

# 동기화된 멀티미디어 프레젠테이션을 위한 XHTML 확장

이 수 철<sup>†</sup> · 황 인 준<sup>††</sup>

## 요 약

멀티미디어 프레젠테이션은 강의나 업무 보고, 화상 회의 등 여러 분야에서 활용되고 있다. 이러한 프레젠테이션은 대개 시간적인 구조를 가진 멀티미디어 객체들로 구성되어 있다. 여기서 시간적인 구조는 프레젠테이션을 구성하는 객체들이 전달되어야 하는 시점과 재생 시간 그리고 이러한 객체들 사이에 존재하는 시간적 관계 등을 명시한다. 본 논문에서는 XHTML의 시간적 확장을 통해 웹 문서와 동기화된 멀티미디어 데이터의 통합을 가능하게 하는 방법을 제안한다. 시간적 확장을 위해 (i) 시간적인 구성(temporal composition)을 위한 XHTML+TIME (Xtime) link을 정의하고, (ii) 상호 공통된 시간을 바탕으로 한 미디어 객체간의 긴밀한 동기화 표현을 지원하며, (iii) 프레젠테이션 스크린의 범위를 명시하는 동적인 레이아웃(layout)을 가능하게 하였다. 또한 멀티미디어 프레젠테이션에서 위와 같은 개념들을 지원하기 위한 연성 있는 실행 구조를 보인다.

## Extending XHTML for Synchronized Multimedia Presentation

Soo Cheol Lee<sup>†</sup> · Een Jun Hwang<sup>††</sup>

## ABSTRACT

Multimedia presentations are a class of documents that are used for lectures, tour guides, business presentations, and in many other applications. These presentations comprise multimedia objects with a temporal structure that specifies when objects are to be delivered, for what duration, and what the temporal relationships among objects are. In this paper, we propose temporal extensions to XHTML that allow seamless integration of synchronized multimedia into web documents. The extensions are based on three concepts: (i) XHTML+TIME (in short, Xtime) links for temporal composition, (ii) common time bases for close synchronization between media objects, and (iii) dynamic layout for specifying regions of a screen for presentation. Also, we will show a flexible execution architecture to support these concepts in a multimedia presentation.

**키워드:** 멀티미디어 프레젠테이션(Multimedia Presentations), 멀티미디어 동기화(Multimedia Synchronization), XHTML

### 1. 서 론

멀티미디어 기술이 발달하고 인터넷 사용이 활성화됨에 따라 전자 상거래, 웹 출판, 전자 도서관 등의 다양한 디지털 서비스가 등장하고 있다. 이러한 응용 서비스에서 텍스트나 이미지, 오디오 및 비디오와 같은 멀티미디어 데이터에 대한 요구가 증가하고 있다. 멀티미디어 데이터를 효과적으로 사용하기 위해서는 여러 가지 사항이 고려되어야 하는데 그 중의 하나가 비디오, 오디오, 텍스트 등으로 이루어진 멀티미디어 객체간의 동기화(synchronization)이다. 동기화란 원래 사건들의 시간적 순서를 지키도록 하는 것으로 운영체제나 병렬 프로그래밍 언어에서 많이 연구되어 온 주제이지만, 멀티

미디어에서의 동기화란 미디어 객체들의 상영 에 있어서 미디어들간의, 혹은 시간에 의존적인 미디어 내의 시간적 사건의 동기화와 공간적인 요소를 나타낼 수 있는 공간적 사건의 동기화를 의미한다.

WWW 컨소시엄(W3C)에서는 최근에 멀티미디어 문서의 동기화를 위해 SMIL이라는 마크업 언어를 표준으로 제정하였다. SMIL은 XML을 기반으로 하여 XML에 시간적인 속성과 관련된 몇 가지 기능이 추가된 것을 말하며 이미지나 비디오, 텍스트와 같은 멀티미디어 객체가 삽입되어 있는 문서를 다루기 위해서 나온 마크업 언어이다. 그러나 SMIL은 다음과 같은 몇 가지 단점을 가지고 있다. 첫째, SMIL의 시간적 속성은 두 가지 다른 개념(시간 간격과 타임 포인트)이 혼합된 형태인 복합적 방법을 기반으로 하고 있다. 예를 들어, SMIL에서 사용되는 요소(element)인 *par*와 *seq*는 트리에서 노드의 형태로 표현이 되고, 멀티미디어 객체들은 각 노드의 리프 노드에 위치하게 된다. *par* 요소만을 이용해서 멀

\* 본 논문은 2001년도 두뇌한국 21사업(BK21), 정보통신부 주관 2001년도 대학기초 연구지원 및 아주대학교 정학 연구비 지원에 의한 것이다.

† 정 회 원 : 아주대학교 정보통신 전문대학원

†† 총신회원 : 아주대학교 정보통신 전문대학원 교수

논문접수 : 2001년 7월 23일, 심사완료 : 2001년 10월 8일

멀티미디어 객체의 동기화를 표현할 수 없기 때문에 SMIL에서는 *switch*, *anchor*와 같은 부가적인 속성(attribute)을 사용한다. 이러한 부가적 속성들 때문에 멀티미디어 객체를 표현할 수 있는 방법이 통일되어 있지 않고, 명확하게 규정되어 있지 않기 때문에 혼란을 유발할 수 있다. 둘째, SMIL은 *par* 요소와 밀접한 관계가 있는 여러 요소를 지원하기 위해 *lipsync* 라는 추가적인 속성을 제공하고 있다. 그러나 이것은 멀티미디어 객체들간에 상호 동기화를 표현하기에는 충분하지 않고, 더욱이 속성의 정확한 동작이 정의되어 있지 않아서 구현 시에 어려움이 따른다. 셋째, 문서의 레이아웃이 *tuner* 요소를 기반으로 하고 있고, 이것은 CSS positioning 버전을 단순화한 것이기 때문에 시간에 따라 레이아웃을 변경시킬 수 없는 정적인 속성을 가지고 있다. 시간 객체는 시간의 변화에 따라서 배치가 달라지기 때문에, 정적인 레이아웃은 이러한 융통성을 제한한다.

위와 같은 SMIL의 문제점을 해결하기 위해서, 본 논문에서는 다음의 세 가지 개념을 기반으로 XHTML의 시간적 확장을 제안하고 실행구조를 정의하였다. 첫째는, 시간적 구성을 위한 *Xtime link*이고 둘째는, 미디어 객체들간의 *lipsync* 동기화를 위한 *time bases*이며 셋째는, 미디어 객체의 확장을 위한 동적인 레이아웃이다. *Xtime link*는 현재 웹에서 많이 사용되고 있는 *hypertext link*와 유사하다. 그러나 이것은 *hypertext link*와는 달리 시간적 의미를 내포하고 있고, 비디오 프레임이나 연속된 시간선상에 할당되어 있는 샘플 데이터와 같은 두개의 미디어 샘플과 관련이 있다. 그리고 *Xtime link*는 연속된 시간선상에서 대상이 되는 미디어 샘플로의 이동이 가능하도록 *skipping instance*로 구성되어 있다.

<표 1>에서는 본 논문에서 제안하는 XHTML의 시간적 확장과 SMIL언어의 차이를 보여주고 있다.

<표 1> SMIL과 XHTML의 시간적 확장 비교

	SMIL	XHTML의 시간적 확장
동기화	<ul style="list-style-type: none"> <li>• <i>par</i> 요소와 <i>switch</i>, <i>anchor</i> 와 같은 부가적 속성사용</li> <li>• 사용하기 복잡하다</li> </ul>	<ul style="list-style-type: none"> <li>• <i>time bases</i> 속성 사용</li> <li>• 사용법이 통일되어 있음</li> <li>• 사용하기 쉬움</li> </ul>
레이아웃	<ul style="list-style-type: none"> <li>• 시간에 따라 변경 불가능</li> <li>• 시간의 변화에 따라 레이아웃 작성 필요</li> </ul>	<ul style="list-style-type: none"> <li>• 시간에 따라 변경 가능</li> <li>• 시간의 변화에 따라 새로운 레이아웃이 필요 없음</li> </ul>
구현	<ul style="list-style-type: none"> <li>• 새로운 형태의 브라우저가 필요</li> </ul>	<ul style="list-style-type: none"> <li>• 기존 브라우저(익스플로러 5) 사용가능</li> </ul>

본 논문의 구성은 다음과 같다. 먼저 2장에서는 현재 나와 있는 XML 언어와 이로부터 파생된 XHTML과 SMIL언어에 대해서 간략하게 소개를 한다. 3장에서는 XHTML의 시간적 확장에 대해서 설명하고, 4장에서는 실행구조의 구현에 관해 설명한다. 마지막으로 5장에서는 본 논문의 요약과 향후 연구 과제를 언급하며 결론을 내린다.

## 2. 관련 표준

### 2.1 XML

XML(eXtensible Markup Language)은 1996년 W3C에서 제안한 것으로써, 웹 상에서 구조화된 문서의 전송이 가능하도록 설계된 표준화된 텍스트 형식이다. 이는 인터넷에서 기존에 사용하던 HTML의 한계를 극복하고 SGML의 복잡함을 줄여주는 해결책으로 HTML에 사용자가 새로운 태그(tag)를 정의할 수 있는 기능이 추가되었다. XML 태그들은 정보가 어떻게 표시될 것인지를 정의하는 것이 아니라 그 정보가 어떤 정보인지를 정의한다. 그 정보를 어떻게 활용하는가는 전적으로 XML 문서를 사용하는 시스템에 달려있다. 시스템에 따라서는 서술적 정보를 하나의 표로 표시할 수도 있고, 계산에 활용할 수도 있으며, 정보를 데이터베이스에 기록할 수도 있다. 즉, XML 문서 작성자는 데이터 구조를 마크업하기만 하면 된다.

XML의 장점은 확장성이다. 간단히 말해서 누구라도 새로운 태그를 만드는 것이 가능하다. 그런데 누구나 자신만의 새로운 태그를 만들 수 있다면 브라우저 역시 그것의 의미를 알 수 있어야 한다. 그러나 새로운 태그들마다 개별적인 DTD(Data Type Definition)를 첨부하거나 새로운 태그들이 나타날 때마다 브라우저를 버전업해서 그 태그들에 대한 규칙들을 브라우저에 집어넣는 것은 실용적인 해결책이 못된다. 이 문제에 대한 해결책은 아주 간단하다. XML은 SGML보다 훨씬 더 엄격하며, 문서가 적격성(well-formedness)을 갖추기만 하면 사용자 에이전트는 DTD가 없이도 문서를 해석할 수 있다. 브라우저가 특정한 XML 규칙들을 인식하도록 설정되어 있다면 문서를 해석하는데 특별한 DTD가 필요 없는 것이다. 이것은 사용자 에이전트가 문서를 처리하는 데 필요한 부담을 크게 줄일 수 있게 한다.

### 2.2 XHTML

XHTML은 HTML의 어휘와 XML의 문법을 사용하는 하나의 언어이다. XHTML의 태그와 요소들은 HTML의 그것들과 동일하며, XML의 문법을 사용하기 때문에 모든 XML 사용자 에이전트들에 의해 표시 또는 해석될 수 있다. 간단히 말해서, XHTML은 XML과 HTML의 장점만을 결합한 언어라고 할 수 있다. XHTML은 이식성이 뛰어난 언어로서 현재 적격성을 갖추지 못한 HTML문서들은 브라우저의 종류에 따라, 즉 개별 브라우저가 HTML을 처리하는 특정한 방식에 따라 약간씩 다른 모습으로 표시된다. 하지만 XHTML의 경우에는 모든 브라우저와 사용자 에이전트들이 동일한 방식으로 문서를 해석한다. XHTML이 주목받는 또 다른 이유로는 다음을 들 수 있다. 첫째, XHTML이 XML과 HTML에 모두 호환이 된다. 둘째, XHTML 문서는 스크립트나 애플릿과 같은 애플리케이션과 같이 활용을 할 수 있다. 셋째, XHTML의

경우에는 새로운 이름 공간(namespace)을 정의함으로써 새로운 태그들을 추가할 수 있다. 이와 같은 장점때문에 XHTML은 HTML을 대체하는 웹 언어가 될 것이다.

### 2.3 SMIL

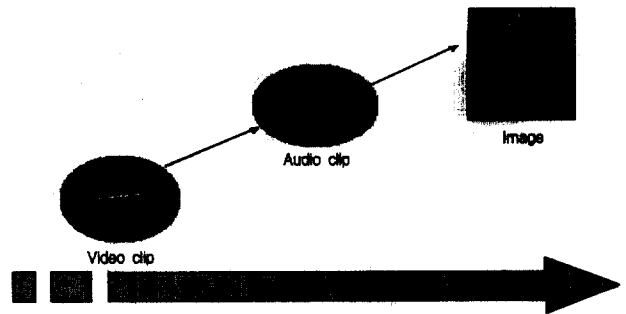
SMIL(Synchronized Multimedia Integration Language)은 W3C에 의해 개발되어 1998년 7월에 발표된 멀티미디어 데이터의 통합 표현을 위한 동기화 언어의 표준이다. 멀티미디어 프레젠테이션의 작성자는 SMIL을 사용하여 표현하고자 하는 미디어 요소들을 화면상에 위치시키고, 시간적인 동기화를 명시함으로써 하나의 멀티미디어 프레젠테이션으로 표현할 수 있으며, 사용자의 기호나 언어 등에 따른 미디어의 선택적 표현이 가능하다. SMIL에서의 미디어 공간 배치는 CSS와 흡사한 표기법을 사용하며 각 미디어의 표현 위치와 미디어간의 상하 관계 및 배경 색상 등을 지정함으로써 기존의 CSS 사용자들이 쉽게 사용할 수 있다. 또한 시간적인 동기화 정보의 표현은 *seq*와 *par* 블록의 사용을 통해서 간편한 표현기법을 사용하고 있다. SMIL 문서는 실제의 멀티미디어 콘텐츠를 설명하는 것이 아니라, 언제 어디서 재생될 것인지를 설명하는 것이다. SMIL 문서는 데이터의 타입 시퀀스를 지정하는 것뿐만 아니라 개별적인 화면에서 그래픽 개체의 위치를 지정할 수도 있고, 미디어 개체로의 링크도 첨부할 수 있다. 예를 들어, 동영상과 사운드가 재생되는 동시에 관련된 텍스트가 자막으로 나타나게 할 수도 있는 것이다. SMIL은 자신의 태그를 정의할 수 있는 언어이고 XML에서 파생되었기 때문에 XML 파서를 이용해서 오브젝트 트리를 생성할 수 있다. 오브젝트 트리는 모든 미디어 오브젝트의 시간 정보와 같은 것을 포함하고 있다. SMIL의 이벤트중 하나인 *playback* 이벤트는 *time line*에 따라서 스케줄링 되고, 미디어 객체는 병렬 혹은 순차적으로 보여줄 수 있다. SMIL은 다음과 같은 특징을 가지고 있다.

- 웹 저작자가 미디어 객체의 특성(location, size, zindex ordering)을 지정할 수 있고 이것에 시간적 속성(begin time, duration or, end time)을 부여할 수 있다.
- 컴퓨터 기반의 대화형 CD와 같은 대화형 멀티미디어 프레젠테이션을 만들 수 있는 기능을 제공한다.
- 미디어 객체들을 로컬이나 웹에 분산시킬 수가 있다.

### 3. XHTML의 시간적 확장

일반적인 멀티미디어 문서는 논리적 구조와 물리적 구조라는 두 가지 구조를 가지고 있다. 논리적 구조는 문서내의 서로 다른 컴포넌트들이 어떻게 연관되어 있는지를 정의한다. 이것은 문서의 페이지나 레이아웃과 같은 물리적인 미디어에 관련된 속성에서 스타일에 관련된 속성을 분리하고 수정할 수 있도록 한다. 물리적 구조는 문서에서 논리적 컴포넌트들이

물리적인 미디어에서 어떻게 표현되는지를 정의한다. 특히, 문서 내에서 논리적 컴포넌트들의 공간적 레이아웃을 표현한다. 대부분의 멀티미디어 문서 모델들은 공간적인 개념을 분리한 시간적 구조를 가지고 있지만, 기존에 있는 논리적 구조와 물리적 구조를 통합한다는 것은 쉬운 일이 아니다. 시간적 기술과 공간적 기술은 서로가 독립적이고 혼합될 수가 없다. 예를 들어, 시간의 변화에 따라서 레이아웃을 변화시킨다는 것은 불가능한 일이다. 그 결과, 시간적인 속성을 가지고 있는 문서일지라도 레이아웃은 정적일 수밖에 없다. (그림 1)은 시간에 따라서 레이아웃이 변화는 문서의 예를 도식화 한 것이다. 그림에서 비디오 클립이 끝났을 때, 다음의 나오는 오디오 클립을 표현하기 위해서 새로운 레이아웃이 사용된다. 오디오 클립 이후에, 세번째 레이아웃은 이미지를 어떻게 표현할 것인가를 정의한다. 문서의 컴포넌트들뿐만 아니라 레이아웃도 시간속성을 가지고 있다. 그렇기 때문에, 시간의 변화에 따라서 XHTML 문서에 시간적 속성을 통합하기 위해서는, 문서의 시간적 구성 엘리먼트를 명시하고 각 엘리먼트간의 긴밀한 동기화를 표현함으로써 유연한 구조를 가지게 해야 한다.

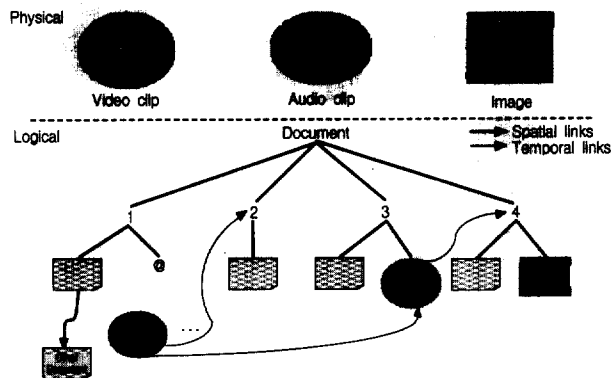


(그림 1) 동적인 시간에서의 레이아웃

#### 3.1 Xtime link

시간의 변화를 나타내는 링크의 순서는 시간적 속성을 표현하기 위해 사용된다. 시간 링크(temporal link)는 원본(original)과 대상(target) 오브젝트의 위치를 정의하는 것이다. 우리는 이것을 *Xtime link*라고 하며 웹에서 사용되고 있는 *hypertext link*와 유사한 개념을 갖는다. *Xtime link*는 시간 속성을 가지고 있으며, 이를 바탕으로 두개의 미디어 샘플을 시간 따라서 연결한다. *Xtime link*는 사용자의 개입을 필요로 하지 않는 능동적 개념을 가지고 있어서, 원본 미디어에서 대상이 되는 미디어로 가는 시간이 그만큼 빨라진다. 링크에 의해서 활성화되는 동작은 대상이 되는 미디어에 따른다. 만약 대상이 미디어 오브젝트의 시작 부분이거나 오브젝트 내부 어딘가에 있는 샘플이면, 링크는 대상이 되는 미디어를 활성화시킨다. 만약 대상이 미디어 오브젝트의 끝부분이면, 링크는 오브젝트를 종료시킨다. *Xtime link*는 원본의 위치와 대상의 위치간의 관계를 표현하며, 동작을 하게 되면 문서 안의 다른

곳으로 이동할 수 있도록 허용한다. (그림 2)는 멀티미디어 문서의 논리적 구조와 물리적 구조를 모두 보여주는 예이다. 굵은 선은 문서의 논리적 컴포넌트들 간의 관계를 나타내며 굵은 화살표는 다른 문서의 한 부분을 브라우저할 수 있도록 해주는 *hyperlink*를 나타낸다. 점선으로 된 화살표는 *Xtime link*를 나타내는 것으로 시간적 속성을 표현하는데 사용된다. (그림 2)의 예제는 비디오 클립의 작업이 끝나면 두 번째 노트의 텍스트를 디스플레이하고 오디오 클립을 활성화시킨다. *Xtime link*는 여러 가지 장점을 가지고 있다. 첫째, 현재 웹 상에서 널리 사용되고 있는 *hypertext link* 패러다임과 유사하다. 이것은 링크가 활성화되었을 때 시간과 장소를 정의하기 위해 *Xtime link*와 *hypertext link*를 혼합해서 사용할 때 유용하다. 예를들어, 프레임 50에서부터 60까지가 원본 링크라고 명시되어 있고, 사용자가 프레임이 재생되고 있는 중간에 클릭을 하게 되면, 링크는 활성화된다. 둘째, *Xtime link*는 어떠한 절대 시간에도 독립적이다. 왜냐하면, *Xtime link*는 단지 미디어 객체들간의 상대적인 시간 관계만을 명시하기 때문이다. 셋째, *Xtime link* 개념은 단순하지만 강력하다. 즉, 어떠한 형태의 시간적 시나리오라도 *Xtime link*로 표현할 수 있다.



(그림 2) 멀티미디어 문서의 논리적, 물리적 구조

### 3.2 XHTML+TIME 애트리뷰트

XHTML+TIME은 엘리먼트의 시간에 따른 동작을 표현하는데 사용된다. 본 장에서는 *Xtime link*, *Time points*, *Media objects*, *Dynamic layout*의 사용법을 예를 통해서 설명한다.

- **Xtime link** : *Xtime link* 태그는 미디어 오브젝트나 레이아웃에 정의되어 있는 원본과 대상의 관계를 나타낸다. 여기서 원본과 대상을 정의하는데 *time point*라는 개념을 사용한다.

```
<Xtlink orig="object-id.time-point-id | layout-id.time-point-id" target="object-id.time-point-id | layout-id.time-point-id">
```

### Examples :

```
<Xtlink orig="video.audio-trigger" target="audio.beg">
- audio-trigger가 time point에 이르면 오디오 클립이 시작된다.
```

```
<Xtlink orig="audio.end" target="video.end">
- 오디오 클립이 끝나면 비디오 클립도 끝난다.
```

- **Time points** : *Time points* 태그는 *Xtime link*의 원본이나 대상에서 사용되는 미디어 오브젝트나 레이아웃의 위치 지정을 정의할 때 사용되며, 두개의 포지션 타입이 있다. 우선 *nominal* 위치 지정은 미디어 오브젝트의 *nominal* 프레젠테이션 속도(rate)를 이용하여 정의한다. 또다른 하나는 *absolute* 위치 지정인데, 이것은 *time point*의 *absolute time* 조정자를 기반으로 하고 있다.

```
<time-point id="name" value="integer value | float value" unit="frame | sample | timestamp | second | ..." type="nominal | absolute">
```

### Examples :

```
<time-point id="audio-trigger" value="300" unit="frame" type="nominal">
- 만약 비디오 오브젝트가 있으면, time point를 frame 300으로 지정한다.
```

```
<time-point id="wait-20s" value="20" unit="seconds" type="absolute">
- 오브젝트나 레이아웃이 시작된 후에 time point를 20초로 지정한다.
```

- **Media objects** : 미디어 오브젝트들은 미디어 콘텐츠의 URL을 참조함으로써, 마스터(master)나 슬레이브(slave) 동기화 규칙을 표시하고 시간 변화의 정도(scale)를 정의한다. 태그안에는 다른 오브젝트들을 캡슐화(encapsulation)하고 *time points*와 동기화를 정의한다. SMIL과 달리, 여기서는 비디오와 이미지 오브젝트들을 구분하지 않는다. 미디어 오브젝트 태그는 레이아웃에 따라서 임의의 내용을 포함하고 있게 된다.

```
<object id="name" src="url" role="master | slave" scale="object-id | layout-id | float value"> objects synchronization points time points Xttime links </object>
```

**Examples :**

```
<object id="video" src="clip.mpg" role="slave" scale
="+2">...</object>
```

- 슬레이브 동기화 규칙을 가진 clip.mpg 파일을 두번 재생한다.

```
<object id="music" src="http://dbdoc.ajou.ac.kr/music.wav
role="master" scale="++1">...</object>
```

- http://dbdoc.ajou.ac.kr/music.wav에 있는 마스터 동기화 규칙을 가진 오디오 파일을 재생한다.

- **Dynamic layout :** 레이아웃 태그는 미디어 객체의 특별한 경우로써, 미디어 콘텐츠나 동기화 관계에 대해서는 규정을 하고 있지 않다. 하지만 레이아웃 태그는 스크린에 나타날 미디어 오브젝트의 위치를 정의해서 다른 오브젝트와 캡슐화한다.

```
<layout id = name scale = object-id | layout-id | float
value> frames time points Xtime links </layout id>
<frame id = name src = object-id | layout-id | url layer
= integer value shape = shape mask = mask>
```

레이어 속성은 오브젝트들이 서로 겹쳐질(overlaid) 수 있도록 오브젝트들간의 우선순위를 정의한다. shape와 mask 속성은 오브젝트에 특수 효과를 만들어 낼 때 사용된다.

**Example :**

```
<layout id = layout1 scale = video>
  <frame id = frame1 src = video>
  <frame id = frame2 src = logo>
  <frame id = frame3 src = layout2>
</layout>
<layout id = layout2>
  <frame id = frame4 src = movie>
</layout>
```

- 3개의 프레임을 중첩하여 사용한다.

**3.3 XHTML+TIME 선언부**

웹 브라우저가 XHTML+TIME의 내용을 인식하기 위해서는 몇 가지 선언 부분이 필요하다.

```
<html xmlns = "http://www.w3.org/TR/xhtml1">
- <html> 엘리먼트는 선언시에 XHTML+TIME의 네임
스페이스(namespace)를 포함 해서 선언해야만 한다.
<!DOCTYPE PUBLIC "-//W3CDTD XHTML 1.0 Strict//
EN" "">
```

- 파일 제일 앞부분에 DTD 선언을 제공해야 한다.

```
<style>
time {behavior: url (#default#TIME) ;}
</style>
```

- XHTML+TIME의 속성들은 스타일 블록 내에서 선언이 된다.

```
<?IMPORT namespace = "t" implementation = "#default
#time">
```

- 필요에 따라서 네임스페이스를 import하여 시간 속성을 추가할 수 있다.

아래에 나오는 코드는 웹 브라우저에서 XHTML+TIME을 다루기 위해 필요한 선언부를 포함한 XHTML 코드이다. 다음의 코드를 이용해서 기본적인 XHTML+TIME 페이지를 작성할 수가 있다.

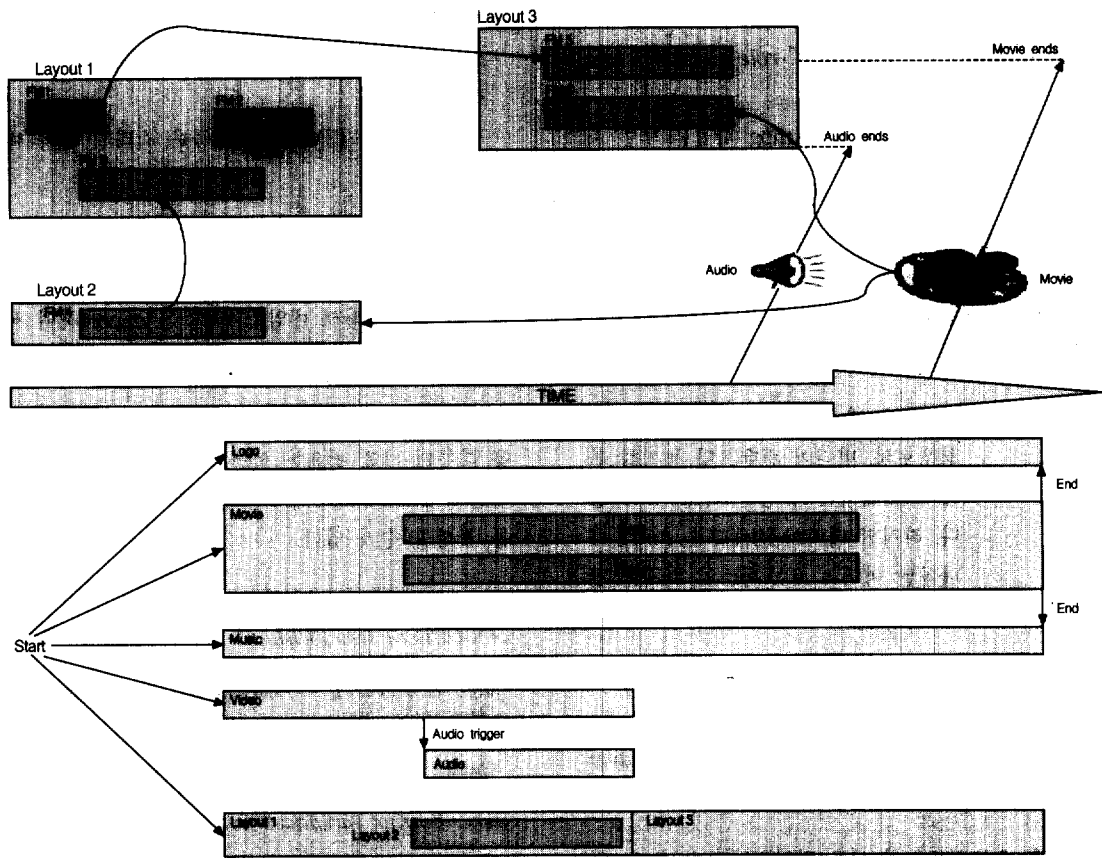
```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML1.0 Transitional//EN"
http://www.w3c.org/TR/xhtml1/DTD/xhtml1-transitional.dtd>
<html xmlns="http://www.w3c.org/1999/xhtml" xml:lang="en" lang="en">
<head>
<style>
time {behavior: url(default#time)}
</style>
<?IMPORT namespace="t" implementation=#default#time">
</head>
<body>
  <!-- replace this line with content -->
</body>
</html>
```

**3.4 멀티미디어 문서의 동기화 예**

(그림 3)은 동적 레이아웃을 이용한 하나의 시나리오를 보여주고 있으며, (그림 4)는 이 코드를 도식화한 그림이다. (그림 3)의 완전한 코드는 논문의 부록에서 다루고 있다.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML1.0 Transitional//EN"
http://www.w3c.org/TR/xhtml1/DTD/xhtml1-transitional.dtd>
<html xmlns="http://www.w3c.org/1999/xhtml" xml:lang="en" lang="en">
<head>
<style>
.....
.....
</style>
</head>
<?IMPORT namespace="t" implementation=#default#time">
<body>
<p> this is example of synchronization multimedia document
</p>
</body>
</html>
```

(그림 3) XHTML+TIME 코드



(그림 4) 코드의 도식화

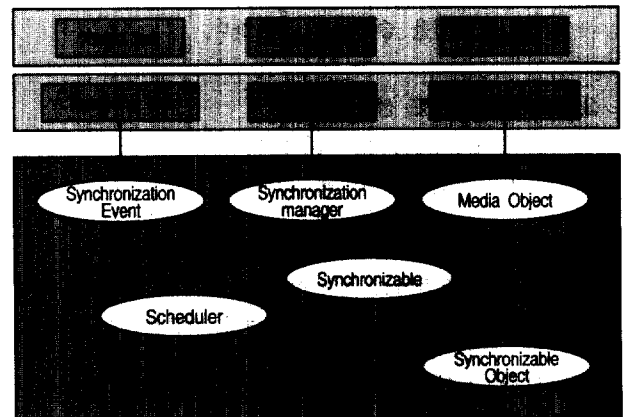
문서가 시작되면, 네개의 오브젝트가 활성화된다. 즉, 로고가 활성화되고, 나서 영화, 비디오, 음악을 나타내는 오브젝트들이 동시에 활성화된다. 여기서 영화는 *See*라는 이름의 비디오 세그먼트와 *Hear*라는 이름의 오디오 세그먼트를 포함하고 있는 복합 오브젝트이다.

이 예에서 영화 오브젝트는 내부에 시간 속성이 캡슐화되어 있다. 즉 비디오 세그먼트인 *See*를 시작하고 이것이 끝나게 되면 정지하는 것을 말한다. 모든 오브젝트는 두개의 중첩된 레이아웃을 이용해서 나타나게 된다. 비디오가 프레임 300에 이르면 오디오가 시작된다. 그리고 오디오가 멈추게 되면 레이아웃 1에서 레이아웃 3으로 레이아웃이 바뀌게 된다. 영화가 끝나게 되면, 음악과 로고도 또한 끝나게 되고 프레젠테이션이 끝나게 된다. 레이아웃 1과 레이아웃 2는 시간에 따라서 긴밀한 동기화가 이루어지고, 이것들은 미디어 오브젝트인 비디오와 오디오를 포함하고 있다. 다시 말해서 오브젝트들의 규칙에 따라서 동기화를 어떻게 시킬 것인가를 결정하게 된다.

오디오와 비디오가 같은 시간대에 위치하게 되면, 각 오브젝트가 가지고 있는 동기화 규칙이 마스터인지 슬레이브인지에 따라 오브젝트의 동기화를 제어하게 된다. 예를 들어 슬레이브 규칙을 가지고 있는 오브젝트는 마스터 오브젝트의 수행 결과에 따라서 오브젝트를 활성화시키거나 비활성화시키게 되는 것이다.

#### 4. 구현

(그림 5)는 논문에서 제안하는 XHTML의 시간적 확장을 그대로 문서를 실행하기 위한 실행 구조를 나타낸다. 이것은 하위레벨에서 이루어지기 때문에, 여기서 제안하는 모든 개념들은 시스템 레벨에 대응하는 부분을 가지고 있다. 실행 구조는 세계의 컴포넌트들로 구성되어 있다. 동기화 객체(synchronizable object), 동기화 이벤트(synchronizable event), 동기화 매니저(synchronization manager).

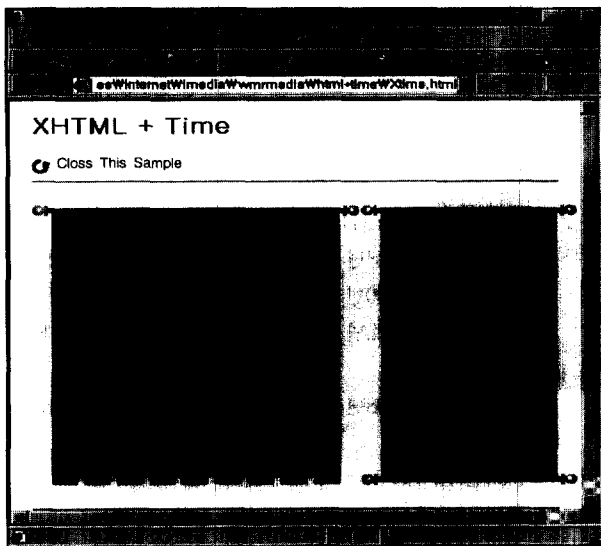


(그림 5) 실행 구조

동기화 객체는 미디어와 동기화를 통합하며 미디어가 재생되는 데 필요한 서비스들을 캡슐화시키고, 실행을 조정한다. 또한, 이것은 동기화 객체의 재생 비율에 따라 미디어 샘플의 스케줄링을 조절한다. 동기화 객체는 *synchronizable* 인터페이스를 이용한 래핑된 미디어 객체를 말하는 것으로 이러한 인터페이스는 미디어 객체를 동기화하기 위한 멀티미디어 구조가 필요로 하는 방법들을 정의하고 있다. 동기화 객체는 동기화 이벤트를 생성해 내고, 동기화 매니저에 의해 제어된다. 동기화에서 미디어 처리 함수들을 분리함으로써, 실행구조는 동적으로 확장이 가능해지고, 모듈화가 가능하다. 즉 새로운 미디어나 압축 형태의 미디어를 다룰 수 있는 오브젝트는 다른 오브젝트와 문제없이 통합이 가능하다는 의미이다. 동기화 이벤트는 여러 개체(entity) 사이에 시간정보를 전달하고 이후에 일어날 대상에서 실행되어야 하는 동작을 정의한다. 이것이 데드라인에 이르게 되면, 이벤트는 대상이 되는 개체가 수행해야 하는 동작을 전달하게 된다. *Xtime link*는 동기화 이벤트를 사용하면 쉽게 구현이 될 수 있다. *Xtime link*의 데드라인은 링크 정보가 함께 있는 상대적 시간정보에서 얻을 수가 있다. 동기화 매니저는 같은 시간대에 속해있는 동기화 오브젝트들을 관리하기 위한 동기화 정책을 정의한다. 매니저는 그들 자신의 시간속성에 대해서는 핸들링을 하지 않지만, 스케줄링 이벤트를 위한 내부 *scheduler*를 사용한다.

〈표 4〉 시스템 구현 환경

서버측 운영체제	Windows 2000 Server
웹 서버	Internet Information Server 5(IIS5)
웹 클라이언트	Microsoft Internet Explorer 5.5 이상
개발도구 및 언어	Java Script eXtensible Markup Language(XML) eXtensible Hypertext Markup Language(XHTML) XML Spy 3.5



(그림 6) 실행화면

본 논문에서 제안한 XHTML을 이용한 멀티미디어 프레젠테이션의 동기화는 SMIL과 같이 플레이를 하기 위한 특별한 플레이어나 브라우저를 필요로 하지 않으며 기존에 나와 있는 웹 브라우저인 마이크로소프트 익스플로러 5.5를 기반으로 하고 있다. 본 논문에서 제안한 시스템의 구현 환경은 표 2와 같다. XHTML과 시간 정보의 상호 연관성을 기반으로 한 데이터 모델을 구현하기 위한 프로토타입 시스템은 Windows NT 환경을 기반으로 하였다.

사용자가 작성한 *Xtime* 코드는 웹 브라우저 상에서 동작하는데 이를 전송해주는 웹 서버로는 Internet Information Server 5를 사용하였으며, 웹 클라이언트로는 마이크로소프트의 익스플로러 5.5 이상을 지원한다. (그림 6)은 마이크로소프트사가 제공하는 HTML+TIME의 샘플 예제를 본 논문에서 제안하고 있는 *Xtime*으로 바꾸어서 실행한 화면이다. 그림에서는 오른쪽의 비디오 클립과 왼쪽의 텍스트가 서로 동기화 되어서 보여지고 있다.

### 5. 결 론

멀티미디어 기술은 미디어 포맷의 인코딩, 디코딩과 네트워크를 통한 미디어 전송, 미디어 오브젝트의 서비스 품질 등과 같은 여러 분야를 포함하고 있다. 네트워크 전송 대역폭의 큰 발전에 힘입어 일반 문서에 애니메이션이나 오디오, 비디오와 같은 멀티미디어 오브젝트들을 포함하는 웹 문서들을 쉽게 전송할 수 있게 되었다. 본 논문에서는 멀티미디어 문서의 동기화를 이루기 위해 필요한 XHTML에 시간적 속성을 제안하였으며, 실행구조를 구현하였다. 제안된 시간 속성은 미디어 오브젝트들 간의 연결을 위한 *Xtime link*, 동기화 특성 제공을 위한 *time point*, 미디어 오브젝트들이 미디어 콘텐츠의 URL 참조와 마스터(master)나 슬레이브(slave) 동기화 규칙의 표시, 시간 변화의 정도(scale)를 정의하는 *Media objects* 그리고 마지막으로 미디어 오브젝트들이 스크린 상에 나타날 위치를 정의하는 *Dynamic layout* 등이 있다.

본 논문에서 제안하고 있는 XHTML 확장을 통해서 현재 표준화 작업이 진행되고 있는 SMIL2.0과 경쟁할 수 있을 것이라 기대된다. 향후 연구 과제로는 본 연구에서 제안한 XHTML에 다양한 프레젠테이션 효과나 프레임 재배치와 같은 새로운 속성들을 추가하는 것이다.

### 부 록

#### 1. XHTML+TIME 코드

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML1.0 Transitional//EN"
http://www.w3c.org/TR/xhtml1/DTD/xhtml1-transitional.dtd>
```

```

<html xmlns = http://www.w3c.org/1999/xhtml xml:lang="en"
lang="en">
<head>
<style>
<layout id="lay1">
<frame id="frm1" src="video"/>
<frame id="frm2" src="logo"/>
<frame id="frm3" src="lay2"/>
</layout>
<layout id="lay2">
<frame id="frm4" src="movie">
</layout>
<object id="video" src="clip.mpg" role="slave" scale="+2">
<time-point id="audio-trigger" value="300" unit="frame" type
="nominal">
</object>
<object id="audio" src="comment.wav" role="master">
</object>
</style>
<style>
<layout id="lay3">
<frame id="frm5" src="logo">
<frame id="frm6" src="movie">
</layout>
</style>
<style>
<object id="music" src="http://dbdoc.ajou.ac.kr/music.wav role
="master" scale="+1">
</object>
</style>
<style>
<object id="movie">
<object id="see" src="hello" role="slave">
</object>
<object id="here" src="hello.wav" role="master">
</object>
<xlink orig="here.beg" target="see.beg">
<xthlink orig="here.end" target="see.end">
</object>
</style>
<xlink target="lay1.beg">
<xlink target orig="video.beg target="music.beg">
<xlink target="video.beg">
<xlink orig="video.beg" target="music.beg">
<xlink orig="video.audio-trigger" target="audio.beg">
<xlink orig="audio.end" target="lay3.beg">
<xlink target="movie.beg">
<xlink orig="movie.end" target="music.end">
<Xtlink orig="movie.end" target="logo.end">
</head>
<?IMPORT namespace="t" implementation="#default#time">
<body>
<p> this is example of synchronization multimedia document </p>
</body>
</html>

```

**참고문헌**

[1] B. Jiao and B. Prabhakaran, "Using XML for Multimedia Presentations," *Draft*, Dec. 19, 2000.  
 [2] L. Rutledge, L. Hardman and J. V. Ossenbruggen, "Evaluating SMIL: Three User Case Studies," *Proc. of ACM Multimedia*, Orlando, FL Oct. 1999.  
 [3] S. Lee, K. Byeon and E. Hwang, "Template-based XML

Data Integration System," *Proc. of Int'l Conference on Electronic Commerce*, Seoul, Korea, Aug. 2000.  
 [4] S. Lee and E. Hwang, "Synchronized Multimedia Presentation using XHTML," *Proc. of Int'l Conference on Internet Computing*, Las Vegas, July, 2001.  
 [5] S. Lee and E. Hwang, "Extending XHTML for Multimedia Presentation," *Proc. of Int'l Conference on Distributed Multimedia Systems*, Taipei, Taiwan, Sept. 2001.  
 [6] E. Maler, S. DeRose and D. Orchard, "XML Linking Language(XLink)," *World Wide Web Consortium Working Draft*, <http://www.w3.org/TR/WD-xlink>, March, 1998.  
 [7] E. Maler, S. Daniel Jr. and S. DeRose, "XML Pointer Language(XPointer)," *World Wide Web Consortium Working Draft*, <http://www.w3.org/TR/WD-xptr>, March, 1998.  
 [8] D. Raggett, A. Le Hors and I. Jacobs, "HTML 4.0 Spec. (revised)," *World Wide Web Consortium Recommendation*, <http://www.w3.org/TR/REC-html40/>, April, 1998.  
 [9] P. Schmitz, J. Yu, J and P. Santangeli, "Timed Interactive Multimedia Extensions for HTML (HTML+TIME): Extending SMIL into the Web Browser," *Note submitted to the Wide Web Consortium*, <http://www.w3.org/TR/NOTE-HTMLplusTIME/>, Sept. 1998.  
 [10] SMIL Boston Specification, *Working Draft*, <http://www.w3c.org/TR/2000/WD-smil-boston-20000622/>, June 2000.  
 [11] W3C Recommendation, "eXtensible Markup Language XML 1.0," <http://www.w3c.org/TR/1998/REC-xml-19980210>, Feb. 1998.  
 [12] W3C Recommendation, "eXtensible HyperText Markup Language," <http://www.w3.org/TR/xhtml1/>, Jan. 2000.

**이 수 철**



e-mail : juin@madang.ajou.ac.kr  
 1998년 한남대학교 컴퓨터 공학과(학사)  
 1998년~2000년 아주대학교 정보통신 전문대학원(석사)  
 2000년~현재 아주대학교 정보통신 전문대학원 박사과정

관심분야 : 데이터베이스, 멀티미디어 시스템, 정보 통합, XML 응용

**황 인 준**



e-mail : ehwang@madang.ajou.ac.kr  
 1988년 서울대학교 컴퓨터공학과(학사)  
 1990년 서울대학교 컴퓨터공학과(석사)  
 1988년 Univ. of Maryland at College Park 전산학과(박사)  
 1998년~1998년 Hughes Research Lab. 연구교수

1998년~1999년 Bowie State Univ., Assistant Professor  
 1999년~현재 아주대학교 정보통신전문대학원 조교수  
 관심분야 : 데이터베이스, 멀티미디어 시스템, 정보 통합, 전자 상거래, XML 응용