

## 웹 기반 멀티미디어 프로그래밍 동향

한국전자통신연구원 김지용·고동일\*·김두현\*

### 1. 서론

웹과 관련한 기술의 성장과 활용은 가파르게 증가하고 있고, 이는 인터넷의 보급을 촉발하여 전세계 인터넷 인구가 2002년에는 2억명, 2005년에는 3억 5천만명에 이를 것으로 추정되고 있다. 이러한 현상의 배경에는 기술적으로 웹 하부 구조가 갖는 단순성과 개방성을 들 수 있다. 인터넷의 어느 곳의 어느 정보든 접근할 수 있는 표준화된 URL과 사용자가 쉽게 숙달하고 이용할 수 있는 HTML 표현 그리고 이들 내용을 전송할 수 있는 HTTP 표준 프로토콜의 3가지 요소의 개방형 특성이 사용자 및 개발자의 확대를 유도하였다.

이러한 구조는 기본적으로 단순한 텍스트 또는 그래픽을 포함한 텍스트 만을 다루도록 설계되어 왔다. 그러나 오늘날 많은 일반 사용자나 응용 서비스 개발자 및 콘텐츠 제공자들은 텍스트뿐만 아니라 오디오 및 동영상 데이터를 전송하거나 받을 수 있는 서비스를 요구한다. 이런 멀티미디어 서비스를 제공하기 위해서는 새로운 형식의 프로토콜이나 언어를 필요로 하게 된다. 프로토콜의 경우 IETF에서 제공한 RTP, RTP Control Protocol(RTCP)[2, 3], 및 RTSP[4] 등이 있다. 새로운 언어의 경우 두 가지 방향의 노력이 있는데, XML과 관련하여 스크립트 언어로 정의하고자 하는 노력과 멀티미디어의 복잡한 표현을 위하여 프로그래밍적인 전혀 새로운 언어를 정의하려는 노력이 있다. 각각의 대표적인 예로

SMIL[1]과 VRML[6]이 있다.

웹이나 CD-ROM 타이틀에서 미디어 간에 동기를 맞추어 프리젠테이션을 하기 위하여 제안된 SMIL은 XML의 한 어플리케이션으로 이미 RealNetworks사의 RealPlayer 등의 프로그램에서 멀티미디어 프리젠테이션 서비스를 위한 기본 언어로 일반화 되어 있다. 현재는 1.0버전의 발표 이후 다른 언어들과의 상호 사용을 위해 두 번째 워킹그룹이 활동 중이다.

1995년도에 발표된 VRML은 기존의 웹이 텍스트 기반의 정적인 환경인 제약을 극복하고자 한다. SMIL과는 다르게 3차원의 멀티미디어 객체의 애니메이션을 위한, 상호 작용적인(interactive) 언어인 VRML은 SMIL이 W3C의 권고안인 것처럼 1997년 ISO 표준으로 인정되어 각광을 받고 있다. 역시 XML과 접목하여 타언어와 상호 사용을 위한 내용을 연구 중이다.

본문에서는 멀티미디어 프리젠테이션의 구조적 두 가지 노력의 대표 예인 SMIL과 VRML을 살펴본다.

### 2. SMIL[1, 5]

웹이나 CD-ROM 타이틀에서 미디어 간에 동기를 맞추어 프리젠테이션을 하기 위한 선언적 언어를 필요로 하게 되었다. 1996년 10월에 열린 워킹샵에 따라, W3C에서는 동기화 멀티미디어 워킹그룹을 1997년 3월에 설립하였고, 이듬해 6월에 SMIL이라는 선언적 언어의 1.0 버전을 발표하게 되고, 이 선언적 언어는 W3C의 동기화 멀티미디어 프리젠테이션을 위한 권고안이 된다.

\* 정회원

이미 이 SMIL 1.0의 표준에 따르는 상용화 플레이어들이 존재하고, W3C에서 무료로 다운로드 받아 실행시켜 볼 수 있다.

두 번째 동기화 멀티미디어 워킹그룹 SYMM이 1999년 2월에 설립되었고, 그 그룹에서는 첫 번째 워킹드래프트("Boston")를 1999년 8월에 발표하였다. SMIL Boston에서는 SMIL의 모듈화 아웃라인을 잡았는데, 그 모듈에 대해서는 밑에서 살펴보기로 한다.

여기에서는 SMIL 1.0의 내용을 살펴보고, SMIL "Boston"에서 제안하는 바를 알아본다. 다만, SMIL의 개념을 소개하는 것은 비교적 간단한 일이지만 문법 차원까지 자세히 설명하는 것은 본 내용의 범주를 벗어나므로 생략하기로 하고, SMIL 규격서에 나와있는 예제 등을 인용하여 SMIL의 기본적인 기능들을 중심으로 기술하기로 한다.

## 2.1 SMIL파일의 기본 형식

SMIL 1.0 표준에서도 말하듯이 SMIL 파일은 미디어 간의 시간적 동기를 표현할 수 있고, 화면상에서 공간 정보를 표현하여 배치를 설정할 수 있다. 덧붙여 미디어에 하이퍼 링크를 추가하여 결합할 수도 있다. 기본적으로 이런 정보들을 SMIL 파일 안에 포함시켜야 하며, 형식적인 구조는 표제어 부분(header section)과 본문 부분(body section)으로 나눌 수 있다. 그 모양은 다음과 같다.

```
<smil>
  <head>
    .all header information...
  </head>
  <body>
    ...all body information...
  </body>
</smil>
```

일반적으로 표제어 구간에서는 프리젠테이션 동안 변하지 않는 정적인 정보를 나타내게 된다. 각각의 영역에 대한 정보라든지 미디어 파일이 재생될 때 배치에 대한 지정을 해 줄 수 있다. 반면 본문 부분에서는 시간에 따라 프리젠테이션을 결정하는 동적인 정보들을 포함한다. 또한, 미디어 파일과 하이퍼 링크를 연결하는 일련의 일

도 본문 부분에서 지정된다.

표제어 구간에서는 멀티미디어 프리젠테이션 상태 패널에서 나타나는 작가, 제목, 그리고 저작권 정보를 제공할 수 있다. 이를 위해 당신은 다음처럼 이름과 콘텐츠 속성을 가진 <meta> 표시기호를 사용한다.

```
<head>
  <meta name="author" content="Jane Morales"/>
  <meta name="title" content="Multimedia My Way"/>
  <meta name="copyright" content="(c)1998 Jane Morales"/>
</head>
```

본문 부분에서 멀티미디어 프리젠테이션 안의 원시 미디어 파일은 <src> 표시기호를 가진 SMIL파일 안에서 지정된다. 그 표시기호는 원시 미디어 파일 유형과 위치를 제시한다.

```
<audio src="http://server company com/audio/first.snd"/>
```

이 표시기호의 처음 부분은 원시미디어 파일의 유형을 지정한다. 그 내용에는 오디오가 있을 수 있다. 예를 들면 RealAudio, WAV, SND, AIFF, 또는 AU 형식 등이 될 수 있는데, 시작은 <audio> 표시기호로 시작한다. <video>로 시작되는 비디오 파일은 RealVideo, AVI, QuickTime MOV, MPEG. 비디오 등이 있는데, SMIL 원시 표시기호는 애니메이션, VRML, 또는 지속적인 비디오 형식의 동작을 보여주는 다른 파일을 위해 사용될 수 있다. 이미지 역시 미디어 형식으로 포함될 수 있다. GIF와 JPEG 같은 정지 이미지들을 위해 "image"를 사용한다. 텍스트 역시 미디어 파일이 될 수 있다. 간단한 텍스트 파일과 스트림 텍스트 파일을 위해 "text"를 사용한다. 그 외에 정확한 파일 형식을 알 수 없거나 멀티미디어 동기화 파일과 같은 다른 파일 유형을 위해 ref를 지정한다.

## 2.2 미디어 파일의 실행

미디어 파일을 실행시키는 형식은 기본적으로 동시에 실행시키거나 순차적으로 할 수 있다. 연속적으로 클립들을 실행시키기 위해서, <seq>

표시기호를 사용한다. 다음 예에서 첫번째 클립이 끝났을 때 두번째 클립이 시작된다.

```
<seq>
  <audio src="audio/newsong.wav"/>
  <audio src="audio/oldsong.asf"/>
</seq>
```

또는, <par> SMIL 표시기호를 사용함으로써 동시에 두 개 이상의 클립들을 실행시킬 수 있다. 예를 들면 다음은 동기화 된 WAV, MOV, 그리고 AAF 클립들의 집단을 실행시킨다.

```
<par>
  <audio src="audio/newsong.wav"/>
  <video src="video/newsong.mov"/>
  <ref src="lyrics/newsong.asf"/>
</par>
```

아래와 같이 제시된 지연, 클립 시작, 그리고 클립종료시간을 지정함으로써 클립을 언제 시작하는지 그리고 얼마동안 그것을 실행시키는 것을 위한 오프셋(offset)이다. 이런 시간 요소들은 다음 형식을 사용한다.

- hh:mm:ss.xy

hh는 시간, mm은 분, ss는 초, x는 수십번째의 초, 그리고 y는 수백번째의 초를 나타낸다. 다음과 같은 지연 속성은 프리젠테이션 시간 라인 안에 있는 특정 포인트에 클립 시작을 지정하기 위해 사용될 수 있다. "begin" 이나 "end" 속성의 값에 시간을 대입함으로써 미디어의 실행 시작 시각과 종료 시각을 정할 수 있다.

- delay="20.5s"

<par> 그룹화에서 이것은 20.5초까지 클립시작을 지연시킨다. <seq> 그룹화에서 이것은 클립사이에 20.5초의 공백시간을 더한다. 지연 속성은 이미지, 오디오, 비디오, 또는 다른 어떤 파일 유형을 위해 작동한다.

다음과 같은 클립 시작과 클립 종료 속성들은 playback이 시작하고 끝나는 클립의 내부 시간 마크를 지정한다.

- clip-begin="10.5s" clip-end="50.7s"

이 예에서, 클립은 그것의 평소 시작점에서 보다 그것의 내부 10.5초에서 실행을 시작한다. 총 40.2초 동안 실행하면서 50.7초에 다다랐을

때 그것은 실행을 멈춘다.

<par>와 <seq>를 중첩하여 사용함으로써 훨씬 복잡한 일련의 연출을 지정할 수 있다. 여기서 상세한 언급은 하지 않겠다.

### 2.3 하이퍼링크의 결합

SMIL에서 또 하나 빠뜨릴 수 없는 것은 하이퍼링크와 미디어 파일의 결합이다. <a>나 <anchor> 표시기호를 가지고 저작자는 미디어 파일에 하이퍼 링크를 삽입할 수 있다.

<a> 표시기호는 HTML 4.0 [HTML40]의 <a> 원소와 매우 유사한 기능을 가지고 있다. SMIL은 하이퍼링크를 선택했을 때, 원천의 시간적 행위를 제어하는 "show" 속성을 덧붙였다. "show"의 속성에 따라 원천이 사라지든지, 원천은 그대로이고 목적이 되는 미디어 파일이 새로 뜨든지, 원천이 되는 미디어는 정지되는지를 결정할 수 있다.

```
<a href="http://www.cwi.nl/somewhereelse.smi">
  <video src="rtsp://foo.com/graph.mf"
    region="L_window"/>
</a>
```

<anchor>는 <a> 표시기호의 기능적인 제한을 극복하기 위하여 확장한 개념의 하이퍼링크를 제공한다. 우선, "href" 속성을 사용하여, 연결의 목적지를 미디어 파일의 공간적 또는, 시간적인 일부분에 한정할 수 있다.

```
<video src="http://www.w3.org/CoolStuff">
  <anchor href="http://www.w3.org/AudioVideo"
    begin="0s" end="5s"/>
  <anchor href="http://www.w3.org/Style"
    begin="5s" end="10s"/>
</video>
```

또한, anchor 원소는 "id" 속성을 이용하여, 매체 객체의 일부분을 연결의 목적지로 만들 수 있도록 한다.

Presentation A:

```
<a href="http://www.cwi.nl/inm/presentationB
  #um">
  <video id="graph" src="rtsp://foo.com/graph.
    mf" region="L_window"/>
```

</a>

```
Presentation B:
<video src="http://www.w3.org/CoolStuff">
  <anchor id="joe" begin="0s" end="5s"/>
  <anchor id="tim" begin="5s" end="10s"/>
</video>
```

## 2.4 대체 선택들 사이에서 전환 및 미디어 파일의 배치

<switch> SMIL 표시기호를 가지고 저작자는 시청자에게 프리젠테이션을 위한 다양한 옵션을 지정할 수 있다. <switch> 표시기호는 어느 정도의 선택들을 지정한다. 멀티미디어 프리젠테이션은 트리거(trigger)를 만족시키는 첫번째 조건에 기반을 두어 연출될 것이다. 예를 들어, 만일 트리거가 시스템 언어라면, 각 스위치 문장은 다른 언어를 위해 파일의 위치 목록을 만든다. SMIL 파일은 그것의 언어 우선순위 설정에 따라 파일을 선택한다.

```
<switch>
  <type src="location1" trigger="value1"/>
  <type src="location2" trigger="value2"/>
</switch>
```

각각의 SMIL 레이아웃 창은 왼쪽 위에서 아래로 측정된 단순 좌표 시스템을 사용하여 설계될 수 있다. 0픽셀을 디폴트 값으로 가지고 오프셋 측정치들은 픽셀단위로 표시된다.

```
<head>
  <layout>
    <region id="videoregion" left="103"
      top="20" height="180" width="240"/>
    <region id="textregion" left="103"
      top="200"/>
  </layout>
</head>
```

이 예에서 <region> 표시기호는 "videoregion"과 "textregion"으로 불리는 두 영역을 만든다. 두 영역은 가장 중요한 RealPlayer 창의 왼쪽 끝에서 103픽셀 만큼 옮긴 오프셋이다. 비디오 영역은 창 맨 위에서 디스플레이한다. 그러나 텍스트 영역은 200픽셀 아래에 있다. 0픽셀은 디폴트이기 때문에 당신은 비디오 영역의 "top" 속

성없이 같은 결과를 얻는다. 비디오 영역이 또한 픽셀들의 높이와 넓이를 지정한다. 이것은 클립의 암호화된(encoded) 크기를 고려하지 않고 비디오 클립을 그 크기들로 제한한다(만일 부호화된 비디오가 그 영역보다 크다면 클립화가 일어난다). 높이와 크기가 주어지지 않는다면 텍스트 영역에서처럼 클립은 그것의 보통의 크기와 부호화된 크기에 디스플레이 한다.

SMIL 레이아웃에 각 영역을 위한 배경색을 지정할 수 있다.

```
<head>
  <layout>
    <region id="videoregion" background-color="aqua"../>
    <region id="textregion" background-color="#C2EBD7"../>
  </layout>
</head>
```

하나의 영역 클립이 활동적일 때, 그 영역 클립은 배경색을 덮고 숨긴다. 그러나 만일 클립이 투명한 속성을 가지고 있다면 그 배경색이 투과하여 보인다. 그 색값을 위해 다음의 미리 정의한 색 이름들 중의 하나와 마찬가지로 HTML에 의해 지원되는 빨강, 초록, 파랑, 16진 값(#RRGGBB)을 사용할 수 있다. 여기에 그들이 대응하는 16진 값으로 목록을 만들었다.

## 2.5 SMIL BOSTON

SMIL "BOSTON"은 두 가지 디자인 목표를 가지고 제작되었다. 첫번째는 SMIL 1.0 스펙에서는 부족한 대화형 멀티미디어 프리젠테이션을 제작할 수 있도록 저자가 미디어의 시간적 공간적 특성을 지정이 가능하다. 또한, 다른 XML 기반의 언어(예를 들면, XHTML)와 상호 사용이 가능하도록 제작하였다. 이러한 내용은 modularization과 profiling의 개념으로 한다.

Modularization은 언어의 기능을 의미적으로 연관된 원소의 집합으로 나누는 것을 말하고, Profiling은 어떤 문제를 해결하기 위해 이런 기능들을 합치는 것을 말한다. 그래서, SMIL을 모듈화 하여 나누면 다음과 같다.

Animation 모듈 - animate, set, animateMotion, animateColor

Content Control 모듈 - switch  
 Layout 모듈 - layout, root-layout, region  
 Linking 모듈 a, area  
 Media Object 모듈 - ref, animation, audio, img, video, text, textstream  
 Metainformation 모듈 - meta  
 Structure 모듈 - smil, head, body  
 Timing and Synchronization 모듈 - par, seq, excl  
 Transition Effects 모듈 - TBD

이렇게 모듈화 한 후에 실제로 응용에 이용하기 위해 프로파일링을 하는데, 이런 일련의 작업은 SMIL을 다른 XML 기반의 언어들과 같이 사용하기 위해 필요하다. SMIL BOSTON에서는 몇 가지 프로파일링을 제시하고 있다. 이렇게 작성된 프로파일은 두 개의 언어가 복합된 새로운 형태의 언어가 되는데, 'HTML + SMIL'의 프로파일은 XHTML과 더불어 눈여겨 볼 만하다.

SMIL "BOSTON"은 기존 SMIL에서 확장된 표시기호들을 모듈별로 설명하고, 몇 가지 프로파일을 예시로 제공하여 저작자가 프로파일을 어떻게 구성해야 하는지 보여주고 있다. 자세한 내용은 참고문헌을 찾아보기 바란다.

### 3. VRML[6 - 9]

#### 3.1 VRML의 개요

WWW(World-Wide-Web)의 출현으로 인터넷의 확산이 급속화 되자 기존의 컴퓨터 그래픽스 분야의 연구자들은 웹의 텍스트 기반의 정적인(static) 환경을 좀 더 상호 작용적인(inter-active) 3차원 환경으로 확장하는 것에 대한 연구를 진행하였다. 이에 1995년에 VRML 1.0이 처음 세상에 발표되었다. VRML은 그 후 웹의 폭발적인 유행과 함께 인터넷 상에서의 3차원 정보 표현의 표준으로 자리잡으며, 3차원 정보 모델링, 과학 정보 시각화, 3차원 설계 정보 표현 등에 많이 사용되었다. 이후 1997년 12월에 제정된 VRML97은 ISO 표준으로 인정되어 명실상부한 인터넷 3차원 정보 표현의 표준으로 자리잡아가고 있다. 현재 VRML은 Web3D 컨소시엄을 중심으로 기존의 VRML을 인터넷 상에서의 새로운 정보 표현 표준인 XML(eXtended Markup Language)과 접목한 X3D의 표준제정을 눈앞에

두고 있다.

VRML을 보기 위해서는 별도의 VRML브라우저가 필요하다. VRML브라우저는 기존의 일반 웹 브라우저에 일종의 플러그인 방식으로 설치되며, CosmoPlayer나 Blaxxun Contact 등이 유명한 VRML브라우저이다.

#### 3.2 VRML File 문법

VRML의 초창기의 목적은 텍스트 언어로서 3차원 가상환경을 웹 상에서 손쉽게 표현하는 것이었다. VRML의 명칭의 유사성에서도 알 수 있듯이 VRML은 초창기에 HTML을 의식하여 만들어졌다. 각각의 가상환경은 하나 이상의 VRML 파일로 이루어지게 되며 이는 ".wrl" (world의 의미)의 확장자를 가지게 된다. 또한 웹에서는 "model/world"의 MIME 타입을 통해 전달되게 된다. VRML 파일은 다음과 같은 내용들을 주로 가지게 된다.

- VRML 헤더
- 주석(Comment)
- 노드(Node)
- 필드(Field), 필드 값(Field value)

##### 3.2.1 VRML 헤더

모든 VRML파일은 반드시 VRML 헤더를 가져야 한다. VRML 헤더는 반드시 파일의 첫번째 줄에 위치해야 하며, 다음과 같이 기술된다

```
#VRML V2.0 utf8
```

위의 VRML 헤더의 내용은 다음과 같은 것을 나타낸다.

- 이 파일은 VRML 파일이다
- 이 파일은 VRML 2.0 명세와 호환된다(사실상 VRML2.0과 VRML97은 같은 것으로 볼 수 있다).
- 이 파일은 UTF-8 문제집합을 사용하고 있다.

##### 3.2.2 주석(Comment)

일반적인 프로그래밍언어에서와 같이 주석은 실제 파일의 내용에 관계없는 부가 설명적인 내용을 파일에 기술하게 하는 것을 가능하게 해준다 VRML 파일에서 주석은 샵(#)기호로 시작하며 그 줄의 마지막까지를 주석으로 처리하게 된

다. 앞의 VRML 헤더도 VRML 주석의 일부로 처리된 것을 볼 수 있다.

### 3.2.3 노드(Node)

VRML 파일은 각종 3차원 가상환경의 형태(shape)들과 그들의 특성(property)들을 나타내기 위해 노드를 가지게 된다. 각각의 VRML 노드들은 형태(shape), 색상(color), 빛(light), 시점(viewpoint), 위치(position), 방향(orientation), 애니메이션 타이머(animation timer), 센서(sensor), 인터polator(interpolator) 등을 나타내게 된다. VRML 노드의 타입은 다음과 같다.

| Type                          | Nodes  |
|-------------------------------|--|
| 형태, 기하정보<br>(Shape, geometry) | Box, Cone, Coordinate, Cylinder, ElevationGrid, Extrusion, IndexedFaceSet, IndexedLineSet, Normal, PointSet, Shape, Sphere, Text             |
| 외양<br>(Appearance)            | Appearance, Color, FontStyle, ImageTexture, Material, MovieTexture, PixelTexture, TextureCoordinate, TextureTransform                        |
| 그룹화<br>(Grouping)             | Anchor, Billboard, Collision, Group, Inline, LOD, Switch, Transform  |
| 환경설정<br>(Environment)         | AudioClip, Background, DirectionalLight, Fog, PointLight, Sound, SpotLight   |
| 시점관련<br>(Viewing)             | NavigationInfo, Viewpoint  |
| 애니메이션<br>(Animation)          | ColorInterpolator, CoordinateInterpolator, NormalInterpolator, OrientationInterpolator, PositionInterpolator, ScalarInterpolator, TimeSensor |
| 상호작용<br>(Interaction)         | CylinderSensor, PlaneSensor, ProximitySensor, SphereSensor, TouchSensor, VisibilitySensor  |
| 기타<br>(Miscellaneous)         | Script, WorldInfo  |

각각의 VRML 노드는 다음과 같은 항목을 가진다.

- 노드 타입

- 일련의 중괄호("{ }") 집합
- 여러 개의 필드(Field)

이 중 노드 타입과 중괄호는 반드시 필요하며, 필드는 각각의 노드의 특성에 따라 존재하지 않을 수도 있으며, 하나 이상의 필드를 가질 수도 있다. 다음의 예를 보자

```
Cylinder {
    height 20
    radius 20
}
```

여기서는 "Cylinder" 타입을 가지며 하나의 중괄호 내부에 "height"와 "radius"라는 두 개의 필드를 가지는 노드를 표시하고 있음을 알 수 있다.

### 3.2.4 필드(Field)와 필드 값(Field value)

위의 Cylinder노드 예제에서 height, radius 등과 같이 어떤 node 내에서 그 노드의 속성값을 나타내는 것을 필드라 한다. 한 node 내에서 필드의 순서는 중요하지 않으며 각 node는 대부분 미리 정의된 일련의 필드를 가진다. 모든 필드는 그 이름과 하나 이상의 적합한 데이터 타입의 값을 가지게 된다. 필드 값의 데이터 타입들은 정수(integer), 부동소수값(floating-point number), 가부값(boolean), RGB색상, 2차원 및 3차원 좌표 등이 있으며 어떤 다른 node가 다른 node의 필드 값이 될 수도 있다.

### 3.2.5 노드 이름의 정의

VRML파일에서는 어떤 node에 사용자가 이름을 붙일 수 있다. 이는 DEF 키워드를 이용하며 그 사용 문법은 다음과 같다.

```
DEF node-name node-type { . }
```

예를 들어 위의 Cylinder 예제를 다음과 같이 바꿀 수 있다.

```
DEF dikoCylinder Cylinder {
    Height 2.0
    Radius 2.0
}
```

위와 같이 하면 이 높이 2, 반지름 2의 원통형 모형에 대한 노드의 인스턴스는 같은 VRML 파일 내에서 “dikoCylinder”라는 이름으로 참조가 가능하게 된다. 사용되는 이름은 ASCII 문자와 숫자의 조합으로 이루어지며, 숫자가 가장 앞에 나오지는 못한다. 이 정의된 이름은 나중에 USE 키워드를 이용해 사용할 수 있다.

### 3.3 VRML 예제

다음과 같은 VRML 파일을 예로 들자.

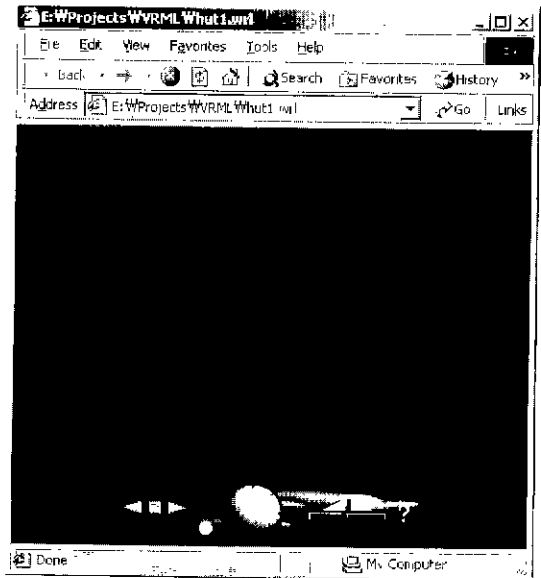
```
#VRML V2.0 utf8
#오두막집
Group {
  children [
    #오두막집의 외벽
    Shape {
      appearance DEF Red Appearance {
        material Material {
          diffuseColor 0.8 0.0 0.0
        }
      }
      geometry Cylinder {
        height 2.0
        radius 2.0
      }
    },
    #오두막집의 지붕
    Transform {
      translation 0.0 2.0 0.0
      children Shape {
        appearance USE Red
        geometry Cone {
          height 2.0
          bottomRadius 2.5
        }
      }
    }
  ]
}
```

가장 첫 줄에는 공통된 VRML 헤더가 위치함을 알 수 있다. 이 VRML파일은 하나의 오두막집 모형을 보여주는 파일로서 첫번째로 Group 노드를 볼 수 있다. Group 노드는 여러 개의 노드를 하나로 묶을 경우 사용하는 노드로 children 필드를 가진다. children 필드는 또한 여러 개의 다른 자식 노드들을 그 값으로 가지는데 각 자식 노드들은 콤마(,)로써 구분된다. 이 예제에서는 Shape와 Transform의 타입을 가지는 두 개의 노드를 자식 노드로 가지게 된다.

첫번째 Shape타입의 자식노드는 오두막집의 외벽을 나타내는 노드로 appearance와 geometry의 두 개의 필드를 가지고 있다 여기서 주목할 것은 첫번째 appearance필드에서 그 값으로 사용되는 Appearance 타입의 노드를 “Red”라는 이름의 노드로 정의하고 있는 것이다. 이 첫번째 Shape 타입 노드의 의미는 높이 2, 반지름 2의 붉은색 원통모형을 원점에 위치시키는 것이다.

두번째 Transform 타입 노드는 이후 자식 노드가 적용 받을 좌표계를 이동시키는 역할을 한다. translation 필드의 값이 “0.0 2.0 0.0”인 것은 원점의 좌표를 y축으로 2만큼 이동시키라는 것이다. 이 Transform 타입노드는 하나의 자식 노드로서 Shape 타입 노드를 가지는데 이는 오두막집의 지붕을 나타낸다.

위 VRML파일을 VRML브라우저로 보면 다음과 같은 결과를 볼 수 있다.



### 3.4 VRML 애니메이션

VRML 1.0은 웹에서의 새로운 3차원 정보 표현 방법의 표준을 제시했다는 점에서 그 의의가 깊었으나 동적 애니메이션 표현 및 상호작용(interactive) 응용을 작성하는 데는 많은 한계점이 있었다. 이에 VRML97에는 애니메이션과 상호작용을 위한 많은 부분이 추가되었다 여기서 는 예제를 통해 이들이 어떻게 작용하는지 알아

본다.

```
#VRML V2.0 utf8
#The floor
Transform(
  translation      0 -0.5 0
  children [
    Shape {
      appearance Appearance {
        material Material
          (emissiveColor 0.6 0 0)
      }
      geometry Box (size 10 0.1 0)
    }
  ]
)
#now for the cone itself
DEF sphere_tx Transform {
  translation 0 0 4
  children Transform {
    rotation 0 0 1 157
    children Shape {
      appearance Appearance {
        material Material
          (emissiveColor 0 0 0.6)
      }
      geometry Sphere {
        radius 0.5
      }
    }
  }
}
DEF pos_interp PositionInterpolator {
  key [0 0.125 0.25 0.375 0.5 0.625 0.75 0.875 1]
  keyValue [0 0 4, 2.831 0 2.831, 4 0 0, 2.831 0 -2.831, 0 0 -4,
    -2.831 0 -2.831, -4 0 0, -2.831 0 2.831, 0 0 4]
}
DEF orient_interp OrientationInterpolator {
  key [0 0.125 0.25 0.375 0.5 0.625 0.75 0.875 1]
  keyValue [0 1 0 157, 0 1 0 0.785, 0 1 0 0, 0 1 0 -0.785,
    0 1 0 -1.57, 0 1 0 -2.36, 0 1 0 -3.14, 0 1 0 -3.93,
    0 1 0 -4.71, 0 1 0 5.50, 0 1 0 6.28]
}
DEF time TimeSensor {
  loop TRUE
  startTime 1
}
ROUTE time.fraction_changed TO pos_interp.set_fraction
ROUTE pos_interp.value_changed TO sphere_tx.translation
ROUTE time.fraction_changed TO orient_interp.set_fraction
ROUTE orient_interp.value_changed TO sphere_tx.rotation
```

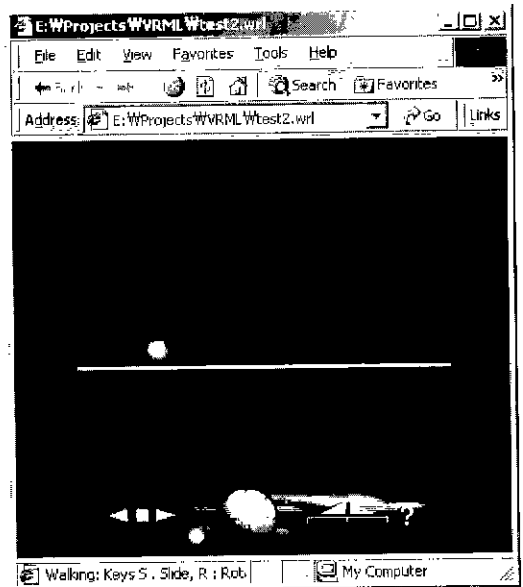
VRML97에서 가장 크게 변한 점은 이벤트와 라우팅의 추가라고 할 수 있다. 이에 따라 VRML 브라우저는 이전의 단순한 정보 표현 터미널에서 일종의 실시간 실행환경(run-time environments)을 관리하게 되었다.

VRML에서 이벤트의 전달은 "ROUTE" 키워

드를 통해 이루어지게 된다. ROUTE키워드는 어떤 한 노드에서 발생하는 이벤트를 다른 노드에 특정 필드에 전달하게 된다 위의 예에서 "ROUTE time.fraction\_changed TO pos\_interp.set\_fraction" 이라는 부분은 "time"이라는 이름의 노드에서 발생하는 "fraction\_changed" 이벤트를 "pos\_interp" 노드의 set\_fraction 필드로 전달하라는 의미이다. VRML 브라우저는 실행중에 이를 계속 전달할 수 있는 일종의 실시간 실행환경을 제공해야 한다.

애니메이션을 위해서는 타이머의 제공이 필수적이다. TimeSensor 타입의 노드는 이를 제공한다. PositionInterpolator와 OrientationInterpolator 타입의 노드는 일종의 키프레임방식 애니메이션을 위한 각 키값에 대한 중간 값을 기술하기 위한 노드들이다.

위의 예에서는 먼저 첫번째 ROUTE문에 의해 타이머에서 발생한 시간의 변화가 PositionInterpolator 타입 노드인 "pos\_interp"에 전달되게 된다. pos\_interp는 시간의 변화를 키값으로 인식하고 이에 해당하는 값을 정의된 중간 값들을 보간(interpolation)하여 결정한다. 다시 두번째 ROUTE문에 의해 이 값은 Sphere타입의 Shape 노드에게 전달되어 그 노드의 위치를 보간된 값으로 변경되게 한다. VRML 브라우저는 이를 위해 지속적으로 이벤트를 발생시키고, 여기에서





발생되는 이벤트들을 각 노드에 전달해야 한다. 위의 VRML 파일을 실행시키면 다음과 같이 중심을 계속 선회하는 구 모양의 3차원 모형의 애니메이션을 볼 수 있다.

#### 4. 결 론

본 논문에서는 웹을 기반으로 멀티미디어 서비스를 제공하기 위한 대표적인 언어인 SMIL과 VRML을 살펴보았다. 이미 이 두 언어의 경우 기존의 노력들과는 달리 제품화도 많이 되어 있고, 이미 상용화 단계라고 말할 수 있다.

SMIL은 간단한 구조를 가지고 멀티미디어 프리젠테이션의 동기화 등을 표현하는데 우수성을 입증하고 있다. VRML은 간단한 스크립트 언어의 범위를 넘어서 프로그래밍적인 개념을 도입하여 복잡한 3차원 멀티미디어 객체를 정의하는데 적합하다. 이 두 언어는 표준이 이미 제정되었지만 아직도 많은 보강과 연구가 필요하다. 특히 SMIL은 QoS를 반영하는 문제와 SMIL 문서를 WYSIWYG 방식으로 편집할 수 있도록 하기 위하여 멀티미디어 데이터베이스나 멀티미디어 처리 기술과 연계된 많은 연구가 필요하다. 또한, 기존의 다른 XML기반의 언어들과 상호 결합하여 사용하는 데에 더 많은 연구가 현재 진행 중이다. 이는 VRML도 마찬가지다.

앞으로 기반 네트워크 환경이 구축되면 초고속 인터넷 서비스가 일반화되고 멀티미디어 서비스들 역시 점점더 중요해 질 것이다, 그를 위해 간단한 구조를 가지고 있으면서도 복잡한 멀티미디어 또는 3차원의 표현을 나타낼 수 있는 언어가 더욱 절실하다. 그 노력에는 스크립트 언어만으로 되는 것이 있고, 복잡한 프로그래밍 기법을 필요로 하는 것도 있어서 이 두 가지 노력은 상호 보완적으로 발전되어야 할 것이다.

#### 참고문헌

[1] W3C, "Synchronized Multimedia Integration Language(SMIL) 1.0 Specification," W3C, 1998.

[2] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, RTP:A transport protocol for real-time applications., RFC-1889, Feb.,1996.

[3] H. Schulzrinne, RTP Profile for Audio and Video Conference with Minimal Control, RFC-1890. May, 1996.

[4] H. Schulzrinne, A Rao, R. Lanphier, Real Time Streaming Protocol(RTSP) , RFC-RFC2326, 1998.

[5] W3C, Synchronized Multimedia Integration Language (SMIL) Boston Specification. W3C, Feb., 2000 <http://www.w3.org/TR/smil-boston/>.

[6] R.Carey, "The Virtual Reality Modeling Language Explained," IEEE Multimedia, Vol. 5. No. 3. pp.84-93, 1998.

[7] D.R.Nadeau "Building Virtual Worlds with VRML," IEEE Computer Graphics and Applications, Vol. 19, No. 2, pp.18-29, 1999.

[8] Bernie Roehl et al. "Late Night VRML 2.0 with Java," Ziff-Davis Press, ISBN 1-56276-504-3, 1997.

[9] Andrea L. Ames et al. "VRML 2.0 Sourcebook," John Wiley & Sons, Inc. ISBN 0-471-16507-7. 1997.

---

#### 김 지 용



1990~1995 서울대학교 컴퓨터공학  
과(공학사)  
1995~1997 서울대학교 컴퓨터공학  
과(공학석사)  
1998~1999 (주)신 테크  
1999~현재 한국전자통신연구원 멀  
티미디어그룹웨어팀 연구원  
관심분야:인터넷 실시간 멀티미디어  
서비스, 전자상거래, 인텔리전  
트 에이전트  
E-mail: kjy@etri.re.kr

---

고 동 일



1997 서울대학교 컴퓨터공학과(학사)  
1999 서울대학교 컴퓨터공학과(석사)  
1999~현재 한국전자통신연구원 연구원  
관심분야 분산가상현실, 컴퓨터네트워크  
E-mail diko@etri.re.kr

김 두 현



1981~1985 서울대학교 컴퓨터공학과(공학사)  
1985~1987 한국과학기술원 전산학과(이학석사)  
1987~현재 한국전자통신연구원 책임연구원 멀티미디어그룹웨어팀 팀장  
관심분야 인터넷 실시간 멀티미디어 서비스, 분산 멀티미디어 시스템, 멀티미디어 그룹웨어  
E-mail doohyun@etri.re.kr

• JCCI 2000 •

- 일 자 : 2000년 5월 25 ~ 27일
- 장 소 : 경주
- 주 최 : 정보통신연구회
- 심사결과통보 : 2000년 4월 15일
- 최종본제출마감 : 2000년 4월 29일
- 제 출 처
  - 1) Hard Copy : 121-742 서울특별시 마포구 신수동 1번지  
서강대학교 컴퓨터학과 최명환 교수  
Tel. 02-705-8495 Fax. 02-704-8273  
Email:mchoi@ccs.sogang.ac.kr
  - 2) Electronic Copy : 한양대학교 전자전기공학부 정재일 교수  
Tel. 02-2290-0352  
Email:jijung@sophia.hanyang.ac.kr
- 논문제출 양식 및 설문지 양식 : 홈페이지 참조  
홈페이지: <http://ccl.cnu.ac.kr/jcci/2000>