Style Sheet 언어를 이용한 내용중심 마크업 언어와 SMIL의 통합에 관한 연구

이 원 익[†] · 염 세 훈[††] · 방 혜 자[†††]

요 약

웹상에서의 다양한 멀티미디어 지원에 있어 이들 미디어들은 서로 분리되어 있으며 정교한 프로그래밍 작업 없이는 다른 요소들과 상호작용하지 못했다. 1998년 6월 발표된 SMIL1.0은 다양한 멀티미디어들간의 상호작용을 가능하게 하였다. 지금까지의 웹기반 멀티미디어 동기화를 위해서는 DHTML이나 Plug-In등의 부가적인 방법이 사용되어 왔으나 본 논문에서는 SMIL Timing과 Style Sheet 언어를 이용하여 내용중심 마크업 언어에서의 구현이 용이한 멀티미디어 동기화 표현의 방법을 제시하고 그 특징을 살펴본다. 그리고 이를 실험하기 위하여 마크업 언어의 하나인 HTML에 제안된 방법을 적용하였으며 자바언어의 쓰레드 기술과 JMF 기술을 이용하여 시간적으로 미디어간의 동기화가 지원되는 브라우저를 설계 및 구현한다.

A Study of Integrating Contents-Oriented Markup Language and SMIL using Style Sheet Language

Won-Ik Lee[†] · Sae-Hun Yeom[††] · Hye-Ja Bang[†††]

ABSTRACT

In the case of supporting various multimedia on the Web, they are separate and can't interact with other multimedia without delicate programming. The SMIL 1.0, published on June 15, 1998, enables interaction between various multimedia. Until recently, DHTML or Plug-In has been used in order to synchronize web-based multimedia. However for SMIL Timing and Style Sheet language we propose a method for multimedia synchronized expression that can be embodied easily in contents-oriented markup language. To test this approach, we applied the proposed method to HTML, one of the markup languages. Also we designed and implemented a browser that is able to support synchronization between various multimedia using JAVA Thread technology and JMF technology.

키워드 : SMIL, HTML, XML, Web, Integration, Syncronization, Multimedia

1. 서 론

컴퓨터를 기반으로 하는 통신기술의 발전과 크게 성장한 인터넷 대중화는 디지털 멀티미디어 처리 환경을 요구하고 있다. 다양한 미디어들에 대한 동기화를 지원하기 위한 많은 기술과 툴들이 소개되고 있지만 웹을 기반으로 하는 동기화 처리 기술은 미약한 상태이다. 웹의 하부구조의 근간을 이룬 HTML은 누구나 사용할 수 있을 정도로 매우 단순하며, 매우 일반적이다. 단순히 텍스트와 이미지 중심으로 문서를 구성할 수 있다. HTML 표준안의 목적은 웹 개발자들이 페이지의 화면표시 방식에는 신경 쓰지 않고도 페이지의 구조를 기술할 수 있도록 하기 위한 것이다. 그러나 HTTP(Hyper Text Transfer Protocol) 프로토콜의 특성은 시간에 따른 멀티미디어의 상호 연관성을 여전히 표현하지 못하고 있다. 웹 사용자들은 이런 정적인 웹의 표현을 벗어나 다양하면서도 동기화 적인 멀티미디어의 표현을 원하고 있다. 기존 웹의 단순성과 정적인 면을 탈피하고자 스크립트 언어(Script Language) 및 스타일 언어(Style Language)들이 사용되었지만 여전히 시간 기반의 멀티미디어 표현은 부재한 상황이다.

웹 기반에서의 멀티미디어 동기화 연구의 필요성이 인식되어 오기 시작하여 SMIL 1.0이 발표되었다. SMIL은 멀티미디어를 시간에 따른 상호 연관성을 표현하기 위한 마크업 언어로 미디어간의 동기를 맞추는 프리젠테이션을 제공한다. SMIL 문서는 다른 여러 가지 미디어의 시간에

† 정 회 원 : (주)한올테크놀로지 부설연구소 연구원
†† 정 회 원 : 숭실대학교 대학원 컴퓨터학과 박사과정
††† 종신회원 : 서울산업대학교 전자계산학과 교수
논문접수 : 2000년 10월 18일, 심사완료 : 2001년 6월 25일

대한 상호 관련성을 정의하는 문서로서 HTML과 같이 단순한 구조로 이루어져 HTML사용자면 누구나 쉽게 제작이 가능하다. HTML은 웹상에서 표현대상에 대해서 시간에 독립적인 텍스트와 정지 영상의 단순한 표현으로 제한이 되지만 SMIL은 텍스트와 정지 영상 외에도 동영상과 음악 파일등의 여러 미디어들을 동시에 서로 유기적으로 연결시켜서 사용자들에게 TV와 같은 효과를 느끼게 할 수 있으며 표현 대상에 대해서 프리젠테이션이 가능하다[4, 7].

이러한 SMIL문서를 읽어 구동되는 SMIL Player들은 Applet Player와 Standalone Player이다. 이들은 모두 SMIL 문서(*.smil)를 인식하고 문서 내에 기술된 미디어와 SMIL Timing 정보를 분석하여 동기화된 미디어를 표현하고 있다.

SMIL은 다른 XML기반 언어에 적용하여 동기화적인 표현을 할 수가 있다. 일반적인 XML기반 언어는 내용 중심의 문서를 정의하는데 사용되며 이들 문서는 문서내의 미디어들간의 시간 관련 정보를 정의할 수 없다. 반면 SMIL은 미디어들간의 시간 관련정보를 정의하는 언어이다. 내용중심의 어떤 XML기반 언어에 SMIL의 Timing 기능을 결합할 수 있으면 XML기반 문서에서도 멀티미디어간의 시간 관련 상호 작용하기 위한 정보를 정의할 수 있다.

본 논문에서는 내용중심의 마크업 언어와 멀티미디어의 동기화 표현이 가능한 SMIL Timing을 통합하기 위한 방법에 대해 연구하였으며 HTML을 대상으로 한 SMIL Timing을 적용한 시스템을 설계하고 이를 구현하였다.

2. SMIL

2.1 SMIL의 발표

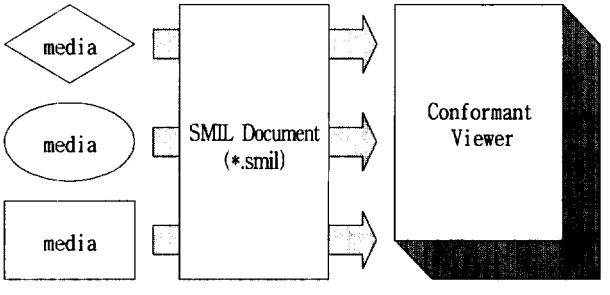
웹 사용자의 요구증대로 멀티미디어의 서비스를 제공하기 위해서 새로운 형식의 프로토콜이나 언어를 필요로 하였고 XML과 연관하여 스크립트 언어로 정의하고자 하는 노력과 멀티미디어의 복잡한 표현을 위하여 프로그래밍적인, 새로운 언어를 정의하려는 노력이 있다. 이러한 노력으로 미디어 간의 동기를 맞추어 프리젠테이션을 하기 위하여 제안된 선언적인 언어인 SMIL Specification 1.0을 발표하게 되었다[1-3].

2.2 SMIL의 특징

SMIL은 XML을 사용하는 장점이 있다. XML은 Audio, Video, Image, Text와 같은 동기화가 필요한 멀티미디어 객체의 시간적 공간적 연관성을 위한 SMIL Markup Tag를 정의할 수 있는 표준을 제공한다. SMIL의 중요 특징은 웹 개발자들이 시각적인 멀티미디어의 위치(location, size, z-index ordering)을 나타낼 수 있고, 서로간의 시간적인 속성(begin, duration end time)을 할당 할 수 있는 능력이 있다. SMIL은 Content Data를 웹과, 내부적인 측면에서 사용하는 것을 제외하고는 컴퓨터 기반의 상호작용의 CD들과 비슷한 상호작용의 멀티미디어 프리젠테이션을 만들 수 있는 능력을 제공한다. 추가적으로 SMIL은 웹 개발자들이 한번 생성된 Content는 밀접하게 연결된 상호작용의 CD에서처럼 Content 부분에서 어떠한 멀티미디어 객체(URL 주소화 되어진)들을 선택하여 나타낼 수 있다. 한번 생성한 후에는 전체의 Content를 재생성하거나 재패키지화 하지 않으면 어떤 데이터도 쉽게 변경할 수 없다. 마지막으로 SMIL 미디어 객체는 비의존적으로 갱신할 수 있고, 원격적으로 접근할 수 있다[6].

2.3 SMIL의 응용

1998년 6월 SMIL 1.0의 발표이후 많은 실험적인 도구들이 나왔다. 이러한 도구들은 크게 SMIL문서 제작 툴과 SMIL문서의 Viewer 두 부분으로 나누어 볼 수 있다. 제작 툴은 SMIL문서를 제작하기 위한 도구로 멀티미디어들의 시간적인 표현과 위치에 대한 표현을 Visual하게 작업할 수 있는 환경을 제공하고 있다. 이러한 툴의 예로는 TAG1.0 (Digital Renissance), GRiNS(CWI), SMIL Composer(Sausage Software)등을 들 수 있다. 그리고 동기화가 표현된 문서를 표현해서 보여주는 Application으로 SMIL문서의 동작기이다. 이들은 *.smil 문서를 읽어 처리하는데 Standalone Player이거나 Applet형태로 후자는 웹에서 보여질 수 있다. Standalone Player로써 예를 들면 RealPlayer(RealNetwork), GRiNS(CWI)등을 들 수 있으며 Applet기반 프로그램으로는 SOJA(Helio), S2M2(NIST)등을 들 수 있다.



(그림 1) SMIL문서와 Viewer와의 관계

지금까지 나와 있는 Viewer들의 SMIL문서처리 흐름은 (그림 1)과 같다. 미디어들에 대한 동기화 정보가 표현된 SMIL문서를 읽어 Applet이나 Standalone Player의 형태를 띄고 있다.

2.4 SMIL의 구조

SMIL은 HTML과 달리 텍스트와 정지영상 외에도 동영상과 음악 파일들을 동시에 유기적으로 연결시켜서 TV와 같은 효과를 줄 수 있고 표현 대상에 대해서 프리젠테이션이 가능하다. SMIL의 구조는 HTML과 아주 유사한 형태를 지니고 있다. XML 기반언어이므로 태그는 항상 쌍을 이루며 쌍을 이루지 않는 태그는 <element… />같은 형태를 지닌다. 문서 전체적인 구조는 <smil>…</smil>로 시작과 끝을 표현한다. 그리고 이 요소는 두 부분으로 크게 나뉘워진다. <head>…</head>와 <body>…</body>로 HTML과 유사하다. 하지만 내용 표현은 다음과 같이 다르다. <head>…</head>에서는 문서의 정적인 부분을 정의한다. 요소의 위치나 영역을 정의하며 <body>…</body>에서는 멀티미디어 객체들에 대한 시간적인 동기화에 대한 정의를 하게 된다[4].

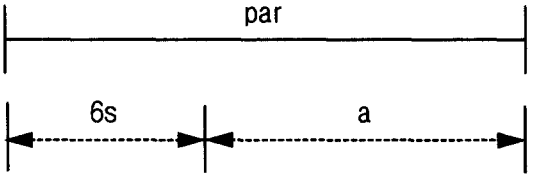
2.4.1 HEAD와 BODY 요소

<head>요소는 표현의 시간적인 동작과 관련이 없는 정보를 갖으며 layout, meta, switch 요소를 갖을 수 있다. <head>원소는 <META>요소를 무제한으로 포함가능하며, <layout>과 <switch>요소의 하나를 포함할 수 있다. <layout>요소는 <body>에 있는 요소들이 어떻게 추상적 출력물에 위치할 것인가를 결정한다. 이는 <region>요소를 갖으며 이 요소가 시각적으로 표현되는 동영상이나 정지영상을 표현하는 영역을 지정한다[4].

<body>는 문서의 시간적 그리고 링크 행위와 관련된 정보를 포함하고 있는 부분이다. 멀티미디어 개체들에 대한 시간적, 공간적 실행에 대한 정의와 각 객체들과의 링크를 정의하는 곳으로 개체들은 <a>, <animation>, <audio>, <img>, <ref>, <text>, <textstream>, <video>로 되어 있다. 시간적인 동기화를 표현하는 동기화 요소는 90%가 <par>와 <seq>로 사용된다. SMIL-Boston 워킹드래프트에서는 이것에 <excl>[5]등을 추가로 하고 있다. <par>는 병렬적인 실행으로 여러 멀티미디어 개체들은 동시에 수행하여야 한다. 이에 따른 각 개체들의 시간적인 정의를 하여야 한다. <seq>는 각 멀티미디어 객체들을 순서대로 실행을 시킨다.

SMIL의 <par>는 많은 속성을 갖고 있지만 Timing 관련 세 가지 중요 속성을 살펴보면 우선 begin은 원소의 명시적인 시작을 나타내는 값이며 end는원소의 명시적인 끝을 나타낸다. 그리고 dur은 원소의 명시적인 지속값 즉, 재생시간을 나타내는 값이다.

(그림 2)는 <par> 원소에서의 지연값이 어떤 의미를 갖는지를 보여주는 그림이다. <par>내의 미디어 들은 명시된 값만큼 시작을 지연하여 나타나게 된다. a라는 미디어는 par 속성을 갖으며 이는 begin값이 6초로 지정되어 있다. 동일한 par에 속해 있는 미디어들과 함께 시작되며 a 미디어는 시작으로부터 6초 뒤에 활성화 된다.



(그림 2) par 원소에서의 시작 지연 값의 사용

이와 마찬가지로 dur속성과 end의 속성을 명시적으로 지정하여 사용할 수 있다. end 값은 지정되지 않으면 암시적으로 <par>의 종료시간이지만 dur값이 지정되어 있다면 미디어가 활성화된 이후 dur 속성값 만큼의 시간이 지난 시간이 end값으로 처리된다.

```
<smil> <head> · </head> <body>
  <seq>
   <par>
    <img region = background  src = back.gif />
    <audio src = default_music.mid />
   </par>
   <img region = image_region begin = 5s  src = image1.gif >
  <seq>
</body></smil>
```

(그림 3) SMIL의 BODY 요소의 예

<body>요소는 또한 웹 상에서 다른 문서와의 연결을 위한 중요한 요소인 <anchor>요소를 사용할 수 있다. 이는 연결문서, 위치좌표, z-index, begin, end의 속성을 가진다. (그림 3)은 <body>부분을 보여주는 간단한 예다.

2.4.2 미디어 오브젝트

미디어들을 시간에 동기화 시켜 나타내기 위해서는 미디어의 특성에 따라 달라질 수 있다. SMIL에서는 미디어를 두 가지의 부류로 보는데 첫째, 연속적인 매체(Continuous Media)는 본질적 지속을 갖는 미디어 오브젝트로 예를 들면 동영상이나 음성을 들 수 있고 둘째, 불연속적인 매체(Discrete Media)는 본질적 지속을 갖지않는 미디어 오브젝트로 예를 들면 이미지나 텍스트 등을 들 수 있다.

3. 시스템 설계 및 구현

3.1 시스템의 설계

HTML문서는 태그의 조합으로 구성되고 있다. 이러한

구조에 SMIL Timing을 미디어들에게 적용하기 위해서는 HTML상의 태그로 표현되는 미디어들에게 어떻게 Timing을 적용할 것인가에 대해 살펴보아야 한다. 타이밍을 적용하기 위한 미디어에 대해서 시스템에서는 두 가지 미디어의 실험을 위하여 연속적인 미디어로써 동영상과 음악을 사용하기로 하고 불연속적인 미디어로써는 Text를 사용하기로 한다. 이들은 Media Player를 정의함에 있어 두 가지로 구분되어 사용되도록 한다.

3.1.1 Time Framework

Timing을 도큐멘트의 Content에 적용하기 위한 방법을 Time Framework이라 한다. 이러한 Time Framework은 여러 가지가 있을 수 있다. SMIL Timing을 XML언어에 추가하기 위한 개념을 SMIL-Boston에서 살펴보고 있다. SMIL Boston의 워킹그룹은 몇 개의 방법을 제안하고 연구중에 있다. 세번째 Working Draft에서는 In-Line Timing 방법, CSS(Cascading Style Sheet)를 통한 방법, Time-Sheet를 통한 방법을 제시하였으며 본 논문에서 구현하고자 하는 방법은 CSS로, 본 논문에서 구현하고자 하는 대상인 HTML에서 이미 사용하고 있는 것으로서 적용에 적합하다. 이는 동기화 되지 않은 문서(Non-Timed Presentation)를 보여줄 수도 있으며, 저작권이나 다른 제약이 있는 문서의 내용으로부터 분리하여 Timing을 적용할 수 있는 장점을 가지고 있다[5].

CSS를 이용하는 방법은 Timing을 Style Attribute를 이용하는 방법으로 CSS1이나 CSS2가 다른 Style을 Content에 적용을 가능하게 하는 것과 마찬가지로 도큐멘트의 Content에 있는 원소에 SMIL timing Style을 적용 가능하게 하고 있다. 이러한 CSS Timing은 도큐멘트에 추가되던가 아니면 참조될 수 있는 외부 문서로서 포함되어질 수 있다. 이는 도큐멘트의 Content와 Style정보의 분리를 가능하게 하여 재사용성을 높일 수 있도록 하고 있다[5].

3.1.2 SMIL의 Timing의 표현 위치

SMIL을 CSS를 사용하여 표현하면 기존의 CSS가 HTML에 적용해 왔던 방법을 그대로 사용하여 작성에 대한 새로운 방법을 배울 필요가 없다. HTML에 CSS를 적용하기 위해서는 CSS의 Syntax에 따라 Style을 작성하고 이를 외부 파일에 유지하거나 HTML자체내의 <head>~</head>안의 <style>~</style>안에 표현 할 수 있다. 따라서 SMIL Timing 또한 이와 같은 방법을 사용할 수 적용할 수 있다. HTML문서에서 CSS가 지금까지 사용하여오던 방법 그대로 적용이 가능하다. SMIL Timing을 적용하고자 하는 미디어는 HTML문서의 <head>~</head>안에서 <style>~</style>안에 CSS Syntax를 사용하여 기술하기로 한다.

3.1.3 timeContainer 속성의 정의

SMIL 1.0에서는 동기화 원소로 <par>와 <seq>를 제공하고 있다. 이 들은 Timing이 적용된 원소를 그룹화 하기 위한 원소로 Time Container Elements라고 표현한다. 미디어 원소와 이 time Container 원소들은 서로 내포관계에 놓일 수 있고 이를 표현할 수 있어야 한다. 미디어의 Timing 관련 그룹을 정의할 수 있도록 표현하기 위한 timeContainer 속성을 본 논문에서 사용하는 의미를 미디어의 동기화 방법의 표현을 위한 속성이라고 정의한다. 즉, 어떤 태그의 속성을 CSS를 사용하여 정의하는데 그 속성 중에 timeContainer가 있다면 이는 미디어들을 timeContainer값에 의해 동기화를 조절한다. timeContainer속성이 갖을 수 있는 속성값은 SMIL에서 제공하는 모든 동기화 원소의 의미를 가진 값이 될 수 있다. 본 시스템에서는 SMIL의 동기화 원소인 <par>와 <seq> 원소의 의미를 갖는 "par"와 "seq"로 한다.

어떤 태그가 timeContainer 속성을 갖고 있으면 이 태그 안에 기술된 다른 미디어 태그들은 timeContainer속성의 속성값의 동기화 방법에 따라 동기화 할 것이다. 또한 내포관계를 허용하여 par속성값을 갖는 timeConatiner속성이 있는 태그 안에 또 다른 timeContainer속성을 갖는 태그가 올 수 있다.

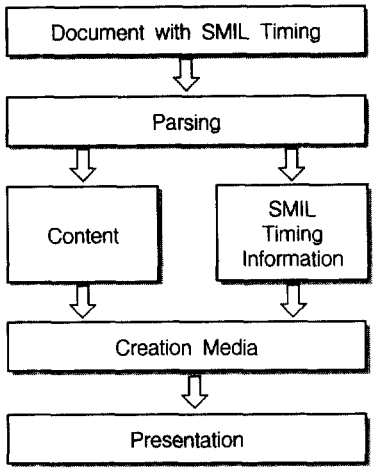
3.1.4 미디어의 SMIL Timing적용방법 정의

미디어에 SMIL Timing을 적용하기 위한 방법은 timeContainer속성을 사용하여 태그를 정의하는 것과 마찬가지로 HTML문서의 <head>~</head>의 <style>~</style> 안에서 표현한다. 미디어의 시작시간을 지정하는 begin 속성, 지속시간을 지정하는 dur, 종료시간을 지정하는 end를 속성으로 갖을 수 있으며 이들은 모두 정수를 갖으며 단위는 초(second)이다. 아울러 기존에 CSS를 사용하여 태그의 style을 지정하던 방법을 계속 사용 가능하다.

3.1.5 단계별 처리 내용

(그림 4)는 본 논문에서 구현하고자 하는 시스템 처리 흐름에 대한 블록도이다.

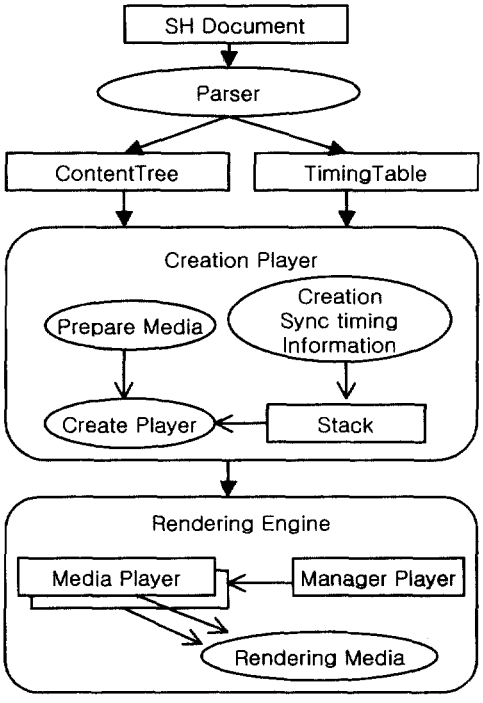
SMIL Timing이 적용된 문서를 파싱블록에서 파싱하여 도큐멘트 내용과 어떤 SMIL 타이밍 정보가 있는지를 생성하여 이 정보를 토대로 필요한 Player를 생성하고 이를 표현하기 위한 스케쥴 정보를 만들어 낸다. 이러한 정보로 시간에 동기된 미디어를 화면에 재생한다. 입력되는 문서의 확장자는 물론 *.html로 되어 HTML문서임을 확인한다.



(그림 4) 시스템의 처리 단계

3.2 시스템의 구현

SMIL Timing을 적용한 HTML 문서의 브라우저에 대한 시스템의 구성도는 (그림 5)와 같다. SMIL Timing이 통합된 SH Document(SMIL+HTML Document)로 본 논문에서는 HTML을 대상으로 하고 있다. SH Document의 문서구조와 SMIL Timing정보의 분리를 위한 Parser, 문서의 구조를 유지하기 위한 ContentTree, 적용된 SMIL Timing정보와 태그의 속성을 유지하기 위한 TimingTable, ContentTree와 TimingTable 자료구조를 순항하며 Player를 생성하고 등록하는 Player 생성 및 등록, 그리고 동기화된 멀티미디어의 표현을 담당하는 표현(Rendering Engine) 부분으로 구성하고 있다



(그림 5) 시스템의 구성도

3.2.1 SMIL Timing이 적용된 문서

HTML문서에 SMIL Timing을 적용하기 위해서 Style Sheet 언어를 사용한다. Style은 인터넷 익스프로러나 넷스케이프에서 모두 스타일 시트 기능들을 지원하고 있으며 문서의 구조와 문서를 표현하기 위한 정보를 따로 분리하여 표현하기 때문에 재사용성을 높일 수 있다. 현 HTML 문서에 사용되는 CSS Style 기능을 사용하여 timeContainer를 갖을 수 있는 태그를 정의하고 timeContainer의 속성값을 Style 지정방식으로 기술하게 된다. 또한 동기화 Time을 갖는 요소를 정의한다. 태그는 새로 정의하여 사용할 수 있으며 이는 <head>·</head>의 <style>·</style>에서 기술할 수 있다. 기존 HTML 태그가 동기화 정보가 필요하다면 새로 정의하는 태그와 마찬가지로 <style>…</style>에서 SMIL Time value를 사용하여 기술하여야 한다. 미디어 요소의 속성값은 begin, dur을 갖을 수 있으며 이들의 값은 정수로 표현되며 값의 의미는 초(second)이다.

문서에 사용할 수 있는 미디어 요소는 크게 세 가지로 구분한다. Text값을 Content로 갖는 태그들, 동영상을 표현하는 <video> 태그, 소리나 음악을 표현하는 <audio>태그로 구분할 수 있다. HTML문서에 적용하기 때문에 HTML에서 지원하는 태그 중에 Text를 Content로 갖는 태그는 위 세 부류에서 첫 번째에 해당한다. HTML문서 내에 사용된 태그 중에서 SMIL Timing을 적용하고자 한다면 이 또한 <style>…</style>에서 SMIL Model을 적용할 수 있다.

(그림 6)은 SMIL Timing을 적용하는 태그를 <style>…</style>에서 정의하는 것을 보이는 것이다.

```
<style>
        aa { timeContainer : seq ; }
        li { dur : 5 ; begin : 1 ; }
        qq { timeContainer : par ; }
        h1 { dur : 10 ; begin : 5 ; }
        video { begin : 5 ; dur : 10 ; }
</style>
```

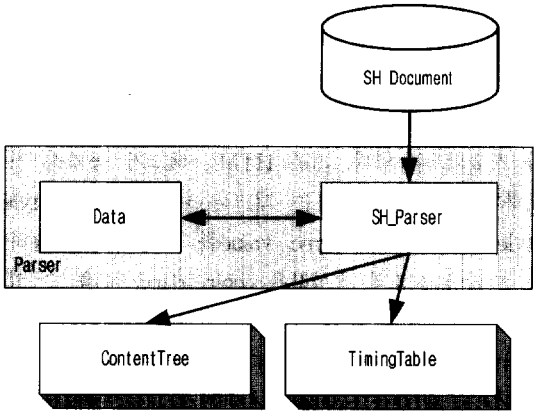
(그림 6) SMIL Timing을 적용하여 기술한 태그

3.2.2 SMIL Timing이 적용된 문서의 파싱

SMIL Timing이 적용된 문서 SH Document를 본 논문에서의 시스템에서는 HTML에 적용하므로 HTML구조 내에 SMIL Timing이 Style Sheet언어를 이용하여 적용된 문서이다. 따라서 HTML의 구조를 유지하며 이 자료를 각각 문서구조와 Timing정보를 추출하여야 한다.

Parser가 받아 들이는 문서는 HTML문서이다. 단지 기존의 HTML문서와 다른점은 SMIL Timing이 Style Sheet를 사용하여 적용되었다는 것이다. 이 HTML 문서를 Contet-

Tree와 TimingTable로 만들기 위해 Parser의 역할이 필요하지만 본 시스템에서는 HTML의 문법에 문제가 없음을 가정하고 태그의 이름과 속성값 등의 Token 분리와 문서 구조의 인식에 초점을 맞춘다. 따라서 Parser의 구성은 HTML문서를 바이트 스트림으로 유지하는 Data Class와 이와 상호 작용하여 ContentTree와 TimingTable을 생성하는 SH_Parser Class로 이루워지며 (그림 7)과 같은 구조를 갖는다.



(그림 7) Parser의 구조도

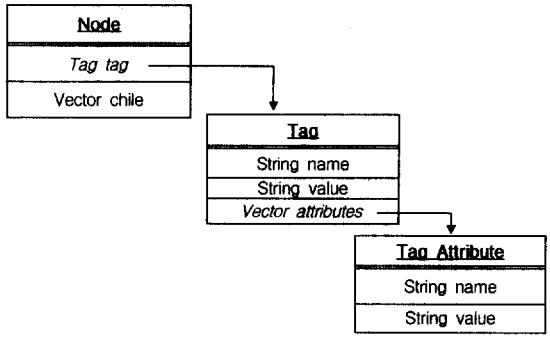
3.2.3 Content Tree

Parser에 의해 생성된 ContentTree는 SMIL Timing이 적용된 문서의 구조를 유지하기 위한 자료구조이다. 도큐멘트의 구조를 유지하기 위해서는 HTML문서의 <body>의 내용을 모두 정의할 수 있어야 한다. HTML의 본문은 HTML 마크업언어에서 사용하는 tag와 사용자가 <style>…</style>에서 정의하여 사용할 수 있는 태그로 이들은 여러가지 속성과 속성 값을 가질 수 있다. 동기화를 위한 태그는 SMIL Time Value 값을 갖는다. 따라서 이러한 구조를 유지하기 위해 다음의 클래스를 정의한다.

(1) Tag Attribute Class : 하나의 속성 이름과 속성값을 유지하기 위한 속성의 기본구조 Class (속성 : Tag , child Node List)

(2) Tag Class : Tag Class는 문서 내에 나타나는 모든 태그를 정의할 수 있는 자료구조로 이는 여러 가지 속성을 갖을 수 있으므로 Tag Attribute를 여러 개 유지하기 위해 Vector를 사용하여 Tag Attribute를 등록하여 사용한다(속성 : Tag Name , Tag value , Tag Attributes).

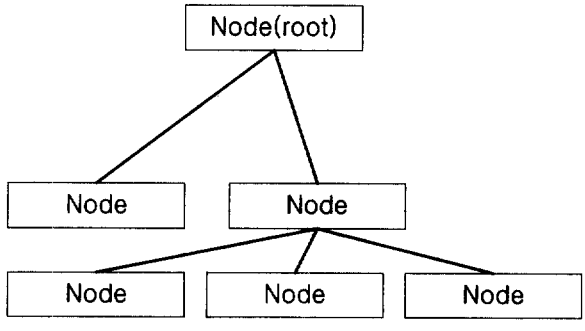
(3) Node Class : 다음으로 Node Class는 본문의 구조를 Tree로 표현하기 위한 Class이다. 각 노드는 즉, 태그는 자신의 하위에 여러 가지 태그를 갖을 수 있으므로 이들을 Vector로 유지하고 있어야 한다(속성 : Tag Attribute Name , Tag Attribute Value).



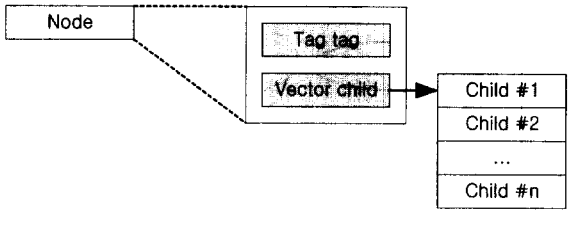
(그림 8) ContentTree 클래스 연관도

(그림 8)은 이들 클래스의 연관도를 나타내고 있다. 본문의 구조는 Tree로 구성하여 사용하는데 각 노드는 태그와 자신이 갖는 Node를 Vector로 갖는다. 이 Tree의 구조와 세부 내용을 다음 (그림 9)와 (그림 10)에 나타내고 있다.

객체 리스트를 표현하기 위해 JAVA에서 지원하는 자료구조인 Vector를 사용한다. Java.util.vector는 Object를 상속하여 설계된 클래스의 instance이면 등록이 가능하고 이들을 등록, 삭제, 조회등의 Method들을 제공한다.



(그림 9) ContentTree 의 전체구조

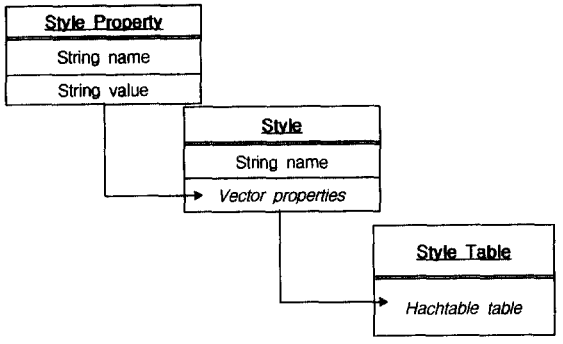


(그림 10) 한 노드의 세부구조

3.2.4 TimingTable

HTML문서내에 표현된 동기화 요소들의 Timing 정보를 유지하기 위한 자료구조가 TimingTable이다. 이는 <style>…</style>에서 정의한 태그의 이름과 속성 그리고 속성값을 갖는다. 이 자료구조는 태그의 속성을 갖는 것으로 SMIL

-Timing 정보와 함께 HTML에 적용해 오던 CSS Style Sheet 정보도 포함이 가능하다. 이는 SMIL Timing정보를 표현하기 위한 방법이 Style Sheet를 사용하는 방법이기 때문이다. 속성은 여러 개일 수 있으므로 여러 개를 유지할 수 있는 구조이어야 한다.

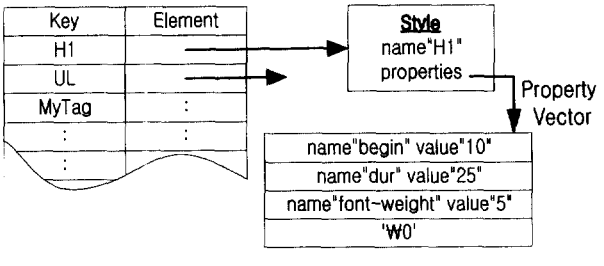


(그림 11) TimingTable의 클래스 연관도

따라서 Style Property Class와 이를 리스트로 관리하는 Style Class 그리고 문서의 전체 Style정보를 관리하기 위한 (그림 11)과 같은 자료구조 Style Table Class가 있다.

(1) Style Property class : 태그에 적용된 속성을 이름과 값으로 표현하는 단위 클래스
(2) Style class : 하나의 태그에 적용된 모든 속성을 유지하며 사용된 태그의 이름이 있는 클래스
(3) Style Table class : 문서에 적용된 모든 태그와 속성을 관리하는 클래스로 이 태그들을 리스트로 관리리스트로 구현하기 위한 정보는 역시 자바의 Vector 자료구조를 이용하였다.

StyleTable은 다음 절에서 설명할 Player를 만들 때 적용된 속성을 빠르게 찾아야 한다. 많은 미디어들이 동기화되어 나타내기 위해서는 Player의 생성과 재생이 빠르게 동작하여야 한다. 따라서 빠른 검색을 위한 메소드를 제공하는 자료구조인 자바의 HashTable을 사용하여 구현하였다. (그림 12)는 TimingTable의 내부구조를 설명하는 그림이다.



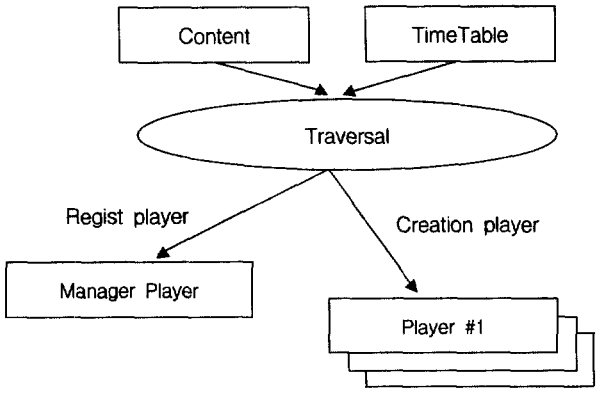
(그림 12) TimingTable의 구조

3.2.5 미디어 플레이어와 마스터 플레이어(Media Player, Master Player)

Player 생성 및 등록 부분에 대한 구현 내용으로 문서 구조와 Timing정보를 이용하여 Player를 생성하여 동기화 스케줄 정보를 생성하기 위한 부분이다. 각 미디어 마다 Player를 생성하고 이들을 구동하기 위한 MP(Master Player)가 있어야 한다.

Traversal은 Master Player와 Media Player를 만들어 내는 중요한 메소드로 이는 ContentTree를 Root Node부터 검색하여 leaf에 이르기까지 Recursive알고리즘으로 구현하고 있으며 tree를 순항하는 방법은 Depth First Algorithm을 사용하여 작성하였다. 노드에서 태그를 추출하여 태그가 갖고 있는 속성 중에 SMIL Timing 기능이 있는지를 검사한다. 만약 timeContainer속성을 갖고 있으면 Master Player를 생성한다. (그림 13)은 이러한 Player 생성와 등록에 관한 흐름을 도식화하고 있다.

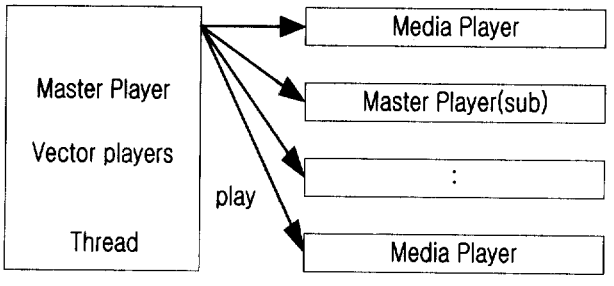
Player의 생성과 등록에 있어서는 태그의 속성 중에 timeContainer속성이 있으면 이는 Sub Master Player를 timeContainer속성의 값 형태로 동기화를 하겠다는 것이므로 Master Player를 생성하여 <body> 태그에 기 등록된 Master Player에 Child로써 등록을 한다. 그리고 다시 Tree의 Depth를 내려가 Child Node를 처리한다. 만약 노드의 태그 속성 값 중에 timeContainer가 없고 begin이나 dur값이 설정되어 있으면 이는 Media Player를 생성할 수 있는 조건인 것이다. 따라서 Master로부터 적절한 Play시간을 구해 Media Player에 Sleep 시간을 설정한다. 그리고 이렇게 생성된 Media Player는 자신을 감싸고 있는 timeContainer 속성을 가진 Master Player에 등록을 한다. Player는 크게 Master와 Media Player로 나뉘게 되는데 Master Player는 Sub Master Player를 가질 수 있으며 마지막 Master Player는 Media Player를 갖는다. Master Player는 동기화 처리를 위해 Thread로 작성되었다.



(그림 13) Player 생성 및 등록 부분의 흐름도

이러한 Thread 구성으로 병렬처리가 가능하다. par로 정의된 timeContainer속성을 갖는 태그는 이하의 태그에 대해 미디어 들을 병렬로 활성화 시켜야 한다. 이러한 병렬 처리를 하기 위해 프로세서의 사용이 몰릴 수 있으나 이를 Thread로 구현하여서 다른 미디어 처리도 함께 가능하게 되었다. 단, 하나의 CPU에서는 Thread 효과가 시분할 효과처럼 나타나므로 다중 프로세서 시스템에서보다는 멀티미디어의 동기화 역량이 떨어질 수 밖에 없다.

불연속적인 매체와 연속적인 매체 두 가지 매체를 실험하기 위해 전자로는 Video와 Audio를, 후자로는 Text를 사용하였다. 이들은 begin과 duration값을 유지하고 있으며 각 계산된 시작 시간까지 Sleep하고 있다가 begin시간만큼 후에 활성화 한다. 활성화란 각 미디어에 맞게 활성화 시키는 것으로 Text는 Visible을 조절하고 Video와 Audio는 미디어의 Play를 On/Off하는 것이다.



(그림 14) MP의 동작

(그림 14)와 같이 가장 상위 Master Player부터 자신에게 등록된 Player를 모두 Play 시킨다. 등록된 Player가 Master Player이면 Thread로써 동작시키고 아니면 나머지 Media Player를 Play한다. 그리고 각 미디어는 Thread Sleep을 진행하다가 정해진 시간에 활성화 또는 비활성화를 수행한다.

3.2.6 표현 (Rendering)

엔진의 중요부분은 두 가지로 크게 나눌 수 있다. 동기화를 위한 미디어의 객체 생성 부분과 이를 활성화 시키는 Play구조라 할 수 있다. 엔진의 중요 부분은 위에서 설명한 바와 같다.

본 논문의 시스템에서는 두 가지의 미디어를 실험하는데 불연속적인 미디어로는 Text를 기반으로 하는 사용자 정의 태그 및 HTML의 Text기반 태그로 화면에 표현하기 위하여 자바의 제공 클래스에 나타나 있는 AWT의 Label을 상속하여 정의한 TextPlayer가 있다. 연속적인 미디어로는 Video와 Audio를 사용하는데 이는 자바의 미디어 처리를 위한 JMF API를 사용하여 처리하는 MediaPlayer가 있다.

(1) TextPlayer : JDK 1.2.2. 버전의 AWT클래스 중 화면에 표현할 수 있는 Component로 Label이 있는데 이 클래스를 상속 받아 Text 미디어를 표현한다.

(2) MediaPlayer : JMF(Java Media Framework) 2.0버전을 사용하여 미디어의 표현을 처리하였다. JMF는 Audio 및 Video등의 미디어를 다양하게 처리 할 수 있는 능력을 제공한다. MediaPlayer는 JMF의 javax.media.Player를 생성하고 이 javax.media.Player의 Controll 이벤트를 받아 처리 할 수 있도록 ControllerListener를 상속 받아 구현하였다.

TextPlayer의 Play Method는 Master Player에 의해 수행된다. 미디어의 begin값과 duration의 값이 같으면 Freeze와 동일한 효과를 얻도록 하였다. Begin시간이 지정되어 있으면 그 시간까지 Thread를 사용하여 Sleep한다. Master Player는 Play()를 수행하고 돌아와 다른 미디어의 Play Method를 수행하게 된다. 이 때 호출하는 시간을 단축하고자 최소한의 기능으로 Thread만을 수행하게 하였다. Thread의 Sleep이 수행되면 TextPlayer를 Component의 SetVisible() Method를 사용하여 보이게 한다. MediaPlayer의 Play메소드는 src에 지정된 미디어 소스 파일을 사용하여 미디어 재생기를 만들어낸다. 이는 JMF2.0에서 제공하는 미디어 관련 API를 사용하도록 하였다. MediaPlayer의 Stop Method는 브라우저의 정지 상태변수에 정지상황으로 세트하고 할당되어 있는 자원을 해제하여 마무리한다.

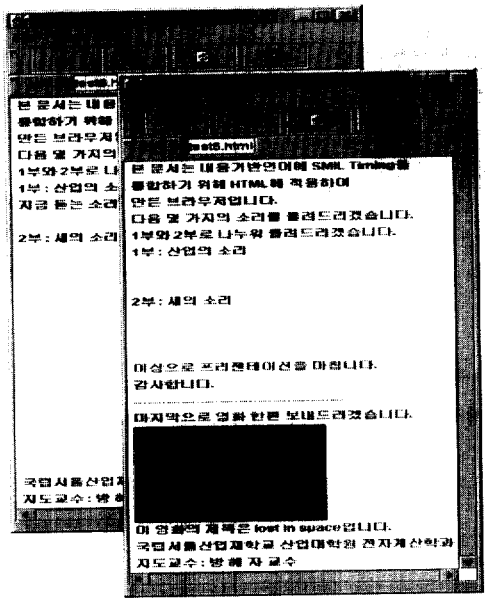
3.2.7 시스템 구현

시스템은 Java 언어를 사용하여 구현한다. 자바는 멀티미디어 프로그램을 위한 강력한 기능의 JMF를 제공하는데 본 논문의 구현에서 비디오와 오디오 미디어의 처리를 이 JMF를 사용하여 처리한다.

JMF는 자바에서 다양한 음악 파일과 동영상 파일을 재생할 수 있게 해주는 API 이다. 즉 여러 가지 미디어를 다루기 위한 자바의 솔루션이다[8]. 본 논문에서의 구현하고자 하는 시스템에서는 여러가지의 미디어를 처리하기 위해서 이 JMF 솔루션을 사용하기로 한다.

3.3 시스템 결과

(그림 15)는 4가지의 소리를 순차적으로 들려주며 이들의 소리를 설명과 동영상을 텍스트 미디어와 동기하여 나타내는 예제를 보인 화면이다.



(그림 15) 동기화되어 나타나는 브라우저

컴퓨터의 음향출력 장치로 소리가 출력되는 동안 브라우저의 화면에는 이와 동시에 이 사운드에 대한 설명을 간단히 표현한다. <seq> 내에 <par>로 사운드와 Text를 동기화하여 나타나게 하였으므로 소리가 나올 때 그에 병렬로 동기된 설명문장이 화면에 나타나게 된다. 정해진 시간동안만 나타나고 다음으로 넘어가 계속 진행하게 된다. SMIL Timing의 기능을 갖지 않은 태그는 시간에 관계 없이 프리젠테이션이 시작되면서 브라우저의 화면에 나타나게 된다. 앞의 (그림 15)는 이들 결과의 화면이다.

4. 결 론

본 논문에서는 멀티미디어의 동기화 표현에 관한 현황과 98년 발표된 SMIL언어와 SMIL의 현 상황에 대해 알아보았다. 그리고 SMIL Timing을 내용중심 마크업 언어에 적용하기 위한 방법을 제안하였으며 HTML언어를 대상으로 SMIL동기화 표현을 위한 브라우저를 설계 및 구현하였다. 웹상에서 멀티미디어를 동기화 하여 표현하기 위해 SMIL 문서를 달리 작성할 필요 없이 기존의 HTML문서를 작성하면서 동기화 하고자 하는 미디어들의 SMIL Timing을 <style>~</style> 에서 정의하여 사용하면 쉽게 제작이 가능하게 된다.

본 논문에서 구현한 시스템은 향후 내용중심 마크업 언어를 사용하는 다른 Application에서도 SMIL Timing적용 가능성을 보였다. 따라서 본 논문에서 제안한 방법으로 내용중심 마크업 언어를 사용하는 Application에 SMIL Timing이 적용되면 여러 가지 장점을 누릴 수 있다. 첫째, 하나의 문서로 동기화를 표현. SMIL 문서를 따로 작성할 필요가 없다. 둘째, 외부 Style File로 구현할 수 있어 Content와 Style의 분리가 가능하다. 셋째, Timing Style, 및 Rendering Engine이 대상 Application 브라우저에 적용되면 하나의 브라우저로 멀티미디어 동기화를 보일 수 있다.

웹에서 멀티미디어의 동기화를 효과적으로 보일 수 있다면 그에 따른 많은 응용 효과를 누릴 수 있으며 웹을 기반으로 하는 모든 사업분야에 적용해 볼 수 있다. 예를 들어, 인터넷 교육, 원격 화상회의, 인터넷 방송, 광고 등에 아주 효과적으로 적용할 수 있을 것으로 보며 또한 시공간의 표현의 프리젠테이션 성격이 요구되는 문서에 적용하여 사용할 수 있다.

참 고 문 헌

[1] 김지용, 고동일, 김두현, "웹 기반 멀티미디어 프로그래밍 동향", 정보과학회지, 2000년 4월호, pp41-42, 2000.

[2] 마이크로 소프트웨어 잡지 "HTML의 벽을 넘어 XML의 세계로" - 1, 2, 3회 98. 8~98. 10.

[3] KRNet'99 - 인터넷 마크업 기술 특강 자료 http : //www.krnet.or.kr/krnet99/특강/T131_new/sld001.htm, 1999.

[4] W3C, Synchronized Multimedia Integration Lnaguage (SMIL) 1.0 Specification, http : //www.w3.org/TR/REC-smil

[5] W3C, Second public Working Draft of SMIL-Boston, http : //www.w3.org/TR/smil-boston/, 1999.

[6] Wo Chang, Jin Yu, "S2M2 - A JAVA APPLET-BASED SMIL PLAYER."

[7] 김상국 외 5인, "SMIL 동작기의 설계 및 구현", 한국정보처리학회, '99년 춘계학술발표논문집.

[8] JMF : Java Media Framework, http : //java.sun.com/products/java-media/jmf.

이 원 익

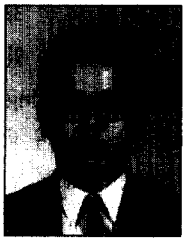
e-mail : slash@hanalltech.com

1998년 서울산업대학교 전자계산학과 졸업(학사)

2000년 서울산업대학교 산업대학원 전자계산학과 졸업(공학석사)

현 재 (주)한울테크놀로지 부설연구소 연구원

관심분야 : 프로그래밍 언어, 객체지향 분석설계, 데이터베이스, 컴퓨터 네트워크, 운영체제 등



염 세 훈

e-mail : shyeom@it.soongsil.ac.kr

1992년 서울산업대학교 전자계산학과 졸업(학사)

1994년 숭실대학교 대학원 전자계산학과(공학석사)

1994년~현재 숭실대학교 대학원 컴퓨터학과 박사과정

관심분야 : 프로그래밍 언어, 인간과 컴퓨터 상호작용, Speech(Voice) Based Markup Language

방 혜 자

e-mail : hjbang@duck.smut.ac.kr

1977년 숭실대학교 전자계산학과 졸업(학사)

1983년 University of North Texas M.S. Computer Science 졸업

1993년 숭실대학교 대학원 전자계산학과 졸업(공학박사)

한국정보처리학회 종신회원

1985년~현재 서울산업대학교 전자계산학과 교수

관심분야 : 프로그래밍언어론, 컴파일러, 계산이론 등