

MP3 태그의 XML 확장을 이용한 동기화된 재생 시스템

곽 미 라[†] · 조 동 섭^{††}

요 약

고품질의 오디오 표준인 MP3 포맷의 사용이 증가하면서, 오디오 데이터 외에 작곡가, 가사 등의 관련정보를 함께 저장하려는 요구가 나타났고 이를 만족하는 태깅 시스템들이 등장했다. 특히 ID3 v1 태그와 Lyrics3 v2 태그를 함께 사용하는 태깅 방법이 많이 사용되고 있다. 그러나 이 태그들은 MP3 파일 내에서 오디오 스트림의 뒷부분에 기록되므로, 이러한 태깅 방법이 적용된 MP3 파일이 스트리밍 방식으로 전달되는 경우 사용자는 전체 스트림이 로컬 시스템에 전송되기 전까지 태그 정보를 볼 수 없다. 또한 태그 정보들 중 오디오 스트림에 시간적으로 동기화된 정보들은 동기화의 기능을 잃는다. 본 논문에서는 원격지로부터 전달되는 MP3 파일의 재생 시 태그 정보가 무시되는 문제를 해결하였다. XML을 사용하여 MP3 오디오 객체를 모델링하였고, 그 요소들의 시간관계성과 동기성을 HTML+TIME 방식으로 표현하는 XSL 문서를 설계하여 오디오 데이터가 시간성과 동기성을 가지고 웹 상에서 재생되도록 하였다.

Synchronized MP3 Playing System Using XML Extension of MP3 Tag

Mira Kwak[†] · Dong-sub Cho^{††}

ABSTRACT

MP3 audio format has good quality and high compression rate ; therefore, the use of MP3 format increases. The requirement of keeping the extra information such as author and lyrics in MP3 files increases. And the tagging systems designed to meet this requirement are suggested. ID3 v1 tag and Lyrics3 v2 tag are two most widely used tagging systems. But ID3 v1 tag and Lyrics3 v2 tag are the last things to arrive when the file is being streamed. Therefore, users cannot get the tag information until the entire audio file is downloaded. Moreover information synchronized with audio stream may lose its feature. In this paper, a system searching and playing audio files based on tag information in MP3 files is implemented. This system solves the problem that the tag information is ignored when an MP3 files is played on internet. An audio object is described in an XML document, and timing and synchronization between elements in that XML document is provided in HTML+TIME style using XSL.

키워드 : MP3, 태그(tag), XML, 동기화(Synchronitaton)

1. 서 론

높은 압축률과 좋은 음질을 가지는 MPEG 오디오 기술 중 하나인 MP3는 그 장점을 인정받아 다양한 분야에서 많이 사용되고 있다. MP3의 사용이 증가하면서 오디오 데이터 자체 외의 추가적인 정보를 함께 저장하는 데 대한 요구가 생겼으며, 이러한 요구를 만족하는 태깅 시스템들이 개발되어 현재의 MP3 파일은 관련 파일, 저작권 정보, 가사 및 관련 이미지 등을 포함할 수 있게 되었다[7, 12].

MP3 파일 내에 오디오 데이터에 대한 추가 정보를 저장하는 방법으로 ID3 v1 태그와 Lyrics3 v2 태그를 함께 사용하는 방법이 가장 널리 사용되고 있다[7]. 사용자들은 태그 정보를 해석할 수 있는 재생기를 사용하여 로컬 디스크의 MP3

파일을 재생할 때 그 태그 정보를 볼 수 있다. 그러나 ID3 v1 태그와 Lyrics3 v2 태그는 MP3 파일 내에서 오디오 스트림의 뒷부분에 기록되기 때문에, 이러한 태깅 방법이 적용된 MP3 파일이 스트리밍 방식으로 인터넷 상에서 방송되는 경우 사용자는 오디오 스트림의 재생 시작과 동시에 태그 내의 정보를 볼 수 없다. 또한 이 경우 오디오 스트림에 대한 시간적 동기성을 가진 정보들은 동기화의 장점을 잃게 된다. 또한 사용자의 로컬 디스크에 저장되어 있는 MP3 파일들의 경우 태그 정보들을 기반으로 검색될 수 있지만, 사용자가 오디오 서버에서 원하는 파일을 찾는 경우 태그 정보에 대한 질의를 수행할 수 없다.

본 논문에서는 이와 같이 스트리밍 방식으로 MP3 파일을 재생하는 경우 해당 파일의 태그 정보가 무시되는 문제를 해결하고자 하였다. 이를 위하여 새로운 MP3 동기화 시스템을 제안하였다. 본 연구에서 제안하는 시스템은 XML을 사용하여 MP3 파일들을 다루도록 설계되었다. 각 MP3 파일의 오

† 정 회 원 : 이화여자대학교 대학원 컴퓨터학과
 †† 종 신 회 원 : 이화여자대학교 컴퓨터학과 교수
 논문접수 : 2001년 2월 6일, 심사완료 : 2001년 12월 12일

디오 정보와 태그 내에 기록된 모든 정보가 XML 문서로 표현되며, 사용자는 오디오 서버에 접속하여 원하는 오디오 데이터를 다양한 정보를 기반으로 찾을 수 있다. 또한 XML 문서로 표현된 각 오디오 데이터는 적절하게 작성된 XSL 파일에 의하여 렌더링 됨으로써 해당 오디오 데이터의 재생 시 오디오 스트림에 대한 가사 및 이미지 파일의 동기성을 표현할 수 있다. 이리 하여, 이미 오디오 데이터에 함께 저장된 유용한 추가 정보들과 동기화된 가사 정보 등을 표준적인 방법으로 활용할 수 있게 되었다.

2. MP3 오디오 포맷과 태깅 시스템

MP3, 즉 MPEG 오디오 레이어 3는 오디오 데이터를 작은 크기의 블록들로 나누어 압축하여 전체 사운드를 표현함으로써 스트리밍에 적합하게 설계된 포맷이다. 각 블록은 디코딩에 필요한 약간의 정보들을 기록한 헤더를 가진다. 헤더 공간은 'copyright'나 'private'와 같은 파라미터에 대하여 예 혹은 아니오의 값을 기록함으로써 부수적 정보의 저장에 사용되기도 하였다.

MP3 파일의 사용이 증가함에 따라 파일 내에 보다 풍부한 텍스트 정보를 포함하고자 하는 요구가 생겼고, Eric Kemp는 오디오 파일의 끝에 위치하며 곡명, 아티스트, 앨범, 연도, 장르, 주석 필드를 가지는 128 바이트 고정 크기의 태그를 제안하였다[12]. 이 태그가 ID3라는 이름으로 구현되었으며, 곧 이어 Michael Mutschler는 ID3 태그의 주석 필드를 수정하여 트랙 정보를 기록할 수 있는 ID3 v1.1 태그를 제안하였다.

ID3 v1과 ID3 v1.1의 단점은 크기가 128 바이트로 제한되어 있다는 점과 표현할 수 있는 내용이 한정되어 있다는 점이다. 이러한 단점을 극복하기 위하여 사용자가 원하는 필드를 정의하여 사용할 수 있고 그 크기가 동적인 태깅 시스템들이 설계되었다. 이러한 태깅 시스템에는 ID3 v1의 확장인 ID3 v2 태깅 시스템, 오디오 데이터를 위한 가사 표시를 주목적으로 설계된 Lyrics3 v1 태깅 시스템과 이를 확장한 Lyrics3 v2 태깅 시스템이 있다[7, 11, 12].

현재 ID3 v1 태그와 Lyrics3 v2 태그를 함께 사용하는 태깅 방법이 가장 널리 사용되고 있으므로, 본 논문에서는 이러한 방법으로 태깅된 MP3 파일을 대상으로 한다.

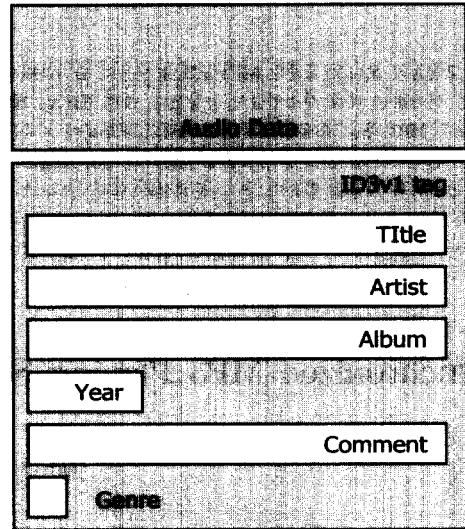
2.1 ID3 v1 태깅 시스템

ID3 v1 태그는 그 감지를 쉽게 하기 위하여 128 바이트의 고정 크기를 사용하며, (그림 1)과 같은 구조를 가진다[12].

ID3 v1 태그에는 (그림 1)에서 보듯이 곡명, 아티스트, 앨범, 연도, 주석, 장르 정보를 기록하기 위한 필드들이 정의되어 있으며, 각 필드의 크기는 <표 1>에서 보는 바와 같다.

<표 1>에 표시된 필드들의 크기의 합은 125 바이트이며, 여기에 태그의 앞머리에 오는 "TAG" 키워드의 크기 3바이트를 더한 것이 ID3 v1 태그의 전체 크기인 128 바이트이다.

아티스트의 이름이 30 바이트 미만인 경우 'artist' 필드의 나머지 부분은 '0'으로 패딩된다. 'genre' 필드에는 0부터 147까지의 숫자를 기록한다. 장르 정보를 위하여 0부터 147까지의 각 숫자에 대응하는 장르 이름이 미리 정의되어 있다[11].



(그림 1) ID3 v1 태그가 사용된 MP3 파일 구조

<표 1> ID3 v1 태그 내 필드들의 사이즈

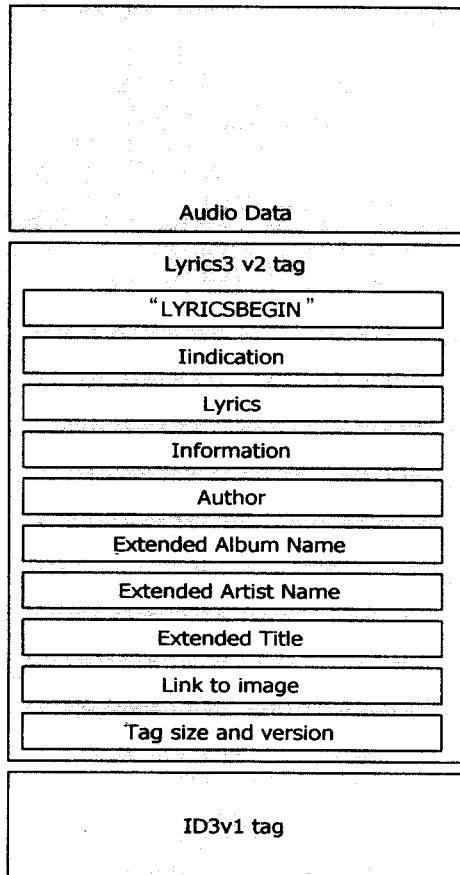
항 목	크기 (단위 : 바이트)
곡명 (Title)	30
아티스트 (Artist)	30
앨범 (Album)	30
연도 (Year)	4
주석 (Comment)	30
장르 (Genre)	1

2.2 Lyrics3 v2 태깅 시스템

초기의 MP3 파일 재생기를 위한 가사 표시기는 오디오 데이터와 분리되어 별도의 텍스트 파일에 저장된 가사 정보를 가져와 화면에 출력하도록 설계되어 있었다. Petr Strnad는 이러한 가사 표시 방법을 개선하기 위해, 가사 정보를 저장하는 태그를 개발하여 오디오 파일 내에 텍스트를 함께 저장할 수 있도록 하였다[12]. 이것이 Lyrics3 v1 태그이며 "LYRICS-BEGIN"이라는 키워드와 "LYRICSEND"라는 키워드 사이에 가사 정보를 담고 있는 데이터 덩어리의 형태로 구현되었고, MP3 파일 내에서 오디오 데이터 뒤, ID3 태그 앞에 저장된다.

Lyrics3 2.00 태그는 이전 버전의 Lyrics3 태그보다 복잡하다. 그러나 보다 많은 능력을 가진다. 이전 버전의 Lyrics3 태그와 같이 Lyrics3 v2.00 태그는 오디오와 ID3 태그 사이에 위치하며, 내용부터 크기 기술까지 모두 텍스트를 사용한다. MP3 파일 내에서 오디오 다음 ID3 태그 앞에 오는 Lyrics3

블록은 "LYRICSBEGIN"이라는 문자열로 시작하고 그 다음에는 여러 개의 필드 레코드들이 온다. Lyrics3 블록은 Lyrics3 태그 정보 크기를 기술하는 여섯 바이트와 "LYRICS-200"이라는 문자열로 끝난다. Lyrics3 v2.00 태그의 구조는 (그림 2)와 같다.



(그림 2) ID3 v1 태그와 Lyrics3 v2 태그를 사용한 MP3 파일 구조

Lyrics2 v2.00은 정보를 표현하기 위하여 필드라는 것을 사용한다. 필드 안의 데이터는 아스키 문자들로 이루어진다. 사용 가능한 필드의 수를 제한하는 것은 태그 사이즈 뿐이다. 모든 필드들은 3 바이트의 필드ID, 정보의 크기를 설명하는 6 바이트, 실제 정보로 이루어진 간단한 구조를 사용한다. 이는 알려지지 않은 필드들을 읽고 이들을 태그에 되쓰는 것을 가능하게 한다. 태그 안에 반드시 나타나야 하는 필드는 없

다. 지켜져야 하는 규칙은 인디케이션 필드가 사용되는 경우, 이것이 Lyrics3 태그 내에서 가장 처음에 오는 필드여야 한다는 것이다. 여러 줄을 포함하는 필드들은 줄 사이의 구분을 위하여 "[CR][LF]"를 사용한다

<표 2> Lyrics3 v2 태그의 정의된 필드들

ID	최대크기	설 명
IND	00002	2 바이트의 고정크기를 가지는 필드로, 이 필드에 가사 필드의 유무, 가사 필드 내 타임스탬프의 유무가 표시된다.
LYR	99999	가사를 저장하는 여러 줄의 텍스트 타임스탬프가 여기에 나타나며 그 형식은 "[mm : ss]"
INF	99999	추가적인 정보를 저장하는 여러 줄의 텍스트
AUT	00250	가사/음악의 저작자 이름
EAL	00250	확장된 앨범 이름
EAR	00250	확장된 아티스트 이름
ETT	00250	확장된 트랙 타이틀
IMG	99999	BMP나 JPG 형식의 이미지 파일로 링크. 이미지 링크 정보가 여러 개 포함될 수 있으며, 각 이미지 링크 정보는 파일이름, 설명, 타임스탬프로 구성된다.

<표 2>는 현재까지 정의된 필드들에 대한 설명이다. Lyrics3 태그의 새로운 버전에는 필요에 따라 더 많은 필드들이 추가될 수 있다[11].

2.3 태그가 있는 MP3 파일의 재생 지원 현황

현재 Wplay, WinAmp, Breathe, C-4, Nad, MPFree, 소리통, K-Jöfol, Xaudio, Digideck, Yo!MPEG, AudioActive, Sonique, Virtuosa, Soundream, Unreal, Jet-Audio, WinPlay3, museARC, Maplay, FreeAMP를 비롯한 많은 MP3 파일 재생기들이 개발되어 있다. 이들 중 특히 널리 사용되고 있는 재생기들의 특징을 ID3 태그와 Lyrics3 태그 지원 여부, 스트리밍 방식의 재생 지원 여부, 해당 재생기 개발자 혹은 개발업체에서 자체적으로 구현한 데이터 동기화 기술 존재 여부를 기준으로 비교해보면 <표 3>과 같다.

재생기 차원에서 ID3 태그를 지원하는 경우는 많으나 Lyrics3 태그를 지원하는 재생기는 드물다. Lyrics3 태그 내에 저장된 가사 정보를 오디오 데이터에 동기화하여 출력하는 기능은 대개 plug-in 형태로 제공되고 있다. 이러한 재생기들은 인터넷을 통하여 스트리밍 방식으로 오디오 데이터를 재생하는 사용자들에게 재생 시작과 함께 태그 정보를 보여

<표 3> MP3 재생기 기능 비교

	WinAmp	Sonique	WplayPro	K-Jöfol	FreeAmp	소리통	Mr.MP3
ID3 태그 지원	○	○	○	○	○	○	○
Lyrics3 태그 지원	○*	×	×	×	×	×	×
스트리밍 지원	○	○	○	○	○	×	×
자체적 데이터 동기화 기술	×	×	×	×	×	○	○

*은 플러그인을 통한 지원임.

주거나 재생되는 동안 가사 및 이미지를 오디오 스트림에 동기화하여 화면에 표시하는 방법을 제공하지 못하고 있다.

본 논문에서는 이러한 한계를 극복하여 인터넷을 통한 MP3 파일의 검색과 재생에 태그 정보를 이용할 수 있고, 파일이 스트리밍되는 동안 가사와 이미지 정보를 시간적으로 동기화하여 화면에 나타내는 재생 시스템을 설계하고 구현하였다.

3. 동기화된 멀티미디어 데이터의 웹 문서 표현

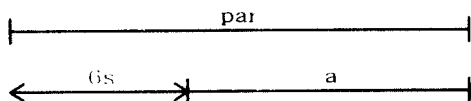
인터넷 환경에서 멀티미디어 데이터의 중요성이 인식되기 시작하면서 오디오와 비디오 데이터를 웹 페이지에 삽입하는 방법들이 개발되어왔다. 근래에는 웹 상에서 오디오 및 비디오를 실시간으로 방송함과 동시에 여러 매체 내에 포함된 다양한 정보를 동기화하여 표현할 수 있는 수단의 필요성이 대두되었다. 본 절에서는 이를 위하여 W3C 표준으로 개발된 웹 언어들인 SMIL과 HTML+TIME을 시간 동기성의 표현 방법을 중심으로 고찰한다.

3.1 SMIL(Synchronized Multimedia Integration Language)

SMIL은 독립적인 멀티미디어 객체들을 동기화된 멀티미디어 프레젠테이션으로 통합하기 위한 표준적인 방법으로 제안되었다. SMIL 문서는 디스플레이 영역을 정의하고 프레젠테이션 내의 다른 영역에 대한 레이아웃을 설정하는 문서 머리(head) 부분과 프레젠테이션 시나리오를 표현하는 문서 몸체(body) 부분으로 구성된다. 문서 헤드 부분은 head 요소로 표현되며, region 요소, root-layout 요소를 가지는 layout 요소를 포함한다. 문서 몸체 부분은 body 요소로 표현된다. body 요소는 par 요소, seq 요소, 여러 가지 미디어 객체 요소와 하이퍼링크 요소 등으로 구성된다. par 요소, seq 요소 그리고 animation, audio, img, video, text, ref, textstream 등의 미디어 객체 요소들은 문서 내의 미디어 객체들 사이의 시간성과 동기성 표현을 위하여 사용된다.

SMIL 문서 내에서 미디어 객체 요소들 사이의 병렬적인 시간 동기화는 par 요소를 통해 구현된다. par 요소는 시간축 상에서 병렬적으로 진행되는 요소들을 자식 요소로 가진다. 여러 가지 미디어 객체 요소들이 par 요소의 자식 요소가 될 수 있으며, 미디어 객체 요소들은 상연 시작 시간을 명시하는 begin 속성, 상연 기간을 명시하는 dur 속성, 상연 종료 시간을 명시하는 end 속성을 가진다. 다음과 같은 코드가 기술하는 오디오 요소의 재생은 (그림 3)과 같이 표현된다.

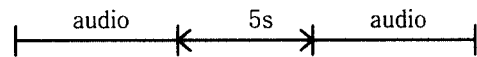
```
<par>
  <audio id="a" begin="6s" src="audio"/>
</par>
```



(그림 3) 병렬적 시간 동기화

문서 내의 미디어 객체 요소들 사이의 직렬적인 시간 동기화는 seq 요소를 통해 구현된다. seq 요소는 시간축 상에서 직렬적으로 진행되는 요소들을 자식 요소로 가진다. par 요소와 마찬가지로 seq의 자식요소는 begin, dur, end 속성을 가지는 미디어 객체 요소이다. 다음의 코드가 나타내는 오디오 요소의 재생은 (그림 4)와 같이 표현된다.

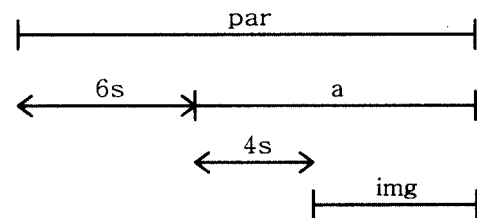
```
<seq>
  <audio src="audio1"/>
  <audio begin="5s" src="audio2"/>
</seq>
```



(그림 4) 직렬적 시간 동기화

미디어 객체 요소는 사건 발생을 기준으로 상연을 시작할 수 있다. 다음의 코드는 사건 발생을 기준으로 시간적으로 동기화된 미디어 요소 재생의 예를 보여준다. (그림 5)는 이것을 시간축 상에 그림으로 표현한 것이다.

```
<par>
  <audio id="a" begin="6s" .../>
  <img begin="id(a)(4s)" .../>
</par>
```



(그림 5) 사건 발생을 기준으로 한 시간 동기화

3.2 HTML+TIME(Timed Interactive Multimedia Extensions for HTML)

SMIL은 벤더들이 개발한 배타적 파일 포맷의 지원을 위해 각기 다른 재생기가 요구되는 문제를 해결하였고 가치있는 여러 개념들을 소개하고 있지만, 몇 가지 한계점을 가진다. SMIL은 미디어 저작 도구와 재생기를 위한 데이터 교환 포맷일 뿐이며 이러한 개념을 HTML과 웹 브라우저에 적용하는 수단을 제공하지는 않는다. SMIL의 기능을 HTML과 웹 브라우저로 확장하기 위하여 제안된 HTML+TIME은 SMIL을 기초로 HTML에 시간성과 동기성을 부여한 것이다. HTML+TIME은 시간성과 상호작용성을 지원하기 위한 HTML의 확장을 포함하며, SMIL 1.0 표준안에 기술된 기능들을 지원하는 새로운 태그들을 가진다. 또한 SMIL의 시간성과 동기성 모델을 확장하여 웹 브라우저 도메인에 적합하게 하였다.

HTML + TIME에서는 timeline을 사용하여 시간성을 표현한다. timeline을 정의하는 방법에는 새로운 HTML 태그를 사용하는 방법과 HTML 컨테이너 요소에 새로운 속성을 추가하는 방법이 있다. 즉, "<t:par>" 태그와 "<t:seq>" 태그를 사용하여 병렬적 시간 동기화와 직렬적 시간 동기화를 지원할 수 있으며, 다른 방법으로 "timeline"이라는 속성에 "par" 또는 "seq"라는 값을 명시함으로써 HTML 요소들을 시간축상에 그룹화할 수 있다. 다음의 코드는 HTML + TIME을 사용하여 네 개의 이미지들을 일정 시간 간격으로 순서대로 디스플레이 하는 예이다.

```
<HTML>
<HEAD>
<STYLE>
.mytime {behavior : url(#default#time); }
</STYLE>
</HEAD>
<DIV CLASS="mytime" t : TIMELINE="seq">
<IMG SRC="slide1.gif" CLASS="mytime" t : DUR="2">
<IMG SRC="slide4.gif" CLASS="mytime" t : DUR="2"><BR>
<IMG SRC="slide5.gif" CLASS="mytime" t : DUR="2">
<IMG SRC="slide6.gif" CLASS="mytime" t : DUR="2">
</DIV>
</BODY>
</HTML>
```

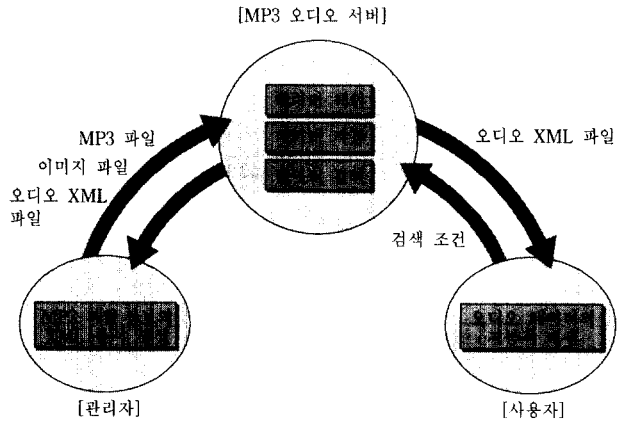
HTML + TIME이 가지는 미디어 객체 요소는 animation, audio, img, video, media, textstream이다. SMIL에서와 마찬가지로 미디어 객체 요소들은 시간적으로 동기화가 가능하도록 하는 속성들을 가진다. 이러한 속성에는 begin, dur, end 외에도 참조하는 요소의 시작 시간을 기준으로 상연을 시작할 수 있게 하는 beginWith 속성, 참조 요소의 종료 시간을 기준으로 상연을 시작하게 하는 beginAfter 속성, 참조 요소의 종료 시간에 상연을 종료하게 하는 endWith 속성 등이 있다.

본 연구에서는 MP3 파일 내의 오디오 데이터와 오디오 데이터에 대하여 시간적으로 동기화된 태그 정보들을 HTML + TIME 형식으로 웹 상에서 재생하도록 하였다.

4. MP3 태그의 XML 확장을 이용한 동기화된 재생 시스템 구성

XML 문서 형식을 통하여 MP3 파일의 태그 내에 기록되어 있는 정보들을 바탕으로 구성된 오디오 객체를 모델링하였고, 이러한 오디오 객체를 웹 상에서 시간관계성과 동기성이 표현되도록 렌더링하는 XSL 문서를 작성하였다. 또한 사용자로 하여금 원하는 오디오 데이터를 찾아낼 수 있도록 하는 검색 모듈을 설계하였다. 이와 같은 내용은 (그림 6)과 같이 통합되어 MP3 태그의 XML 확장을 이용한 검색과 동기화된 재생 시스템을 구성한다.

MP3 파일의 오디오 데이터와 관련 정보들을 XML 문서로 표현하는 방법과 오디오 데이터와 관련 정보들을 동기화하여 재생하는 방법을 5절과 6절에서 자세히 설명한다.



(그림 6) MP3 태그의 XML 확장을 이용한 검색과 동기화된 재생 시스템의 전체적 구성과 흐름

5. 태그가 있는 MP3 파일 정보의 XML 문서 표현

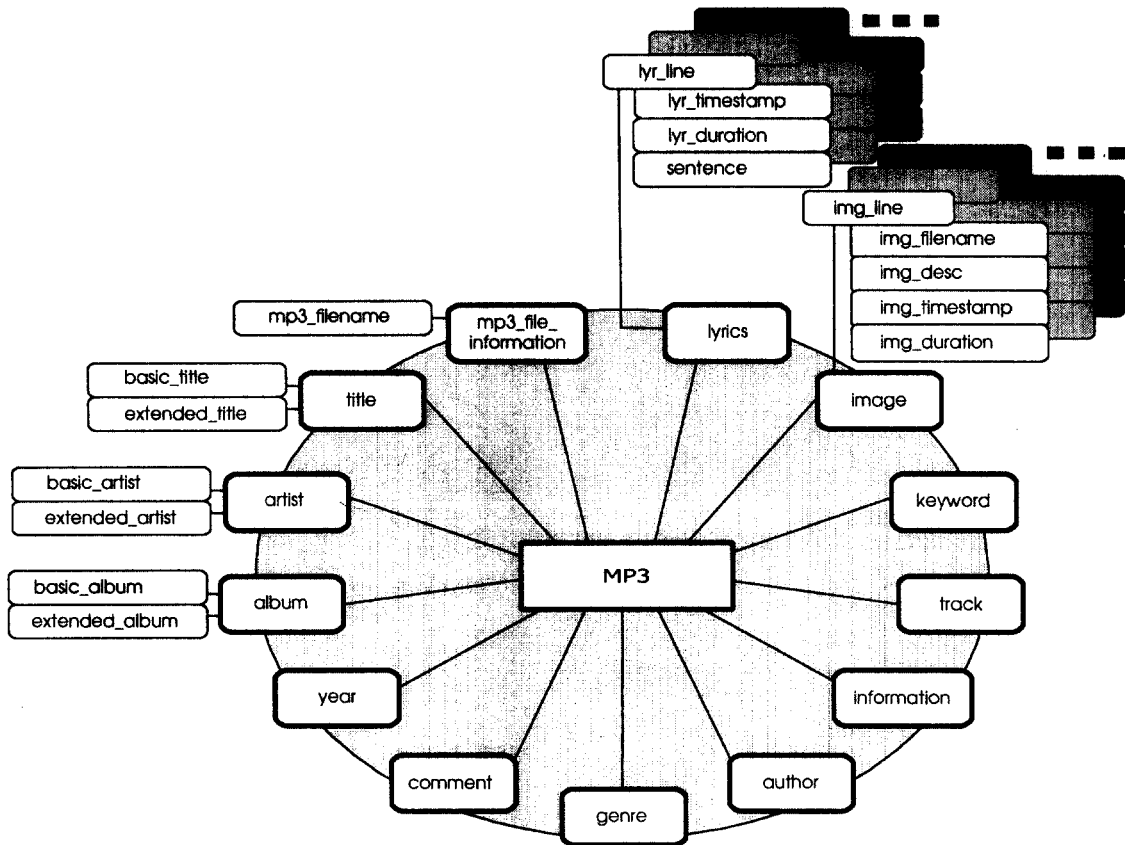
5.1 오디오 객체 설계

본 논문에서 다루는 데이터의 초기 형태는 오디오 데이터 부분과 그에 관한 추가 정보를 담은 태그 부분으로 구성된 MP3 포맷의 오디오 파일이다. 현재 ID3 v1 태그와 Lyrics3 v2 태그를 함께 사용하는 MP3 태깅 방법이 가장 널리 사용되고 있으므로, 본 연구에서는 이러한 방법으로 태깅된 MP3 파일들을 대상으로 하였다.

2절에서 설명한 바와 같이, ID3 v1과 Lyrics3 v2로 태깅된 MP3 파일은 오디오 스트림과 그에 관한 여러 추가 정보를 가진다. 즉, ID3 v1 태그에 곡명, 밴드나 가수의 이름, 그 앨범 이름, 발표된 해, 주석, 장르 정보를, Lyrics3 v2 태그에 확장된 곡명, 밴드나 가수의 이름, 앨범 이름과 일반적 정보, 관련 이미지 링크, 오디오 스트림에 시간적으로 동기화된 가사 정보를 포함한다.

이러한 MP3 파일로부터 정보를 추출하여 (그림 7)과 같은 오디오 객체를 설계하였다.

하나의 MP3 오디오 파일을 표현하기 위한 구성 요소는 13개이며, 'mp3_file_information', 'title', 'artist', 'album', 'year', 'comment', 'genre', 'author', 'information', 'track', 'keyword', 'image', 'lyrics'가 그것이다. 이 요소들은 각각 MP3 파일 정보, 곡명, 밴드나 가수의 이름, 이 곡이 포함된 앨범 이름, 발표년도, 주석, 이 곡이 속하는 장르, 저자, 일반 정보, 앨범 내 이 곡의 트랙번호, 사용자 검색의 편의를 위하여 관리자가 입력한 키워드, 오디오 데이터에 관련된 이미지 링크, 오디오 스트림에 시간적으로 동기화된 가사를 나타낸다. 곡명과 아티스트 이름, 앨범 이름과 같이 ID3 v1 태그와 Lyrics3 v2 태그에 공통적으로 기록되어 있는 내용들은 해당 요소에 하위 요소를 두어 표현한다.



(그림 7) MP3 파일 내 정보를 기반으로 한 오디오 객체

5.2 오디오 객체 표현을 위한 XML DTD 설계

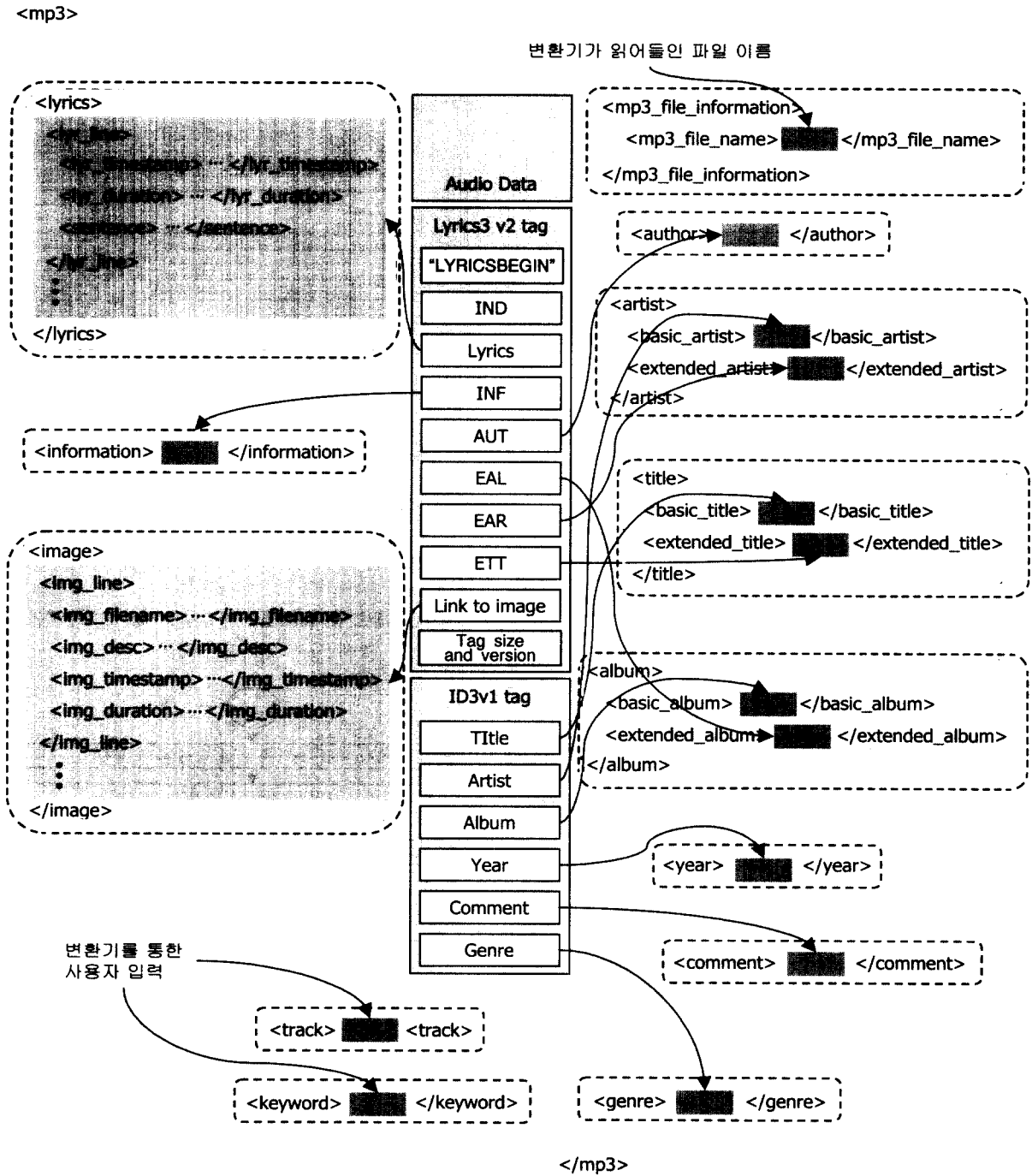
설계된 오디오 객체의 XML 표현을 위해 DTD를 사용하여 오디오 객체 내 데이터 형과 그 사이의 관계들을 정의하였다. 본 논문에서 설계한 오디오 객체를 표현하는 XML DTD는 하나의 MP3 오디오 파일을 표현하기 위한 구성 요소인 'mp3_file_information', 'title', 'artist', 'album', 'year', 'comment', 'genre', 'author', 'information', 'track', 'keyword', 'image', 'lyrics'의 13개 요소를 가진다. 'lyrics' 요소에는 'lyr_timestamp'와 'sentence'로 이루어진 'lyr_line'이라는 요소가 여러 번 나타날 수 있도록 설계되어 있는데, 이는 Lyrics3 태그의 LYR 필드가 가사를 "[CR][LF]"로 구분되는 여러 행의 텍스트로 표현하고 있음을 따른 것이다. 'image' 요소도 'lyrics' 요소의 경우와 비슷하다. 다음은 이와 같은 설계 내용에 따라 작성된 XML DTD 코드이다.

```
<!ELEMENT mp3 (mp3_file_information, title, artist, album, year,
comment, genre, author, information, lyrics, image, keywords, track)>
<!ELEMENT mp3_file_information (mp3_file_name)>
<!ELEMENT mp3_file_name (#PCDATA)>
<!ELEMENT title (basic_title, extended_title)>
<!ELEMENT artist (basic_artist, extended_artist)>
<!ELEMENT album (basic_album, extended_album)>
<!ELEMENT year (#PCDATA)>
<!ELEMENT comment (#PCDATA)>
```

```
<!ELEMENT genre (#PCDATA)>
<!ELEMENT author (#PCDATA)>
<!ELEMENT information (#PCDATA)>
<!ELEMENT lyrics (lyr_line*)>
<!ELEMENT lyr_line (lyr_timestamp?, sentence?)>
<!ELEMENT lyr_timestamp (#PCDATA)>
<!ELEMENT sentence (#PCDATA)>
<!ELEMENT image (img_line*)>
<!ELEMENT img_line (img_filename?, img_desc?, img_timestamp?)>
<!ELEMENT img_filename (#PCDATA)>
<!ELEMENT img_desc (#PCDATA)>
<!ELEMENT img_timestamp (#PCDATA)>
<!ELEMENT keywords (keyword*)>
<!ELEMENT keyword (#PCDATA)>
<!ELEMENT track (#PCDATA)>
```

5.3 태그가 있는 MP3 파일정보의 저장을 위한 XML 문서 생성

태그가 있는 MP3 파일의 정보를 XML 문서로 표현하여 저장하는 부분은 독립적으로 실행되는 윈도 응용프로그램으로 구현되었다. 이 프로그램은 MP3 파일을 불러들여 태그 내용을 읽어 화면에 보인다. 그 다음 읽어들이는 태그 내용과 사용자가 입력한 내용을 5.2절에서 설계한 XML DTD가 정



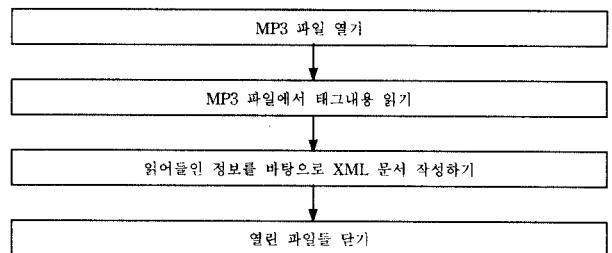
(그림 8) MP3 파일 내 태그 정보의 XML 기록

의한 형태의 XML 문서로 저장한다.

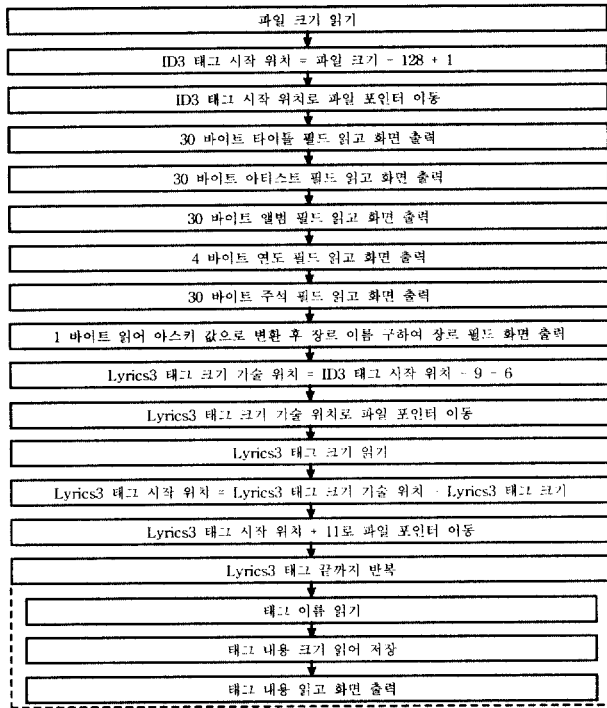
MP3 파일 내 ID3 v1 태그와 Lyrics3 v2 태그의 각 필드 내용들은 (그림 8)과 같이 XML 문서로 출력된다.

이 작업은 (그림 9)와 같은 순서로 수행된다.

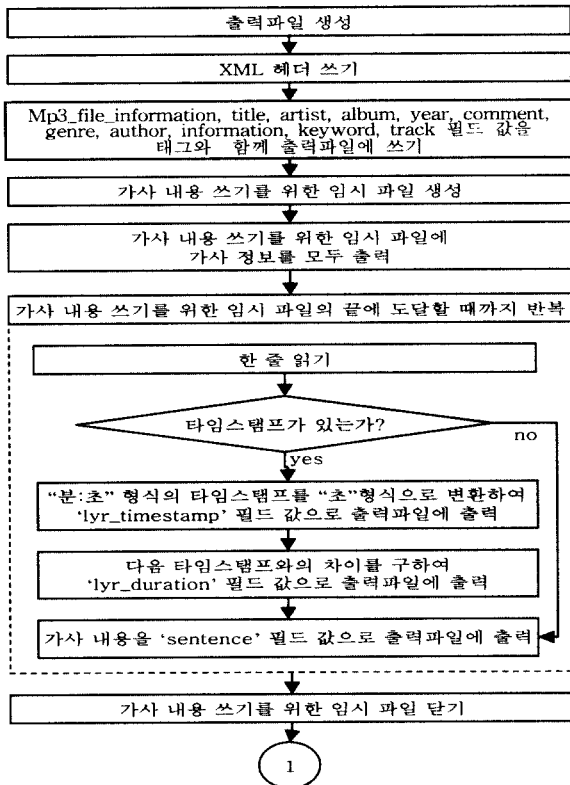
(그림 9)의 각 단계는 사용자의 메뉴 선택에 의해 시작된다. (그림 10)은 MP3 파일에서 태그 내용을 읽어들이는 단계를, (그림 11)은 읽어들이는 정보를 바탕으로 XML 문서를 생성하는 단계를 자세히 설명한다.



(그림 9) MP3 파일 정보의 XML문서기록 진행과정

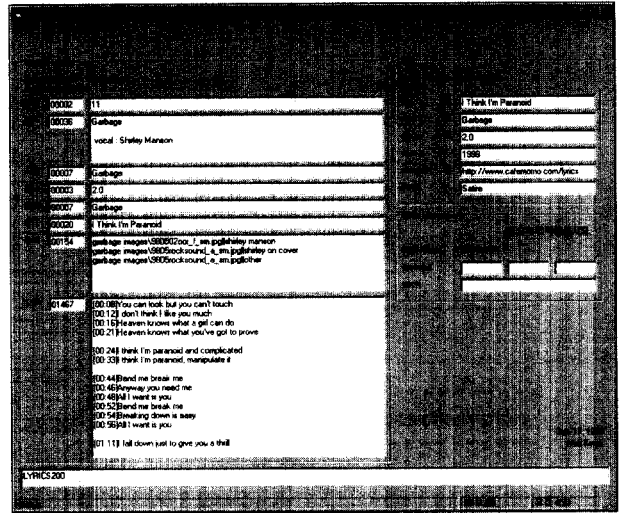


(그림 10) MP3 파일 내 ID3 v1 태그와 Lyrics3 v2 태그 내용 읽는 과정



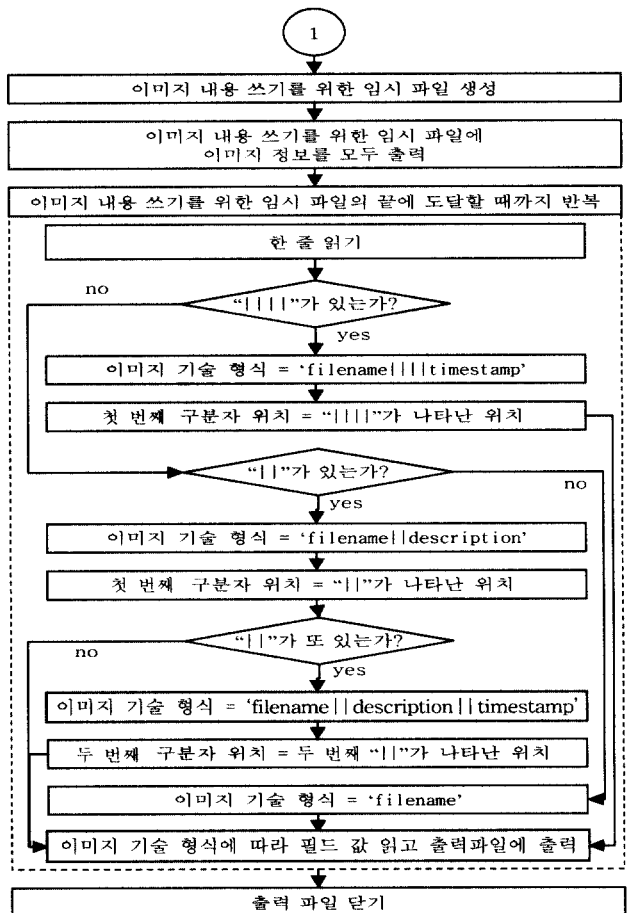
(그림 11) 읽어들이는 MP3 태그 정보를 XML 문서에 쓰기

(그림 12)는 MP3 태그 정보를 XML 문서로 저장하는 응용프로그램의 사용자 인터페이스를 보인다.



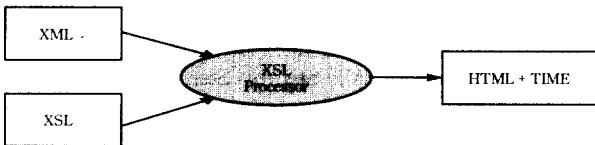
(그림 12) MP3 파일로부터 태그 내용을 읽어들이는 화면

생성된 XML 문서와 MP3 파일들은 서버에 저장되며 HTTP 프로토콜을 통해 사용자에게 의해 접근된다.



6. XML 문서로 표현된 오디오 데이터의 동기화된 재생

오디오 데이터를 표현하는 XML 문서에 포함된 가사 정보와 이미지 정보는 오디오 스트림에 대한 시간관계를 가진다. 또한 이미지 정보에는 이미지 파일 이름, 활성화 시작 시간, 활성화 기간의 정보와 함께 이미지에 대한 설명 정보가 포함되는데, 이미지에 대한 설명 정보는 해당 이미지의 활성화 사건이 발생함과 동시에 활성화된다는 동기성을 가진다. 본 논문에서는 이와 같은 시간관계성과 동기성을 웹 상에서 표현하기 위한 방법으로 HTML+TIME이라는 웹 문서 형식을 선택하였다. 저장된 XML 문서를 HTML+TIME 형식으로 사용자의 웹 브라우저에 렌더링 하기 위한 XSL 문서를 작성함으로써 XML 문서로 표현된 오디오 데이터의 동기 재생 모듈을 구현하였다. (그림 13)은 이와 같은 내용을 그림으로 나타낸 것이다.



(그림 13) XSL에 의한 XML의 HTML+TIME 형식 렌더링

6.1 시간 모델

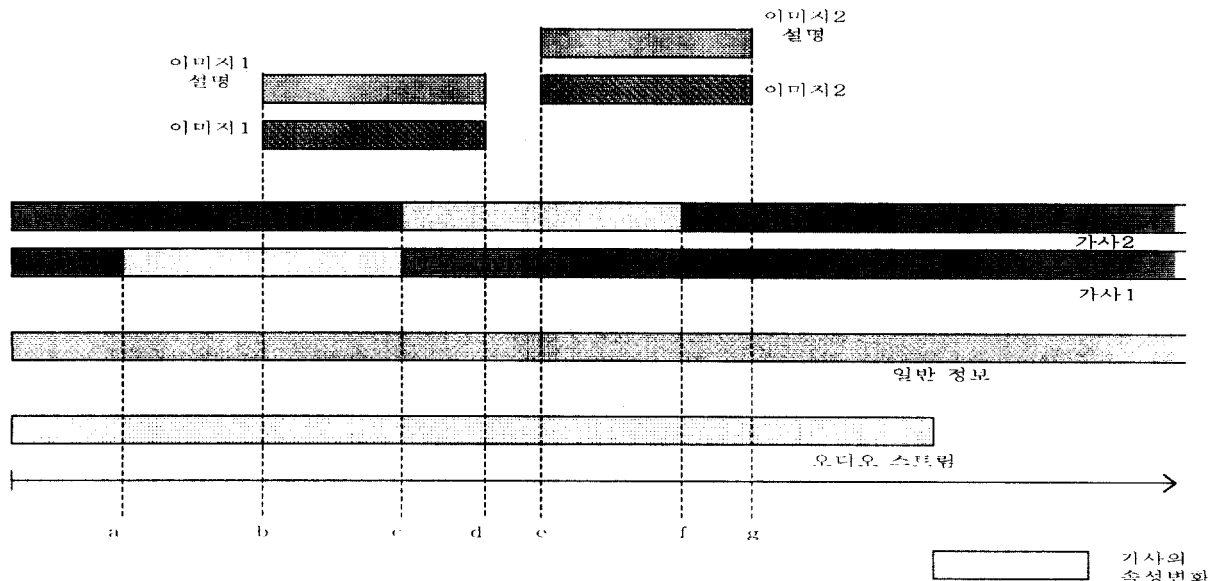
동기화된 가사와 이미지 정보의 시간성을 표현하기 위해서는 이러한 요소들이 활성화되는 시작 시간을 표시하는 시간축 모델의 특징이 필요하며, 어떤 이미지가 나타났을 때 그에 대한 설명을 화면에 나타내기 위해서는 사건 기반 시간 모델의 특징이 필요하다. 본 논문에서는 전형적인 시간축 모델과 사건 기반의 시간 모델을 혼합한 HTML+TIME 시간 모델을 사용한다[11].

(그림 14)은 오디오 객체를 시간 축 위에 표현한 것이다. 오디오 객체를 기술한 XML 문서가 사용자의 브라우저에 표시됨과 동시에 오디오 스트림과 일반 정보들, 가사가 화면에 표시된다. '가사1'은 문서의 화면 표시 후 'a'의 시간이 흐른 후 글자색과 배경색이 변화하여 'c-a'의 시간동안 변화한 속성이 적용된 채 지속된다. '이미지1'은 문서가 화면에 표시된 후 'b'의 시간이 지난 후 화면에 나타나 'd-b'의 시간이 지나면 사라진다. '이미지1 설명'은 '이미지1'과 동시에 화면에 나타나 동시에 사라진다.

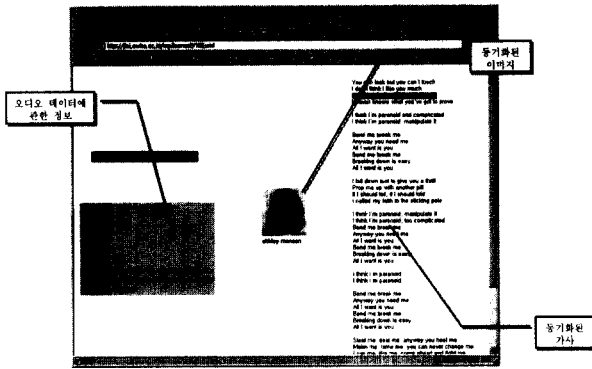
6.2 재생 형식 설계

오디오 객체를 구성하는 요소들 중 'artist', 'title', 'album', 'year', 'comment', 'genre', 'author', 'information', 'track'의 요소는 시간성을 가지지 않으며, 오디오 데이터의 디스플레이 시작시간부터 활성화되어 종료시간까지 지속된다. 'comment' 요소에는 관련된 웹 페이지의 주소가 저장되어 있는 경우가 많으므로 링크 속성을 준다. 'mp3_file_information' 요소의 'mp3_file_name' 요소가 가리키는 MP3 오디오 파일은 문서의 디스플레이 시작시간에 재생이 시작된다. 'lyrics', 'image' 요소의 정보는 시간적으로 오디오 스트림에 대하여 동기화된 정보들로 시간성을 가진다. 'lyrics'의 경우 각 'lyr_line' 요소의 'sentence'에 저장된 내용은 'lyr_timestamp'에 기술된 시간에 배경색과 글자색의 텍스트 표현 속성이 변화하고, 'lyr_duration'에 기술된 시간이 지나면 원래의 속성으로 돌아간다. 'image'의 경우 각 'img_line' 요소의 'img_filename'이 가리키는 이미지 파일이 'img_timestamp'에 기술된 시간이 활성화되어, 'img_duration'에 기술된 시간동안 지속된다. 'img_line' 요소의 'img_desc' 내용은 해당 이미지가 활성화되어있는 동안 활성화된다.

이러한 형식으로 오디오 데이터가 재생될 때의 화면 구성은 (그림 15)와 같다.



(그림 14) 시간축 위에 표현한 오디오 객체



(그림 15) 재생시의 화면 구성

7. 결 론

본 논문에서는 MP3 파일 내에 저장된 정보들을 효율적으로 활용하여 관리, 검색하고 그 시간관계성을 살려 재생할 수 있는 동기화 모델을 제시하였다. 이를 위하여 MP3 파일 내의 태그 정보를 읽어들이고 이를 기반으로 XML 문서를 생성하는 프로그램과 사용자에 의해 선택된 오디오 데이터의 시간관계성과 동기성 표현이 가능한 웹 기반 재생 모듈을 설계하고 구현하였다.

MP3 파일 내의 태그 정보들을 바탕으로 XML 문서를 생성하는 방법을 설계하고 구현함으로써, 시간성을 가지는 요소가 포함된 멀티미디어 객체의 XML 모델을 제시하였다. 기존의 MP3 오디오 서버를 통해 MP3 파일을 전달받아 재생하는 경우, 이미 해당 오디오에 관한 많은 정보가 오디오 데이터와 함께 저장되어 있고 가사 및 이미지들이 시간 동기화 정보와 함께 작성되어 있음에도 불구하고, 클라이언트는 해당 정보들을 바탕으로 원하는 오디오를 검색하거나 오디오 데이터를 재생하면서 가사나 이미지를 함께 볼 수 없었다. 그러나, 본 논문에서 제안한 시스템은 이렇게 태그 내용이 무시되는 문제들을 해결한다. MP3 파일 태그 내의 추가적인 오디오 정보들은 XML 파일로 데이터 서버에 저장되고 그 내용을 기반으로 검색될 수 있다. 또한 사용자가 요구한 XML 문서의 재생 시 그 안에 포함된 요소들의 시간관계를 HTML+TIME가 제시하는 방식으로 표현하도록 설계된 XSL 문서를 통해, 오디오 데이터에 동기화된 가사 및 이미지 정보를 시간 동기성을 가지고 참조할 수 있다.

본 논문에서 설계한 오디오 데이터 표현을 위한 XML 스키마의 정의를 확장하여 시간성 외에 데이터 재생시 각 요소의 화면 위치를 지정하는 위치성을 표현할 수 있게 하고, 매체 사이의 상호작용성을 보다 강력하게 지원하도록 하여 능동적 멀티미디어 데이터 표현 모델을 만들 수 있다. 사용자 상호작용성과 오디오 데이터를 저장하는 서버의 데이터 검색 기능 강화 등이 향후의 연구방향이다.

참 고 문 헌

[1] 고영근, 최윤철, "멀티미디어 정보의 효율적인 검색을 위한

하이퍼미디어 시스템의 설계와 구현", 한국통신학회논문지, 제18권 제8호, 1993.

[2] 김기욱, 김형주, "비디오 주석 시스템의 설계 및 구현", 정보과학회논문지(B), 제24권 제6호, 1997.

[3] 이의택, 박종훈, 호요성, 김정홍, 유경렬, "VOD 시스템 최신 기술 동향", 전자공학회지, 제21권 제10호, pp.996-1009, 1994.

[4] 이택경, 홍진석, 고영근, 최윤철, "하이퍼미디어 시스템을 위한 HyTime 엔진 및 SGML 검색 엔진의 개발," 정보과학회논문지(B), 제23권 제8호, 1996.

[5] DAVIC, "Revision 3.0 of DAVIC 1.0 Specification," June, 1995.

[6] J. D. Koegel, L. W. Rutledge, J. L. Rutledge and C. Keskin, "HyOctane : A HyTime Engine for an MMIS," ACM Multimedia93, Aug. 1993.

[7] J. Frankel, D. Greely and B. Sawyer, "MP3 POWER!," Muska & Lipman, 1999.

[8] L. Hardman, D. C. A. Bulterman and G. Van Rossum, "The Amsterdam Hypermedia Model, Commun," ACM, Vol.37, No.2, pp.50-62, Feb. 1994.

[9] L. Hardman, D. C. A. Bulterman and G. Van Rossum, "Links in Hypermedia : the Requirement for Context," Proceedings of Hypertext '93, pp.183-191, Nov. 1993.

[10] Microsoft Press, "XML Step by Step," Microsoft Press, June, 2000.

[11] P. Strnad, "Tag format specs," November, 2000.

[12] S. Hacker, "MP3 : The Definitive Guide," O'Reilly, 2000.

[13] W3C, "Synchronized Multimedia Integration Language (SMIL) Boston Specification," Aug. 1999.

[14] W3C, "Timed Interactive Multimedia Extensions for HTML (HTML+TIME)," Sep. 1998.

[15] Y. K. Ko, S. H. Jung and Y. C. Choy, "Design and Implementation of an Integrated Information Space for Hypermedia System," Proc. of the 17th Int. Conf. on ITI, pp. 489-494, June, 1995.

[16] Y. K. Ko and Y. C. Choy, "Modeling for Interactive Presentation and Navigation of Time-dependent Multimedia Information," Proc. of Fifth IEEE Workshop on FTDCS, pp.143-151, 1995.

곽 미 라



e-mail : mirakwak@computer.org
 1998년 이화여자대학교 컴퓨터학과(공학사)
 2000년 이화여자대학교 컴퓨터학과(공학석사)
 2000년~현재 이화여자대학교 컴퓨터학과 박사과정
 관심분야 : 멀티미디어, 지식공학

조 동 섭



e-mail : dscho@mm.ewha.ac.kr
 1979년 서울대학교 전기공학과(공학사)
 1981년 서울대학교 전기공학과(공학석사)
 1986년 서울대학교 컴퓨터공학과(공학박사)
 1985년~현재 이화여자대학교 교수
 1996년~1997년 University of California, Irvine Visiting Scholar

관심분야 : 컴퓨터구조 및 인터넷공학, 컴퓨터비전, 컴퓨터그래픽스, 가상교육