설정해 보자. 툴바의 하이퍼 링크 버튼을 클릭하면 다음과 같은 링크 설정 창이 생성된다.



(그림 10) 하이퍼 링크 설정 창  
(Fig. 10) The window of hyperlink setup

우선 <prs2>에 <prs3>과 <prs5>를 연결시켜 보자. 상단의 Current Presentation 콤보 박스에서 <prs2>를 선택하고, 하이퍼 링크의 Location을 지정하기 위해서 Play 버튼을 눌러 현재의 <prs2>를 실행시킨 후 원하는 위치에서 New HyperLink 버튼을 눌러 위치를 설정할 수 있다. 또한 설정된 하이퍼 링크의 위치를 수정하고 싶다면, Location 그룹 상자에서 LTS값을 직접 입력해도 좋다. 이 때, 설정된 시점에서 하이퍼 링크를 실행해야 하므로 LTS값은 at에 입력한다. 그러면 현재 선택된 <prs2>의 라인에 삼각형 모양의 하이퍼 링크 포인터가 생성된 것을 볼 수 있다. 그리고 Action 그룹 상자의 Play 콤보 박스에서 <prs3>을 선택하여 이 하이퍼 링크의 액션을 설정하고, Condition 그룹 상자에서 Active를 선택하여 컨디션을 설정한다.

다음은 시나리오에 따라 <prs3>과 같은 위치에 <prs5>를 연결시켜 보자. 위의 방식으로 <prs5>를 연결시키면 하단의 Multi-Link 그룹 상자가 활성화된다. <prs3>과 <prs5>는 동시에 실행되어야 하므로 Parallel을 선택하고, 컨디션은 Active로 설정한다. Current Presentation을 바꿔가며 이와 동일한 방식으로 시나리오에 맞게 각 프리젠테이션마다 하이퍼 링크를 설정하면 된다. 다음 그림 11은 지금까지 설정된 큐와 링크를 한눈에 보여 준다.



(그림 11) 하이퍼 링크와 큐 설정 후  
(Fig. 11) After setup of hyper links and cues

다시 툴바의 하이퍼 링크 버튼을 클릭하면 설정된 하이퍼 링크를 수정할 수 있다. 모든 설정을 끝낸 후 하이퍼 프리젠테이션을 실행시키면 User Interaction 창이 생성되고 이를 통해 프리젠테이션들을 제어할 수 있게 된다.

5.3.3 API를 이용한 시나리오 구현

```
Core *core:

/* 프리젠테이션 9개는 이미 작성되어 있다고 가정 */
/* (Prs1, Prs2, ..., Prs9) */

HyperPresentation *HP = new
HyperPresentation(core);

//작성된 프리젠테이션들을 하이퍼 프리젠테이션에 추가한다
HP->AddPresentation(Prs1);
HP->AddPresentation(Prs2);
//다른 프리젠테이션들도 마찬가지로 하이퍼 프리젠테이션에
추가한다

//Prs1과 Prs2의 큐 생성
Cue *cue1 = new Cue(core, Prs2, 0, AFTER);
cue1->SetCondition(ACTIVE, Prs1);
HP->AddCue(cue1);
//링크의 속성이 parallel일 때
HyperLink *Hlink1 = new HyperLink(core, Prs2,
100, 100, PLINK);
Hlink1->SetConditionandAction(ACTIVE, PLAY,
Prs3);
HP->AddHyperLink(Hlink1);
```

```
//User interaction
HyperLink *Hlink2 = new HyperLink(core, Prs3, 30,
100);
Hlink2->SetConditionandAction(USER_SELECTION,
PLAY, Prs4);
HP->AddHyperLink(Hlink2);

//Prs5와 Prs6을 의한 하이퍼 링크의 설정은 위와 동일하다

//Prs2와 Prs7의 큐 설정
Cue *cue2 = new Cue(core, Prs7, 0, AFTER);
cue2->SetCondition(ACTIVE, Prs2);
HP->AddCue(cue2);

//Prs8와 Prs9의 하이퍼 링크와 큐도 동일한 방법으로
설정한다

//하이퍼프리젠테이션을 실행
HP->Play();
```

(그림 12) API로 작성한 예제 프로그램  
(Fig. 12) An example of API program

6. 결 론

이 논문에서는 Global-Time, 큐, 하이퍼 링크를 이  
용한 동기화 기법으로 하이퍼 프리젠테이션의 이론을  
정립하였다. 멀티미디어의 연출을 순차적으로만 하는  
것이 아니라 링크개념을 설정하여 특정지점에서 원하는  
다음 정보를 보기 위하여 비순차적으로 연출할 수 있도  
록 하고 그 하위수준의 연출이 완료되면 다시 원래 프  
리젠테이션으로 되돌아 계속 연출하는 것이 하이퍼프리  
젠테이션의 개념이다.

본 논문에서는 우선 기존의 동기화 명시방법을 살펴  
보고 하이퍼프리젠테이션을 위한 새로운 동기화 명시방  
법을 제안하였다. 그리고 하이퍼 프리젠테이션을 위한  
하이퍼 링크 설정 방법에 대하여 살펴보았다. 하이퍼링  
크 설정 관련 가장 핵심이 되는 가정과 요구사항은 압  
축된 연속 미디어 상에서 실시간으로 하이퍼 링크의 조  
건 검사가 가능해야 한다는 점이다. 본 논문에서는  
DCT변환방식을 이용하여 압축된 미디어에 하이퍼 링  
크를 실시간으로 설정하는 방법을 제안하였다. 또한 위  
에서 제시된 명시방법을 이용하여 나타낸 하이퍼 프리  
젠테이션 프로그램이 연출 가능한지 미리 검사할 수 있  
는 방법도 제안하였고 마지막으로 지금까지 제안된 방  
법을 MuX(Multimedia Input/Output Server)에

구현하여 실제 연출될 수 있는 시나리오를 대상으로 실험하였다.

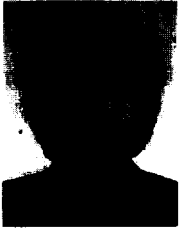
본 논문에서는 우리가 일상 생활에서 많이 경험하였고, 필요성이 있는지는 알지만, 그것이 하이퍼 프리젠테이션 또는 하이퍼 미디어의 개념인지 알지 못하는 상황을 여러 가지 멀티미디어 동기화 기법을 사용하여 프로그램으로 나타내는데 중점을 두었다. 이론적인 배경을 제공하는 것도 중요한 의미를 가지고 있지만 실제 구현이 가능한 방법을 제안할 수 있었다는데 더 큰 의의가 있다. 앞으로는 이 하이퍼 프리젠테이션 개념을 바탕으로 다음과 같은 문제점을 해결하려 한다.

- 하이퍼 프리젠테이션의 QoS 설정 및 기술 방법
- 분산 하이퍼 프리젠테이션의 연출 성능 계산 방법 연구
- Critical path의 QoS를 만족하기 위한 최소 버퍼 할당 방식 연구
- 분산 하이퍼 프리젠테이션 상 독립적인 연출 요소 그룹화 방법 연구
- 미리 가져와야 할 멀티미디어 자료량의 결정
- 독립적인 연출 그룹의 스케줄링 방식 연구

### 참 고 문 헌

- [1] 임영환, "ComBiStation : 분산 멀티미디어 컴퓨팅 환경을 위한 컴퓨터 플랫폼", 정보과학회 논문지, 제2권, 제1호.
- [2] Baker R. A. Downing, K.Finn, E.Rennison, D.H.Kim, and Y.H. Lim, "Multimedia Processing Model for a Distributed Multimedia I/O System".
- [3] Rennison, E. R.Baker, D.H.Kim, and Y.H. Lim, "MuX : An X Co-Existent Time-Based Multimedia I/O Server.", The X Resource, Issue 1, 1992, pp.213-233.
- [4] Ralf Steinmetz, "Synchronization Properties in Multimedia System" COMM.IEEE, Vol.8, No.3, pp.403-405.
- [5] David P. Anderson, George Homsy, "A Continuous Media I/O Server and Its Synchronization Mechanism", COMPUTER, ACM pp.53-57, October, 1991.
- [6] Mitsutoshi Iino, Young Francis Day, Arif Ghafoor, An Object-Oriented Model for Spatio-Temporal Synchronization of Multimedia Information, Multimedia Systems, IEEE, 1997, pp.111-112.
- [7] Maria Jose Perez-Luque, Thomas D. C. Little, A Temporal Reference Framework for Multimedia Synchronization, IEEE Journal on Selected Areas in Communications, Vol. 14, No.1, pp.38-43, January 1996.
- [8] Earl Rennison, Kate Finn, Doohyun-Kim and Younghwan Lim, Multidimensional Hyper Presentation in a Distributed Multimedia System, MCAT 93 Conference.
- [9] T. Wahl and K. Rotheprmel, "Representing Time in Multimedia Systems," Proceedings of International Conference on Multimedia Computing and Systems, May 1994.
- [10] International Standards Organization, Hypermedia/Time-based Document Structuring Language(HyTime), ISO/IEC, 1992.
- [11] M. Salmony and D. Shepherd, Extending OSI to Support Synchronization Required by Multimedia Applications, Computer Communications, 13:399-406, September 1990.
- [12] W. Appelt, HyperODA, ISO/IEC/JTC1/SC18/WG3, 1989.
- [13] F. Halasz and M. Schwartz, "The Dexter Hypertext: reference model," CACM, 37(2): 30-39, Feb. 1994.
- [14] L. Hardman, D. Bulterman and G. Rossum, "The Amsterdam Hypermedia Model: extending hypertext to support real multimedia," Technical Report CS-R9306, Centrum voor Wiskunde en Informatica, 1993.
- [15] C.W.Mercer and H. Tokuda, "The ARTS Real-Time Object Model," IEEE Real-Time System Symposium, pp.2-10, 1990.
- [16] K. Jeffay, D.L.Stone and D.E.Poirier, "YARTOS : Kernel Support for Efficient, Predictable Real-Time Systems," Proceedings of IFAC, Workshop on Real-Time Programming, May 1991.

[17] D.P.Anderson, " Meta-scheduling for distributed continuous media." ACM Transaction on Computing Systems, 11(3). August 1993.



**임영환**

1977년 경북대학교 수학과 졸업 (학사)  
1979년 한국과학원 전산학과 졸업 (석사)  
1985년 Northwestern University 전산학과 졸업(박사)

1979년~1996년 한국전자통신연구원 책임연구원  
1996년~현재 숭실대학교 부교수  
관심분야: 멀티미디어, 초고속정보통신 미들웨어, 멀티미디어 운영체제 등



**공상환**

1977년 숭실대학교 전산학과 졸업 (학사)  
1983년 고려대학교 전자정보처리학과(석사)  
1998년 충북대학교 전산학과(박사)  
1977년~1981년 육군제2군수지원

사령부 전산장교  
1981년~현재 한국전자통신연구원 책임연구원  
1996년~1997년 스텐포드연구소 객원연구원  
관심분야: 멀티미디어, 분산시스템, 객체지향시스템



**김두현**

1985년 서울대학교 컴퓨터공학과 졸업(학사)  
1987년 한국과학기술원 전산학과 졸업(석사)  
1987년~현재 한국전자통신연구원 책임연구원

1991년~1993년 스텐포드연구소 객원연구원  
1993년 정보처리기술사 취득  
관심분야: 분산멀티미디어 처리, 웹 멀티미디어 프리젠테이션 멀티미디어 운영체제, 분산멀티미디어 성능분석