

SMIL 스크립트 생성을 위한 시각적 편집도구

김영환*, 임명수*, 강석찬*, 이남옥**, 임영환*

*송실대학교 대학원 컴퓨터학과

** (주) 이지미디어

e-mail : yhdosa@archi.soongsil.ac.kr

Implementation of Visual Editing Tool for SMIL

Younghwan Kim*, Myungsu Lim*, Suckchan Kang*,

Namock Lee**, Younghwan Lim*

*Dept. of Computing Soongsil Univ.

**EasyMedia co.,ltd

Abstract

본 논문은 SMIL의 작성 시에 편리함을 제공하기 위한 시각적 편집 도구를 제안한다. SMIL은 서로 다른 미디어간의 동기화를 가능하게 하는 스크립트 언어이다. SMIL은 텍스트 편집기를 통하여 작성되는데 이는 작성에 있어 직관적인 면이 없어 많은 불편함을 갖는다. 따라서 시각적 편집 도구가 필요하며 이를 구현하기 위해 본 논문에서는 SMIL의 작성을 프리젠테이션과 하이퍼 프리젠테이션의 두 측면에서 설계하였으며 시각적 편집 도구 구현에는 멀티미디어 아이콘 프로그램 도구인 VIP를 확장하여 완성하는 방법을 택하였다. VIP를 확장함으로써 SMIL 번역이라는 기본 목적 외에 부가적으로 MuX의 공간적 동기화에 관련된 기능의 한계를 극복할 수 있게 된다.

1. 서론

멀티미디어에 있어서 현재의 인터넷에서는 동기화된 서비스를 제공하는데 여러 제약이 있다. 문제가 되는 것은 멀티미디어 자체가 여러 미디어로 구성되기 때문에 서로 다른 미디어 사이에 동기화가 필요하다는 점이다. 이러한 문제점을 해결하기 위해 W3C(World Wide Web Consortium)에 의해 SMIL이란 기술 규격이 나오게 되었다[6].

SMIL을 사용하면 멀티미디어 요소들(오디오, 비디오, 그래픽, 텍스트 등)을 동기화 하여 인터넷상에서 쉽게 표현하거나, 공유할 수 있다. 그러나 스크립트 언어로 멀티미디어를 동기화 시키는 것은 많은 어려움이 있다. 본 논문에서는 이런 문제를 해결하기 위한 방법으로 SMIL을 작성하기 위한 시각적 편집

도구를 작성하는 것을 목적으로 다음의 사항을 연구하였다.

첫째, 여러 미디어간의 시간 관계를 정의하여 구성하는 프리젠테이션을 시각적으로 표현하여 SMIL을 생성하는 방법에 대한 연구

둘째, 프리젠테이션간의 하이퍼링크로 관계를 정의하여 구성하는 하이퍼 프리젠테이션을 시각적으로 표현하여 SMIL을 생성하는 방법에 대한 연구.

2. Synchronized Multimedia integration Language를 이용한 동기화 편집의 문제점

2.1 SMIL의 개요

SMIL의 기본 아이디어는 텍스트, 그래픽, 오디오, 비디오와 같은 미디어 요소를 URL과 함께 명명해

주는 것이고, 그들의 표현을 병렬적으로 또는 순차적으로 스케줄링하는 것이다. 전형적인 SMIL 프리젠테이션은 다음과 같은 특징이 있다. 프리젠테이션은 URL을 통해서 접근 가능한 몇몇 요소로 구성되어 있다. 구성 요소들은 텍스트, 그래픽, 오디오, 비디오와 같은 다른 미디어 형태를 가진다. 다른 구성 요소들의 시작과 종료의 시간은 그것과 관련된 다른 미디어 구성요소로 인한 이벤트에 의해 구체화된다. 제어 버튼을 사용하여 프리젠테이션의 인터럽트나, 프리젠테이션 상의 앞쪽이나 뒤쪽의 다른 지점으로의 빠른 이동이 가능하다. 사용자는 프리젠테이션에 내재된 하이퍼링크를 따라 갈 수 있다.

2.2 SMIL의 응용 및 예제

SMIL은 웹 사이트로부터 다중 영상, 이미지, 사운드를 개별적으로 전송하게 하지만, 그들의 시간을 조정함으로써 마치 하나의 프리젠테이션처럼 보여지게 한다. SMIL은 또한 사용자로 하여금 미디어들을 많은 다른 버전으로 저장하는 것을 허용함으로써 원하는 사용자에게 웹 페이지 전송이 가능하다.

다음은 하나의 비디오와 하나의 오디오로서 SMIL 프리젠테이션을 구성한 예이다. 비디오는 전체 320*320 크기의 바탕 화면에 40, 40(x, y좌표) 위치에 240*240의 크기로 재생되며 오디오는 0초에서 시작하여 20초에서 끝난다.

```
<smil> <head> <layout>
  <root-layout height="320" width="320"
    background-color="black"/>
  <region id="images" left="40" top="40"
    height="240" width="240"/>
</layout> </head> <body>
<par>
  <video src="house.rm" region="images"/>
  <audio src="song.ra" begin="0s" end="20s"/>
</par> </body> </smil>
```

2.3 문제점

SMIL은 HTML과 유사한 형태의 스크립트 언어로 되어 있어 비교적 작성하기도 쉽다. 그러나 미디어의 수가 많고 미디어간의 관계가 복잡한 SMIL을 텍스트 편집기를 사용하여 작성하는 경우에는 직관성이 떨어져 시간 설정 및 영역 설정에 매우 어려움을 초래하게 된다. 이를 해결하기 위해서는 SMIL을 작성하는 시각적 편집 도구가 필요하다.

본 논문의 목적은 SMIL의 시각적 편집 도구의 구현이며, 그 대상으로 VIP(Visual Interface Player)를 확장하는 방식을 택하였다. VIP는 멀티미디어 아이콘 프로그래밍 도구로서 멀티미디어 작성에 있어 시각적 편집기능을 제공하며 미디어간의 시간적 동기화의 기능을 제공한다. VIP의 기본 기능에 SMIL을 위한 확장 기능을 추가하여 시각적 편집 도구를 구현한다.

3 SMIL을 지원하는 VIP

3.1 VIP

사용자는 자신이 생각하는 서비스를 VIP가 제공하는 아이콘과 기능을 이용하여 MuX가 제공하는 모델의 각 메커니즘으로 도식화한 다음 수행하면 인터프리터가 그 명시 프로그램을 해석하여 서버가 제공하는 API를 호출하게 된다.

3.2 MuX(Multimedia I/O Server)

MuX 모델은 멀티미디어 관련 동기화나 데이터 처리 등 필요한 기능을 정의해 줄 수 있는 프레임워크를 제공함과 동시에 응용 서비스를 구성하는 매커니즘을 제공함으로써 객체지향 개념을 이용한 응용 서비스를 개발할 수 있도록 하고 있다[2][4][5]. 따라서 이 개념을 객체 지향 기법으로 구현된 멀티미디어 입출력 서버(MuX)의 API(Application Programming Interface)를 이용하면 응용 서비스 개발자는 내부의 상세하고 복잡한 멀티미디어 처리 관련 사항을 몰라도 쉽고 빠르게 초고속정보통신 서비스를 개발할 수 있게 된다.

응용 서비스에 따른 데이터의 흐름을 살펴보면 크게 3가지로 구분된다.

- 입력된 데이터의 내용을 저장하거나 가공하지 않고 경로만을 출력될 곳까지 보내주는 기능.
- 저장된 데이터를 통합하거나 동기화를 맞추어 데이터의 흐름을 재구성하는 기능.
- 사용자가 조건을 설정하여 그 조건에 만족하는 경우에 알맞은 처리를 할 수 있도록 사용자의 요구에 따른 데이터 흐름을 제어하는 기능.

이 기능들을 각각 순서대로 스트림 계층, 프리젠테이션 계층, 하이퍼 프리젠테이션 계층으로 하여 MuX 모델을 구성한다[1].

3.3 분산 멀티미디어 컴퓨팅 모델 계층

3.3.1 스트림 계층

멀티미디어 정보처리 모델의 최하위 계층으로 특정 미디어와 관련된 데이터의 흐름을 정의하고 데이터의 근원지에서 목적하는 곳까지 전송하는 방법을 제공한다. 이 계층에서 다루는 미디어의 성격은 다음과 같이 구분된다.

- 디지털로 표본된 연속 미디어 스트림 - 일정한 표본율과 패턴으로 표본된 데이터의 집합
- 합성된 연속 미디어 스트림 - 입력 장치에서 표본된 것이 아니라 여러 개의 연속미디어 스트림이 합성된 스트림을 의미한다.
- 사건 발생에 의한 스트림 - 인터럽트나 어떠한 사건 발생들의 흐름이므로 미리 결정된 것도 없고 연속성도 없다. 주로 사용자와 대화 과정에서 발생하는 키보드나 마우스의 동작에 의해 발생하는 스트림 등이다.

이러한 스트림들은 실시간인가 아닌가, 또는 재생용인가 아닌가 등에 따라 구분되어 다루어진다.

스트림 계층에서 제공되는 기능은 다음과 같다.

- 파일이나 입력장치, 통신망, 또는 상위계층에서의 데이터 입력, 수집기능
- 멀티미디어 파일에 저장하거나 통신망 또는 상위계층에 전송하는 기능
- 데이터 압축들과 같이 개별 스트림을 처리하는 기능
- 여러 개의 스트림 중에서 목적지로 보낼 한 개의 스트림을 선택하는 기능과 그 스트림을 목적하는 곳으로 분배하는 기능
- 동기화를 위한 시간 삽입 기능

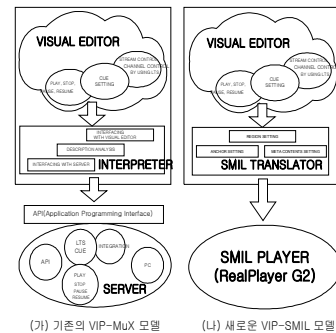
3.3.2 프리젠테이션 계층

프리젠테이션 (혹은 멀티미디어 프리젠테이션) 계층은 스트림 계층 위에 구성된다. 멀티미디어 표현은 시간과 공간 조건에 의해 조정된 스트림의 모음이라고 볼 수 있다. 따라서 이 계층 내의 스트림은 동기화 되어 있다. 또한 음성이나 비디오와 같은 서로 다른 타입을 가지는 스트림들은 공통으로 사용하는 표현제어 방식을 동기화와 별도로 표현한다. 같은 타입의 스트림은 편집, 합성, 통합 등을 통해 새로운 스트림으로 구성된다. 이 계층에서 제공하는 기본 기능은 다음과 같다.

- 스트림간 또는 스트림 내 동기화 처리기능
- 동기화된 스트림의 통합기능
- 프리젠테이션에 관련된 스트림의 처리기능

3.3 SMIL을 지원하는 새로운 VIP 시스템 구조

기존의 MuX의 프리젠테이션 모델에서는 [그림 3-1]의 (가)와 같은 클라이언트-서버 개념으로 구현된 서버의 API를 시각 편집기인 VIP로 제작된 제어 프로그램을 해석하여 직접 MuX의 API를 호출하는 인터프리터 형태를 제공해주고 있다[3]. 그러나 본 논문에서 제시하는 SMIL의 시각적 편집 도구는 [그림 3-1]의 (나)와 같은 VIP-SMIL 모델로서 VIP가 시간에 관계된 프리젠테이션을 구성하면 SMIL을 위한 VIP의 확장된 기능을 통해 영역 설정 및 앵커 설정을 하여 SMIL 파일로 만들어 낸다. 만들어진 SMIL은 SMIL Player를 통해 실행된다.

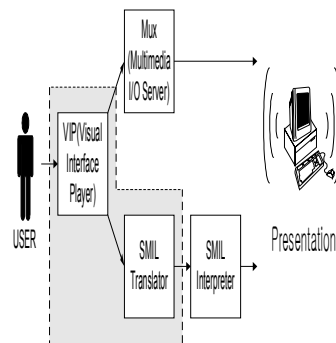


[그림 3-1] 시스템 구조도

4 SMIL Translator의 개발

4.1 SMIL Translator의 구조

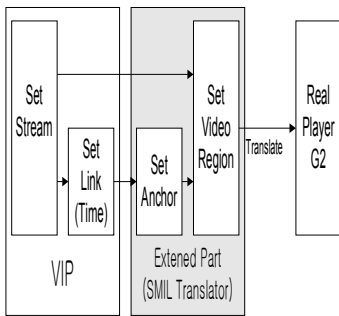
VIP를 사용하여 만들어진 프리젠테이션 또는 하이퍼 프리젠테이션은 SMIL Translator를 통해 SMIL로 만들어진다. 출력된 SMIL은 SMIL 인터프리터를 통해 실행된다.([그림 4-1] 참조) SMIL은 표준 기술 규격이기 때문에 이를 지원하는 어떤 인터프리터로도 재생이 가능하게 된다. 본 연구에서는 현재 일반적으로 가장 많이 쓰이는 Real Network사의 Real Player G2를 사용하였다[10]. 또한 VIP로 작성된 프리젠테이션은 MuX의 API를 사용할 수 있으므로 MuX 서버가 있다면 바로 재생할 수도 있다.



[그림 4-1] SMIL Translator의 전체 구조도

4.2 확장된 VIP의 세부 구조

사용자는 VIP의 본래의 기능을 이용하여 프리젠테이션 또는 하이퍼 프리젠테이션을 제작한다. VIP의 본 기능만을 이용하면 시간에 관계된 설정만을 할 수 있게 된다 즉, 미디어들 간에 시간적인 동기화가 설정되는 것이다. 다음 단계로 SMIL을 제작하기 위해서는 공간적 동기화가 필요하다. 즉, 비디오가 출력되는 영역(Region), 앵커가 지정되는 영역을 VIP의 확장된 기능을 통해 설정한다.([그림 4-2] 참조) 프리젠테이션을 만드는 경우에는 링크를 설정하지 않으므로 미디어간의 시간 설정 후 VIP의 확장된 기능을 이용하여 영역 및 부가 정보를 설정한다. 모든 설정이 완료되면 SMIL Translator를 통해 SMIL을 만든다.



[그림 4-2] SMIL Translator의 세부 구조

4.3 확장된 VIP의 화면 설계

SMIL을 위한 확장된 곳은 다음의 3부분이다.

- 프리젠테이션의 영역 및 부가정보 설정
- 하이퍼 프리젠테이션의 앵커 설정
- 하이퍼 프리젠테이션의 영역 및 부가 정보 설정

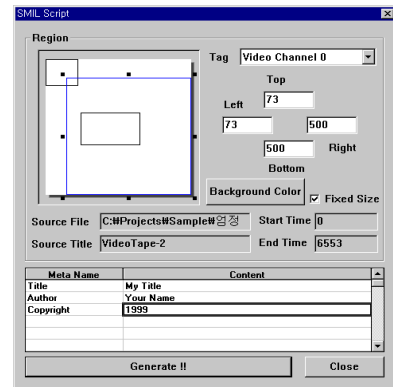
4.3.1 프리젠테이션의 영역 및 부가정보 설정

기본 설정외에 SMIL로 만들어지기 위해 필요한 사항은 다음과 같다.

- SMIL이 재생 되어지는 전체 화면의 크기
- 프리젠테이션을 구성하는 비디오, 이미지, 텍스트의 크기와 위치
- 바탕화면의 색상
- 프리젠테이션 자체의 부가적인 정보

위의 사항을 지정하기 위해 다음의 [그림 4-3]의 화면을 설계하였다. 상단의 Region 은 재생되는 화면의 전체 크기 및 비디오 채널의 크기 및 위치를 설정한다. 상단 좌측의 콤보 박스로 구성된 Tag는

설정할 현재 프리젠테이션 내의 전체 크기 (Root-Layout) 혹은 비디오 채널을 선택할 수 있는 항목이다. Tag 콤보 박스의 바로 아래는 선택된 영역의 크기를 left, top, right, bottom의 값으로 설정하는 항목이다. Background Color는 SMIL이 실행될 때의 배경색을 설정한다. Fixed Size는 비디오 채널이 실행될 때 현재 설정된 영역의 크기에 고정시킬 것인지 원래의 크기를 유지할 것인지를 설정한다. 하단의 정보는 SMIL 프리젠테이션 자체의 정보를 지정한다. 실행에는 영향을 미치지 않으며 단순히 작성자, 날짜, 프리젠테이션의 이름 등을 설정한다.



[그림 4-3] 프리젠테이션의 영역 및 부가 정보 설정 화면

4.4 번역 알고리즘

4.4.1 프리젠테이션의 SMIL로 번역

프리젠테이션을 SMIL로 번역하는 알고리즘은 다음과 같은 과정을 갖는다.

- (1) Head 부분을 생성한다.
`<smil> <head>`를 출력한다.
- (2) Meta Contents를 출력한다.
- (3) Root-Layout과 Region의 ID및 크기정보 출력
- (4) Head를 끝내고 Body를 시작한다.
`</head><body><par>`를 출력한다.
 모든 트랙들은 병렬 구성하여 전체 시간에 맞춘다. 따라서 `<par></par>`내에 모든 트랙을 출력한다.
- (5) Track을 읽어들인 후 오디오, 비디오, 이미지, 텍스트인지 구별하여 출력한다. 이때 비디오, 이미지, 텍스트의 화면에 관련된 트랙은 (3)의 과정에서 생성된 Region의 ID를 부여하여 출력한다.
- (6) 모든 Track을 출력할 때까지 (5)의 과정을 반복
- (7) Body , SMIL 끝 부분을 출력한다.
`</par></body></smil>`를 출력한다.

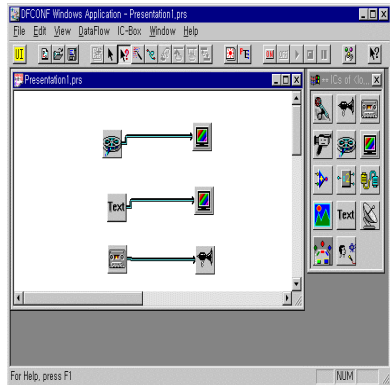
5. 실험 결과

5.1 프리젠테이션의 SMIL 생성

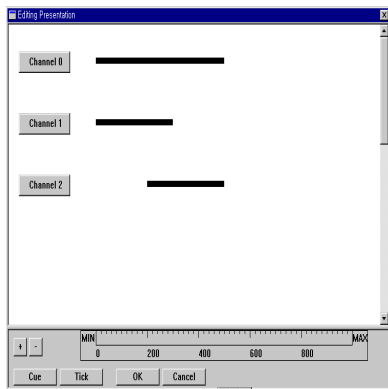
5.1.1 스트림 생성 및 시간 설정

VIP를 이용하여 다음의 [그림 5-1]과 같은 프리젠테이션을 구성한다. 프리젠테이션의 내용은 전체 세 개의 채널로 구성되며 각각 하나의 비디오, 오디오, 텍스트로 구성한다.

구성된 프리젠테이션의 시간 설정을 다음의 [그림 5-2]와 같이 설정한다. 비디오는 0초에서 시작하여 50초까지 재생되도록 설정한다. 텍스트는 0초에서 30초까지, 오디오는 20초에서 50까지 각각 설정한다.



[그림 5-1] VIP를 이용하여 멀티미디어 스트림 작성



[그림 5-2] 각 미디어의 시간 설정

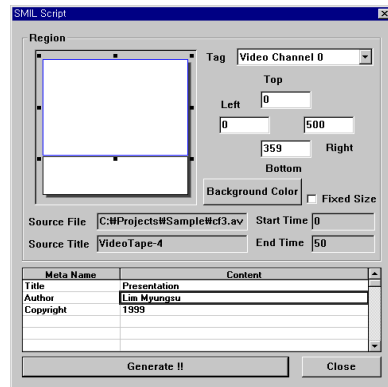
5.1.2 SMIL을 위한 영역 및 부가 정보 설정

프리젠테이션 내의 미디어 중 화면에 표시되는 미디어(동영상, 이미지, 텍스트 등)는 표시될 영역 설정(공간적 동기화)을 해야 한다. 앞의 5.1.1과 5.1.2에서 작성한 프리젠테이션에 [그림 5-3]과 같이 영역을 설정한다. 즉, 전체 프리젠테이션의 크기는 500*500이며, 그 내부에 비디오 화면은 시작점 (0,0)에서 끝점 (359,500)의 크기로 설정하고 텍스트는 비디오 화면 바로 아래의 빈 영역에 나오도록 설정한

다.

5.1.3 SMIL 출력 결과 및 재생

앞의 5.1.1, 5.1.2, 5.1.3의 과정을 거쳐 만들어진 프리젠테이션을 [그림 5-3]의 하단에 있는 “Generate” 버튼을 눌러 SMIL로 출력한다. 출력된 SMIL은 다음과 같다.



[그림 5-3] 프리젠테이션의 영역 설정과 Meta Contents 설정

```
<smil> <head>
<!-- Presentation attributes. -->
<meta name = "Title" content = "Presentation"/>
<meta name = "Author" content = "Kim Younghwan"/>
<meta name = "Copyright" content = "1999"/>
<layout>
<!-- Width, height, and background color of
entire presentation. -->
<root-layout height="500" width="500"
background-color="#ffffff"/>
<!-- region. -->
<region id="region0" left="0" top="0" height="356"
width="500" fit="fill"/>
<region id="region1" left="0" top="359"
height="141" width="500" fit="fill"/>
</layout> </head> <body> <par>
<!--Play these streams concurrently in
parallel.-->
<video src="file:///C:\WProjects\WSample\Wcf3.avi"
region = "region0" title = "VideoTape-4"
alt = "VideoTape Track" begin = "0s"
end = "50s"/>
<text src="file:///C:\WProjects\WSample\WReadme.txt"
```

```

region = "region1" title = "Text"
alt = "VideoTape Track" begin = "0s"
end = "30s"/>
<audio src="file:///C:/WProjects/WSample/Wsong.ra"
title = "AudioTape-10" alt="AudioTape Track"
begin = "20s" end = "50s"/>
</par> </body> </smil>
    
```

실제로 수행한 결과는 다음의 [그림 5-4]와 같다. 프리젠테이션이 시작되면 비디오와 텍스트가 나온다. 프리젠테이션 시작 후 20초 후에는 오디오가 시작된다. 오디오 시작 후 10초 뒤 화면 하단의 텍스트가 없어지며 50초가 되면 프리젠테이션이 끝난다.



[그림 5-4] 실행 화면

6. 결론

시각적 편집 도구는 프리젠테이션, 하이퍼 프리젠테이션의 작성 시 직관적인 모습을 보여주므로 텍스트를 사용한 직접적인 작성보다 훨씬 편리함을 제공한다. 또한 VIP를 확장하여 SMIL 번역기를 만들었으므로 기존의 VIP-MuX 입장에서 보면 MuX가 갖는 기능적 한계(공간적 동기화 기능의 부재, VIP 프로그램을 재생하기 위해 항상 MuX가 존재해야 하는 점)를 넘을 수 있다.

참고문헌

- [1] 김두현, 임영환, "분산멀티미디어 시스템을 위한 범용 멀티미디어 처리 모델의 객체지향, 클라이언트 서버 구조," 한국 정보 처리학회 정보처리 논문지 제 3권 제 1호, 1996년 1월, pp.9 - 32
- [2] 임영환, "ComBiStation : 분산 멀티미디어 컴퓨팅 환경을 위한 컴퓨터 플랫폼," 정보과학회 논문지 제 2권 제 1호, pp. 6 - 8
- [3] 임영환, 이선혜, 임명수, "인터넷상의 동영상 메일을 재생하기 위한 실시간 연출 기법 연구," 한

국 정보 처리학회 정보처리 논문지 제6권 제4호, 1999년 4월, pp.877 - 889

- [4] B. Ozden and R. S. Yu, "Consumption-Based Buffer Management for Maximizing System Throughput of a Continuous Media Data," in Proc. of IEEE International Conference on Multimedia Computing and Systems, Jun 1996.
- [5] Rennison. E, R.Bker, D.H.Kim, and Y.H.Lim, "MuX : An X Co-Existant Time-Based Multimedia I/O Server," The X Resource, Issue 1, 1992, pp.213-233
- [6] W3C, *Synchronized Multimedia Integration Language (SMIL) 1.0 Specification*, <http://www.w3.org/TR/1998/REC-smil-19980615>.
- [7] RealNetworks, *RealPlayerG2*, <http://www.real.com>.