

# 검색 기능을 지원하는 SMIL 플레이어의 개발

이훈범<sup>0</sup> 백인구 한기준

건국대학교 컴퓨터공학과  
{hblee, igbaek, kjhan}@db.konkuk.ac.kr

## Development of a SMIL Player Supporting the Retrieval Facility

Hoon-Bum Lee, In-Gu Baek, Ki-Joon Han  
Department of Computer Science & Engineering, Kon-Kuk University

### 요 약

현대 사회는 웹이라는 것을 통하여 수많은 정보들을 얻을 수 있으며 시간과 공간을 초월하고 있다. 이러한 정보들 가운데 이미지, 동영상, 오디오, 텍스트 등을 실시간으로 동시에 제공받는 것은 아직도 웹 상에서는 한계가 있다. 이러한 한계를 극복하기 위해 W3C(World Wide Web Consortium)에서 XML 어플리케이션의 하나로써 SMIL(Synchronized Multimedia Integration Language)이란 표준화된 방안과 기술을 개발하였고, 이러한 표준화된 SMIL을 바탕으로 기존의 디지털 T.V, VOD, 웹 상에서의 멀티미디어 프리젠테이션 등과 같은 분야와 연계하고자 하는 연구가 활발히 진행되고 있다. 그러나, 리얼시스템사의 G2나 애플사의 QuickTime 과 같은 SMIL을 지원하는 상용 플레이어들을 비롯한 기존의 다양한 SMIL 플레이어에는 SMIL 문서를 검색할 수 있는 해결책을 제시하지 못하고 있다. 현재와 같은 기하급수적인 웹의 발전은 기본적인 웹의 기능 외에 웹에 대한 다양한 검색 기능이 제공되면서부터 이더 웹의 이러한 발전 과정을 미뤄 볼 때 SMIL 또한 이러한 검색 기능의 제공은 필수적이라 할 수 있다. 그러므로, 본 논문에서는 기존의 SMIL 플레이어들이 지원하지 못했던 검색 기능을 지원하기 위해 SMIL에서 의미를 가지는 엘리먼트와 콘텐츠를 가지고 인덱스를 구축하고, 또한 이를 이용하여 산재해 있는 SMIL 문서와 문서내의 미디어 객체 등에 대한 검색을 수행한 후 선별적으로 디스플레이할 수 있게 하는 검색 기능을 갖는 SMIL 플레이어를 설계 및 구현한다.

### 1. 서론

인터넷과 웹 상에서 가장 빠르게 성장하고 있으며 가장 수익성이 높은 분야는 바로 멀티미디어 산업이다. 오디오 화질과 비디오 화질을 웹 페이지에 삽입하는 방법은 상당히 오래 전부터 알려져 왔으며, 개발자들은 이제 인터넷을 통해 다양한 매체 내에 포함된 다양한 정보를 쉽게 활용하기 위해 웹을 실시간 오디오 및 비디오 브로드캐스팅을 하기 위한 수단으로 사용하고 있다[11]. SMIL(Synchronized Multimedia Integration Language)은 개발자들의 이러한 요구사항을 해결하기 위한 XML 어플리케이션으로 웹을 통한 멀티미디어 배포라는 역할을 하기 위해 W3C 표준으로서 개발되었다[8,9].

W3C의 SMIL은 일련의 개별적 멀티미디어 객체들을 동기화된 멀티미디어 표현(presentation)으로 통합할 수 있다. 즉, SMIL을 사용하여 표현의 시간적 행동을 기술하고, 화면상에서 표현의 배치를 기술하고, 또한 하이퍼링크를 미디어 객체와 결합할 수 있다. 다시 말하자면 SMIL이 멀티미디어 콘텐츠를 표현하기 위해서 사용되는 XML 어플리케이션이라는 것이다. SMIL 문서가 가지고 있는 멀티미디어 콘텐츠의 특성상 SMIL 문서를 사용자에게 표현하기 위해서는 현재의 웹브라우저나 일반적인 XML 지원 브라우저로는 불가능하다[5]. 그러므로, SMIL 문서를 위한 SMIL 전용의 브라우저가 있어야 한다.

본 논문에서는 SMIL 문서에 삽입된 자원에 대한 정보를 제공하는 강력한 메타모델인 RDF(Resource Description Framework)[6,7] 및 SMIL 문서의 여러 구성요소를 사용하여 구축된 인덱스를 사용함으로써 질적으로 또는 양적으로 방대한 SMIL 문서들에 대해 효율적인 검색 기능을 갖는 SMIL 전용의 브라우저인 SMIL 플레이어를 설계 및 구현한다.

본 논문의 구성은 다음과 같다. 제1 장 서론에 이어 제2 장에서는 관련 연구로서 W3C SMIL의 주요 구조와 RDF에 대하여

분석한다. 제3 장에서는 검색 기능을 갖는 SMIL 플레이어의 설계 및 구현에 대하여 설명한다. 마지막으로 제4 장에서는 결론과 앞으로의 연구방향에 대해 언급한다.

### 2. 관련 연구

본 장에서는 관련 연구로서 W3C에서 제안한 SMIL의 주요 구조와 RDF의 목적, 개요, 응용분야, 구성에 대하여 설명한다.

#### 2.1 SMIL의 주요 구조

W3C는 웹을 통한 멀티미디어 배포를 위해 SMIL 표준안을 제안하였다[8,9]. 다른 XML 어플리케이션의 엘리먼트와 마찬가지로 모든 SMIL 엘리먼트와 그 콘텐츠는 <SMIL>...</SMIL> 엘리먼트 안에 들어있다. 다음은 본 논문의 질의검색 모듈에서 INDEX DICTIONARY의 구성을 위한 SMIL 문서의 주요한 부분과 더불어 그 안에서 찾게 될 엘리먼트를 기술하고, 또한 유효한 SMIL 문서를 구축하는데 이 엘리먼트를 어떻게 사용할 것인지를 설명한다. SMIL DTD는 본 논문에 포함시키기는 너무 길지만, W3C 웹사이트인 <http://www.w3.org/TR/REC-smil/>에서 그 복사본을 구할 수 있다.

문서 헤드 부분에 사용되는 엘리먼트로는 <HEAD>, <LAYOUT>, <REGION>이 있으며 <HEAD> 엘리먼트는 헤드 정보를 포함하고 있는 SMIL 문서의 부분을 정의한다. 이 엘리먼트의 콘텐츠는 LAYOUT, META(RDF), SWITCH 등이 있으며 속성은 ID가 있다. <LAYOUT> 엘리먼트는 프리젠테이션이 어떤 레이아웃 영역을 사용할 것인지를 지정한다. 이 콘텐츠는 REGION, ROOT-LAYOUT이 있으며 속성은 ID와 TYPE이 있다. EMPTY-REGION 엘리먼트는 레이아웃과 프리젠테이션의 특정한 영역을 지정하는데, 포함하는 콘텐츠로는 SKIP-CONTENT가 있으며, 속성은 BACKGROUND-COLOR, FIT=(FILL / HIDDEN / MEET / SCROLL / SLICE), HEIGHT, ID, LEFT, TITLE, TOP, WIDTH, ZINDEX 등이 있다.

여러 개의 <LAYOUT>...</LAYOUT>과 <REGION> 엘리먼트를 사용하여 여러 가지 종류의 정보를 포함하고 있는 프리젠테이션에 여러 개의 영역에 대한 여러 가지 레이아웃을 정의할 수 있다[5]. <REGION> 엘리먼트에 대한 대다수의 속성은 영역의 물리적인 프로퍼티에 대한 톨을 제공한다. BACKGROUND-COLOR 속성을 사용하여 영역에 대한 배경 색상을 지정하고, HEIGHT와 WIDTH 속성을 사용하여 영역의 치수를 지정한다. LEFT 속성과 TOP 속성의 값은 디스플레이 내에서 영역의 정확한 위치를 지정할 영역의 치수를 지정할 수 있으며, ZINDEX 속성은 몇 개의 영역이 서로 겹쳐져 있는 경우에 영역의 순서를 지정한다. 이 밖에도 INDEX DICTIONARY를 구성하는 엘리먼트로는 다음절에서 기술할 <RDF>와 <TEXT>, <TEXTSTREAM> 등이 있다.

2.2 RDF

RDF(Resource Description Framework)는 메타데이터의 기술과 교환을 위한 구조로서 웹 상의 메타데이터를 지원하는데 필요한 구조를 정의하기 위해 W3C에서 제안한 표준이다[6,7]. RDF는 상이한 메타데이터간의 구문 및 구조에 대한 공통적인 규칙을 지원하는 메커니즘을 통하여 웹 상에서 존재하는 기계 이해형(machine-understandable) 정보의 교환을 위해 구조화된 메타데이터간의 상호운용성(interoperability)을 지원하는 새로운 개념이라 할 수 있다. 그러므로, RDF는 인터넷상에 존재하는 상이한 성격의 메타데이터간의 상호운용이 가능하도록 하는데 그 목적이 있다.

RDF는 더블린 코어나 워킹 구조 등과 같이 웹 상에서 메타데이터를 유동적으로 지원하기 위해 발생한 메타데이터 회의의 산물이라고 할 수 있다. 실제로 RDF는 PICS의 명세뿐만 아니라 더블린 코어나 워킹 구조로부터 영향을 받았다. 현재는 W3C의 RDF 작업 그룹의 주도하에 그 연구가 행해지고 있으며, 넷스케이프사나 마이크로소프트사 등과 같은 W3C의 회원들도 그 개발 및 응용에 동참하고 있다. RDF는 현재 계속적으로 개발 중에 있기 때문에 데이터 모형이나 구문 등의 내용이 인터넷 드래프트의 형태로 발표되고 있으며, 계속적인 수정과 보완 작업이 이루어질 예정이다.

RDF는 웹 자원의 자동 처리를 가능하게 하는 기능을 강조하고 있다. 그러므로, RDF 메타데이터는 자원검색, 자원편류, 지능 소프트웨어 에이전트, 내용순위 부여 및 평가, 디지털 서명과 같은 다양한 분야에 응용될 수 있다. 자원검색 분야에서 검색엔진은 메타데이터가 포함하는 특정 분야의 내용을 쉽게 검색할 수 있으며, 또한 정확한 검색 결과를 얻을 수 있다.

3. SMIL 플레이어의 설계 및 구현

본 장에서는 검색 기능을 갖는 SMIL 플레이어의 개념적 흐름과 전체구성을 살펴보고, 각각의 구성요소에 관한 모듈별 기능들을 설명한다.

3.1 SMIL 플레이어의 개념적 흐름

사용자는 본 논문에서 제시하는 검색 기능을 갖는 SMIL 플레이어를 통해 웹이나 로컬 스토리지 상에서의 SMIL 문서들로부터 원하는 SMIL 문서를 검색하고 렌더링할 수 있는데, 전체적인 시스템의 개념적 흐름은 그림 1과 같다.

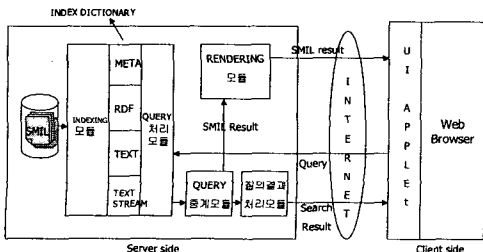


그림 1. SMIL 플레이어의 개념적 흐름

사용자는 그래픽 사용자 인터페이스라고 할 수 있는 UI APPLET을 통해 원하는 콘텐츠를 포함하는 SMIL 문서나 특정 SMIL 문서에 원하는 부분을 질의한다. 입력된 질의는 QUERY 처리 모듈 쪽으로 넘어가서 사용자로부터 입력받은 질의 내용에 따라 INDEXING 모듈로부터 생성된 INDEX DICTIONARY의 내용을 검색하여 사용자가 원하는 결과를 출력하게 된다. INDEX DICTIONARY는 SMIL 문서의 <META>, <RDF>, <TEXT>, <TEXTSTREAM> 등의 엘리먼트를 파싱, 분석하여 불용어를 처리한 후 생성된다. 이때 QUERY 중계 모듈은 질의의 결과로 넘어온 질의 결과를 질의 결과 처리 모듈에 넘겨서 질의 결과만 사용자에게 출력할 것인지, 아니면 질의 결과와 사용자에게 부합되면 질의 결과의 출력과 더불어 해당 SMIL 문서를 사용자에게 출력할 것인지를 결정해 사용자가 원하는 방식으로 질의 결과를 처리하게 된다. SMIL 문서에 포함된 다양한 형식의 멀티미디어 콘텐츠들의 특성상 사용자가 직접 검색하기엔 부담이 될 수 있기 때문에 이러한 형태의 질의 처리과정이 필요하다.

3.2 SMIL 플레이어의 구성

본 논문에서 제시하는 검색 기능을 갖는 SMIL 플레이어의 전체 구조는 그림 2와 같다.

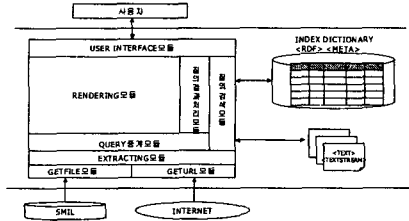


그림 2. SMIL 플레이어의 전체 구조

SMIL 플레이어는 크게 GETFILE 모듈, GETURL 모듈, EXTRACTING 모듈, QUERY 중계 모듈, 질의 결과 처리 모듈, RENDERING 모듈, 질의 검색 모듈 등으로 구성되며, 그 각각에 대한 설명은 다음과 같다.

3.2.1 GETFILE·GETURL 모듈

GETURL 모듈은 웹이나 로컬 저장소에서 SMIL 문서를 추출하는 모듈로서 웹으로부터 SMIL 문서를 읽어 들이기 위해서 먼저 확장자가 \*.smil 이거나 \*.smi 인 문서의 웹 주소를 나타내는 URL 객체를 생성한다. 그런 다음 URL을 읽어 들이고 사이트에 접속하는 URLConnection 객체를 생성한 후 URLConnection 객체의 getInputStream() 메소드를 이용하여 URL로부터의 데이터 스트림을 읽을 수 있는 InputStreamReader를 생성한다. 또한, 이와 같이 생성된 InputStreamReader를 이용하여 효율적으로 입력 스트림으로부터 데이터를 읽어 들이는 BufferedReader 객체를 만든다. GETFILE 모듈은 자바의 java.awt.FileDialog를 사용하여 사용자가 SMIL 문서를 로컬 저장소로부터 선택할 필요가 있을 때 사용되는 모듈이다.

3.2.2 EXTRACTING 모듈

EXTRACTING 모듈은 GETURL·GETFILE 모듈에서 넘겨진 로컬이나 웹상의 SMIL 문서를 SAX를 통해 파싱하고 파스트리를 형성한다. SAX는 파싱시 엘리먼트 단위로 처리하고 이벤트에 기반하여 동작하는 구조를 취하기 때문에 본 논문에서는 이를 통해 이벤트와 관련된 이벤트 테이블을 만들고 RENDERING 모듈에게 이벤트를 발생시켜 처리하도록 하는 방법을 사용한다. 즉, SMIL 문서의 멀티미디어 콘텐츠들을 렌더링 해야 하는 특성상 이벤트 기반의 파싱 방법을 사용하여 메모리 낭비와 필요한 노드를 찾기 위한 불필요한 프로세싱 시간을 줄였다. 본 논문에서는 system 프로퍼티에 지정되거나 구현시 지정한 파서를 생성하는 makeParser() 메소드를 사용하여 위해 ParserFactory 클래스를 사용하였고, 이벤트 핸들러를 정의하거나 DTD 정의 등을 처리하기 위해 DocumentHandler, ErrorHandler, DTDHandler, EntityResolver 등의 클래스를 사용하였다.

### 3.2.3 QUERY 중계 모듈

QUERY 중계 모듈은 앞에서 언급한 바와 같이 질의 결과로 넘어온 질의 결과를 질의 결과 처리 모듈에 넘겨서 질의 결과만 사용자에게 출력해 줄지 아니면 질의 결과가 사용자에게 부합되면 질의 결과의 출력과 더불어 해당 SMIL 문서를 사용자에게 출력해 줄지를 결정해 사용자가 원하는 방식으로 질의 결과를 처리한다.

### 3.2.4 질의 결과 처리 모듈

질의 결과 처리 모듈은 QUERY 중계 모듈로부터 넘겨받은 질의 결과를 UI APPLET 을 통하여 사용자에게 출력한다.

### 3.2.5 RENDERING 모듈

RENDERING 모듈에서는 GETURL, GETFILE 모듈과 EXTRACTING 모듈에서 파싱된 미디어클립을 재생하는 모듈로써 SMIL 문서의 엘리먼트 중 RENDERING 모듈에서 처리될 엘리먼트는 <TEXT>, <TEXTSTREAM>, <AUDIO>, <ANIMATION>, <VIDEO> 등이 있다. 이 모듈의 세부모듈은 그림 3 과 같다.

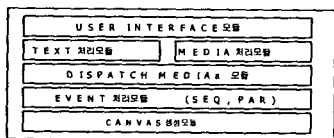


그림 3. RENDERING 모듈의 세부모듈

먼저 CANVAS 생성 모듈은 SMIL 문서에서 파싱된 <HEAD> 엘리먼트의 하위 엘리먼트인 <LAYOUT> 엘리먼트의 위치와 관련된 WIDTH, HEIGHT, LEFT, TOP 등의 속성을 이용하여 미디어클립을 재생할 수 있는 사각형의 점프포인트를 생성한다. EVENT 처리 모듈에선 SMIL 문서의 <SEQ> 나 <PAR> 엘리먼트를 파싱하여 렌더링될 미디어클립들이 순서대로 렌더링될 것인지 아니면 병렬로 처리될 것인지를 판별하게 된다. 만약에 <SEQ> 와 같이 순서대로 렌더링하여야 할 경우에는 하나의 TEXT 처리 모듈이나 MEDIA 처리 모듈을 순서대로 호출하고, <PAR> 엘리먼트일 경우에는 <TEXT> 와 <TEXTSTREAM> 엘리먼트의 미디어클립은 TEXT 처리 모듈에서, 그 밖의 <AUDIO>, <ANIMATION>, <VIDEO> 엘리먼트의 미디어클립은 MEDIA 처리 모듈에서 렌더링하게 된다. TEXT 처리 모듈에서의 렌더링은 버퍼를 형성하여 형성된 버퍼에 미디어클립을 읽어 들인 후 CANVAS 생성 모듈에서 생성된 위치에 drawString() 메소드를 사용하여 버퍼가 낡을 때까지 문자열을 찍어주는 형식으로 이루어진다. MEDIA 처리 모듈은 JMF API 인 java.media.\* 를 импорт 한 후 JFrame 을 상속하고, 미디어의 종류에 따라 createPlayer() 메소드를 통해 미디어에 맞는 player 객체를 생성한 후 start() 메소드로 렌더링하게 된다. USER INTERFACE 모듈에선 사용자 인터페이스를 처리한다.

### 3.2.6 질의 검색 모듈

인덱스는 검색 모듈에서 가장 핵심적인 부분으로 형태소 분석에서 추출된 명사 리스트와 메타 정보인 RDF를 특정한 형태의 화일 구조로 저장한다. 본 논문에서 인덱스 화일의 화일 구조는 다양한 경로에서의 신속한 질의 처리를 위해서 Inverted file 기법을 이용한다. 질의 검색 모듈의 세부모듈 및 처리 과정은 그림 4 와 같다.

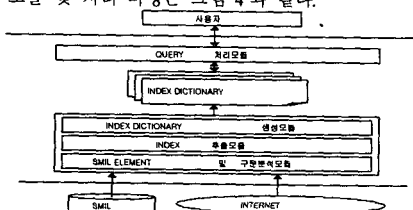


그림 4. 질의 검색 모듈의 세부모듈 및 처리과정

SMIL 엘리먼트 및 구문분석 모듈은 먼저 입력 요소인 SMIL 원문을 입력으로 받아 파싱을 통하여 <META>, <RDF>, <TEXT>, <TEXTSTREAM> 등과 같은 질의 검색 모듈에서 의미를 갖는 엘리먼트를 가지고 분석 과정을 수행한다. 인덱스 추출 모듈은 인덱스 키워드로써 가치가 없는 단어들을 불용어 처리를 한 후 처리된 인덱스 키워드를 가지고 INDEX DICTIONARY 생성 모듈에서 INDEX DICTIONARY 화일을 생성하게 된다. 이와 같이 생성된 INDEX DICTIONARY 화일을 가지고 질의를 처리하게 되는 부분이 QUERY 처리 모듈이다. 사용자는 특정 카테고리, 날짜, 저자 등의 옵션을 가지고 질의를 수행하는데 이러한 사용자의 요구를 수용하기 위하여 QUERY 처리 모듈은 SMIL 문서의 기본적인 엘리먼트 정보 이외에 풍부한 RDF 메타 정보를 활용하여 사용자가 원하는 SMIL 문서를 찾아주는 역할을 한다.

## 4. 결론 및 향후 연구과제

인터넷의 빠른 성장과 더불어 사용자의 웹 상에서의 멀티미디어에 관한 욕구도 증가하게 되었다. 이러한 요구에 맞춰 W3C에서는 XML의 형식을 기반으로 한 SMIL이라는 XML 어플리케이션을 개발하여 표준화 하게 되었고 이로써 개발자들과 사용자들은 좀더 쉽게 다양한 형식으로 웹 상에서의 프리젠테이션이 가능하게 되었을 뿐만 아니라 마치 T.V 같은 인터넷 환경을 기억하게 되었다.

본 논문에서는 이러한 SMIL 문서의 렌더링과 관련된 프리젠테이션뿐만 아니라 검색 기능을 추가함으로써 SMIL 문서의 응용의 폭을 넓게 하였다. 본 논문에서 제시한 검색 기능을 갖는 SMIL 플레이어는 방송국과 같은 방대한 양의 멀티미디어 데이터를 처리하는 시스템이나 교육용 멀티미디어 데이터를 다루는 시스템 등에서 기존의 멀티미디어 데이터베이스를 SMIL 환경으로 교체할 경우 적합할 것이다. 향후 연구 과제는 웹 상에서 자동으로 SMIL 문서를 수집하는 SMIL을 위한 웹 에이전트 기반의 검색 기법에 관한 연구가 필요하다.

## 참고문헌

- [1] Dao, T., and Davis, R., S., "Indexing Structured Text for Queries on Containment Relationships," Proc. of the 7th Australian Database Conf., 1996, pp. 331 - 336.
- [2] Megginson, *The Simple API for XML for DOM*, 1999.
- [3] Pitts, N., *XML in Record Time*, Sybex, 1999.
- [4] Sun, *Java Media Framework API Guide 2.0*, 1999.
- [5] W3Consortium, *Accessibility Features of SMIL*, 1999.
- [6] W3Consortium, *Resource Description Framework (RDF) Model and Syntax Specification*, 1999.
- [7] W3Consortium, *Resource Description Framework (RDF) Schema Specification 1.0*, 2000.
- [8] W3Consortium, *Synchronized Multimedia Integration Language (SMIL) 1.0 Specification*, 1998.
- [9] W3Consortium, *Synchronized Multimedia Integration Language (SMIL) Boston Specification*, 2000.
- [10] Yu, J., and Xiang, Y., "Hypermedia Presentation and Authoring System," Proc. of the 6th Int. WWW Conf., 1997, pp. 153-164.
- [11] 구우석, 한기준, "VOD 시스템을 위한 영상 정보 검색기," 한국정보과학회 학술발표 논문집, 23권2호, 1996, pp. 33-35.
- [12] 장계우, "SGML 정보검색을 위한 인덱스 관리자의 설계 및 구현," 한국정보과학회 학술발표 논문집, 5권2호, 1999, pp. 135-146.