

SMIL 2.0 문서 재생을 위한 이벤트 처리기의 설계 및 구현

김혜은[†], 채진석^{**}, 이재원^{***}, 김성동^{****}, 이종우^{*****}

요 약

SMIL(Synchronized Multimedia Integration Language)은 W3C에서 제안한 멀티미디어 프리젠테이션에 관한 국제 표준 언어이다. 이것은 개별적으로 존재하는 수많은 멀티미디어 데이터들을 통합하여 새로운 멀티미디어 데이터를 만들 수 있고, 기존의 웹 브라우저에서는 불가능했던 멀티미디어 데이터들의 동기화를 수행할 수 있다는 장점을 가지고 있다. 이러한 SMIL 기술을 이용하고 활성화하려면, QoS를 보장할 수 있는 멀티미디어 서버, SMIL 저작 도구, SMIL 재생 도구의 개발이 요구된다. 관련 도구의 개발을 위해서는 SMIL 문서 형태인 텍스트 파일을 읽어서 해석하고, 문서에 나타나는 여러 형태의 미디어 객체들을 시간적인 동기화 관계에 따라 재생할 수 있는 기술이 필수적으로 필요하다. 따라서 본 논문에서는 시간적인 동기화 관계에 따라 미디어들의 재생이 가능하고, 재생 중 이벤트 처리가 가능하여 SMIL 2.0 시간 모델을 만족하는 이벤트 처리기를 설계 및 구현하였고, 이 이벤트 처리기를 이용하여 SMIL Basic 프로파일을 만족하는 재생기를 개발하였다. 이것은 SMIL 콘텐츠를 쉽게 재생할 수 있게 하여 SMIL의 활성화에 기여할 수 있고, SMIL 표준안에 정의된 여러 종류의 프로파일에서 재사용될 수 있어서 XHTML+SMIL이나 SMIL Animation과 같은 SMIL을 통합하는 다른 표준들에 응용될 수 있을 것으로 예상된다.

A Design and Implementation of Event Processor for Playing SMIL 2.0 Documents

HyeEun Kim[†], Jinseok Chae^{**}, Jae Won Lee^{***},
Sung-Dong Kim^{****}, Jongwoo Lee^{*****}

ABSTRACT

The Synchronized Multimedia Integration Language (SMIL), recommended by the World Wide Web Consortium (W3C) in 1998, is an XML-based declarative language to synchronize and present multimedia documents. SMIL can create new multimedia data integrating various types of multimedia objects which exist separately such as text, video, graphics and audio. It can support synchronization of multimedia data which are limited in current HTML-based Web technology. For its popularity, it is required to develop a multimedia server guaranteeing Quality of Service (QoS), authoring tool and player. For developing a SMIL authoring tool and player, the technologies are essentially required to read and analyze a SMIL document and to play synchronized various types of media objects in a timeline. In this paper, we describe a design and implementation of an event processor which supports SMIL 2.0 timing model. Moreover, we also develop a SMIL 2.0 player using the proposed event processor. This will facilitate the play of SMIL contents, so that it can contribute to the prosperity of SMIL technology. It is possible to reuse in various language profiles defined in the SMIL standard. This player is expected to be utilized in other standard integrating SMIL such as XHTML+SMIL and SMIL Animation.

Key words: SMIL, SMIL player(SMIL 재생기), event processor(이벤트 처리기)

※ 교신저자(Corresponding Author): 채진석, 주소: 인천광역시 남구 도화동 177(402-749), 전화: 032)770-8427, FAX: 032)766-6894, E-mail: jschae@incheon.ac.kr
접수일: 2003년 2월 14일, 완료일: 2003년 7월 22일

[†] 비회원, (주)에니스틸닷컴 연구원

(E-mail: heykim@anysteel.com)

^{**} 중신회원, 인천대학교 컴퓨터공학과 조교수

^{***} 비회원, 성신여자대학교 컴퓨터정보학부 전임강사

(E-mail: jwlee@cs.sungshin.ac.kr)

^{****} 비회원, 한성대학교 컴퓨터공학부 조교수

(E-mail: sdkim@hansung.ac.kr)

^{*****} 비회원, (주)아이닉스 소프트 개발이사

(E-mail: jwlee44@daisy.kw.ac.kr)

※ 이 논문은 한국과학재단 지정 인천대학교 동북아전자물류연구센터의 지원에 의한 것임.

1. 서 론

SMIL[1]이란 Synchronized Multimedia Integration Language의 약어로, 1998년 4월 W3C (World Wide Web Consortium) 멀티미디어 워킹 그룹 (Working group)이 제안한 멀티미디어 프리젠테이션에 관한 국제 표준 언어이다. 이것은 웹상에서 다양한 멀티미디어 데이터를 동기화하여 표현하는 방법을 제공하며 1998년 SMIL 1.0이 나온 후에 계속적으로 발전하였다. 2000년 6월에는 SMIL 2.0 표준이 나왔고 이 표준은 양적인 면이나 내용적인 면 모두 크게 확장되고 발전된 것이었다.

SMIL이 가지고 있는 장점은 다음과 같다[1-6].

① XML(eXtensible Mark-up Language) 기반의 문서 형식이라 사용자들이 쉽게 문서를 이해할 수 있고, 일반 텍스트 에디터로도 SMIL 문서를 작성할 수 있다.

② 모듈화 개념을 사용하는 XML 기반 언어이므로, 네임스페이스를 이용하여 다른 XML 기반 언어들과 통합하여 사용할 수 있다.

③ 여러 가지 형식의 파일들 즉, 스트리밍 오디오, 스트리밍 비디오, 이미지, 텍스트 등의 미디어 파일들을 효과적으로 통합함으로써 보다 쉽고 효율적으로 멀티미디어 콘텐츠를 제작할 수 있으며, 이미 웹에 독립적으로 존재하는 수많은 멀티미디어 데이터들을 재사용하여 새로운 멀티미디어 콘텐츠를 만들 수 있다.

④ 멀티미디어 데이터들을 공간적, 시간적으로 자유롭게 배치시킬 수 있고, 각각의 미디어들에 이벤트를 주어 사용자와의 상호작용을 높일 수 있다.

⑤ RTSP(Real Time Streaming Protocol) 등의 실시간 전송 프로토콜을 이용하여 스트리밍 멀티미디어 데이터의 재생이 가능하다.

⑥ switch 요소를 이용하여 다양한 대역의 사용자에게 각기 다른 멀티미디어 데이터를 전송할 수 있기 때문에, 사용자의 네트워크 환경에 맞는 서비스를 제공할 수 있다.

SMIL 2.0의 특징은 데스크탑 PC나 휴대형 단말기에 초점을 맞추어서, 동기화된 멀티미디어 데이터를 전송하고 이용할 수 있도록 하고 있는데, 여러 가지의 프로파일을 구성할 수 있도록 정의하여, 각각의 단말기 환경에 맞는 프로파일을 사용할 수 있도록 하고 있다. 이 외에도, 사용자의 상황에 맞게 프리젠테이션을 중단하거나 계속할 수 있도록 대화형 기능이 대폭 강화됐다. 앞으로 SMIL 기술은 인터넷 방송과 역동적인 광고 콘텐츠 제작, 사용자의 수준에 따라 콘텐츠를 선택적으로 제공할 수 있게 하는 온라인 교육 자료의 제작 등 다양한 분야에서 응용되어 사용할 수 있을 것으로 예상된다. 또한, SMIL은 현재 웹뿐만 아니라 모바일 쪽에서도 새로운 표준으로 자리매김하고 있어서[7], 더욱 활발하게 사용될 것으로 보인다.

SMIL 기술을 이용하기 위해서는 SMIL 문서 형태인 텍스트 파일을 읽어서 해석하고 문서에 나타나는

여러 형태의 미디어 객체들을 시간적인 동기화 관계에 따라 재생할 수 있는 기술이 필요하다. 또한, 올바른 동기화 관계에 따라 재생할 수 있도록 QoS(Quality of Service)를 보장할 수 있는 멀티미디어 서버가 필요하며, 클라이언트 측에서 SMIL로 문서를 제작하고 사용할 수 있도록 저작도구의 개발과 함께 재생도구의 개발이 필수적으로 요구된다. 그러나 아직까지 SMIL에서 제시하고 있는 시간 모델에 대한 연구는 미흡한 실정이다.

이에, 본 논문에서는 관련 연구로서 SMIL 2.0의 표준안을 분석하고, SMIL의 핵심적인 부분인 시간 모델에 대해 알아본다. 연구 내용으로 시간 모델을 이용하여 미디어들의 동기화된 시간을 계산하기 위한 이벤트 처리 및 스케줄링 방법을 제시하고, 이 방법을 이용하여 구현된 SMIL 재생기에 대해 기술한다. 마지막으로 결론과 향후 연구 방향을 제시한다.

2. 관련연구

2.1 SMIL 2.0의 개요

SMIL이란 웹의 콘텐츠에 멀티미디어를 통합하기 위해 XML에 기반을 두어 접근한 표준이다. SMIL 프리젠테이션은 오디오, 비디오, 이미지, 텍스트 등의 다양한 멀티미디어 객체들을 시간 기반으로 조합하여, 화면상의 원하는 위치에 보여줄 수 있다. 예 1은 SMIL 문서의 구조를 보여주고 있다. 이렇게 SMIL 문서에는 보여주고자 하는 미디어 파일들의 파일명이 무엇이며, 어디에 위치하고 있는지, 어느 시간에 화면에 보이거나 소리로 들려야 하는지 등이 나타나게 된다. 즉, SMIL은 멀티미디어 데이터 자체를 해석하는 것이 아니라 이미 존재하는 여러 종류의 멀티미디어 객체를 시간적, 공간적으로 배치하여 표현하는 방법이다.

SMIL을 사용하게 되면 여러 가지 형태의 멀티미디어 프리젠테이션을 생성할 수 있으며, 다양한 네트워크 대역을 가지는 사용자에게 각기 다른 멀티미디어 콘텐츠 서비스를 선택적으로 제공할 수 있고, 트랜지션(Transition), 애니메이션(Animation)과 같은 화려한 화면 변환 효과를 제공할 수 있는 장점을 가지고 있다. 또한, SMIL의 강력한 시간 모델은 멀티미디어 객체들을 사용자의 요구에 따라 동적이고 세련되게 반응하도록 한다.

SMIL 2.0은 기능별로 50개의 모듈과 하나 이상의 모듈로 구성된 10개의 그룹으로 나누어진다. 각 모듈에 따라 기능을 나타내는 여러 개의 요소(Element) 및 속성(Attribute)을 정의하고 있는데, 이들은 프로파일(Profile)을 정의하는 데 사용된다. SMIL 2.0에서는 표준안에 정의된 모든 모듈을 반드시 사용할 필요 없이, 지원하고자 하는 모듈만을 정의해 사용할도록 하는 프로파일링 기능을 제공하고 있다. 따라서 자신의 장비나 환경에 맞도록 적절한 SMIL 응용 도구를 개발하고 사용할 수 있게 된다.

```
<smil xmlns="http://www.w3.org/2001/SMIL20/Language">
<head>
<layout><region id="left" top="0" left="0" height="240"
width="320" /></layout>
</head>
<body>
<par begin="2s" end="20s">

</par>
</body>
</smil>
```

예 1. SMIL 문서의 예

2.2 SMIL Time Model

SMIL에서 가장 핵심적인 부분은 각 멀티미디어 요소들을 동기화시켜 표현하는 것이다. 시간과 동기화 제어를 위해서 따로 정의된 요소는 없으며, 속성들을 이용하여 표현한다[8].

객체가 실행되는 시간을 효율적으로 나타내기 위해서 9개의 보조적인 시작, 종료, 재생 시간을 이용하게 된다[9].

- ① 묵시적인(Implicit) 시간 : begin, end, dur
- ② 명시적인(Explicit) 시간 : begin, end, dur
- ③ 요구되는(Desired) 시간 : begin, end, dur

“묵시적인” 시간은 객체가 고유하게 가지고 있는 시간을 의미한다. 속성에 시간을 따로 명시하지 않았다면 묵시적인 시간을 재생에 이용하게 되는데 비디오 객체와 같이 연속적인 미디어의 경우는 고유의 재생 시간이 묵시적인 재생 시간이 되고, 이미지 객체와 같이 불연속적인 미디어의 경우는 고유의 재생 시간이 없으므로 묵시적인 재생 시간이 0이 된다.

“명시적인” 시간은 객체가 가지고 있는 고유의 시간에는 상관없이 사용자가 속성을 통해서 지정한 시

간을 의미한다. 이것은 묵시적인 시간보다 우선순위가 높다.

마지막으로 “요구되는” 시간은 부모 요소나 또는 다른 요소에 의해 제약을 받는 시간이다. SMIL 시간 모델에서는 자신의 재생은 부모 요소가 재생되는 범위 내에서만 유효하도록 정의하고 있다. 따라서 자신의 재생 시간이 명시되었다 하더라도, 부모의 재생 시간을 초과할 수 없다.

이러한 9가지의 시작, 지속, 종료 시간을 이용하여 수많은 시간 계산 규칙이 정해지게 된다. 이러한 여러 가지 규칙들은 다양한 프리젠테이션 효과를 낼 수 있다는 장점이 있지만, 이에 비해 시간 계산을 복잡하게 하여 직관적으로 이해하기 어렵고 해석하는 사람들로 하여금 혼돈을 일으키는 경우도 발생하는 단점이 있다. 예 2를 살펴보면, 사용자의 입장에서 볼 때 “B”의 이미지는 비디오가 반복될 때마다(0s, 8s, 16s) 시작될 것인지, 아니면 비디오가 처음 시작될 때 한 번만 시작될 것인지 혼동이 오게 된다. SMIL 1.0은 이러한 경우에 대해서 명확한 해석을 제시하지 않았었지만, SMIL 2.0에서는 한 번만 시작하도록 명확하게 정의하고 있다.

이 외에, SMIL 2.0에는 동기화 제어나 XML 기반 언어와의 통합 설정을 위한 속성들이 추가되었다. 그리고 고정된 시간에 의한 동기화와 이벤트 기반 동기화를 제공하고 있고, 동기화된 표현을 위하여 Instance begin/end 리스트, Interval life cycle 모델, pruning/cutting off 개념을 이용하고 있다.

```
<par>
<video id="A" dur="8s" repeat="3" />
<img id="B" begin="A.begin" dur="5s" />
</par>
```

예 2. 반복 속성이 정의된 예

2.2.1 고정된 시간을 이용한 동기화

이것은 SMIL에서 미디어 요소들을 동기화시키는 방법이다. 시간을 나타내는 속성인 begin, end, dur 등을 이용하여 몇 초 혹은 몇 시에서 시작하여 얼마만큼의 재생 시간을 가지고 언제 끝나는지 등을 직접 시간으로 지정한다. 이 때 나타낼 수 있는 시간 형식은 초, 분, 시간 단위 등을 자유롭게 지정할 수 있고 기본값은 초 단위를 사용한다. 직접 시간을 기술하는

방법 외에 기본적인 동기화를 나타낼 수 있는 다른 방법은 다른 요소의 속성에 의존해서 나타내는 것이 있다.

2.2.2 이벤트에 의존하는 시간을 이용한 동기화

SMIL 2.0에서는 사용자의 상황에 맞게 프리젠테이션을 중단하거나 계속할 수 있는 대화형 기능이 대폭 강화되었다. begin, end, dur 속성에 시간이 아닌 이벤트를 기술하면, 이벤트의 발생에 따라 미디어 요소의 재생이 제어된다. 정의된 이벤트를 살펴보면 다음과 같다.

① *beginEvent*, *endEvent*, *repeat*와 같은 DOM (Document Object Model) 메소드에 기반을 둔 이벤트

② 재생되는 도중에 키보드, 마우스 등의 입력 장치들로부터 포커스를 얻거나 잃게 되는 경우에 발생하는 이벤트

③ 재생되는 미디어들의 공간이 속한 topLayout 요소의 활성화 여부에 따라 발생하는 이벤트

2.2.3 Instance Begin/End Times 리스트

미디어 요소들의 시작 시간과 종료 시간을 저장하기 위하여 시간을 담고 있는 instance times 리스트를 각 미디어 요소별로 생성한다. 시작 시간은 begin 속성에 의해 결정되고, 종료 시간은 end 속성 외에 dur, repeat, min, max 속성 등이 복합적으로 계산되어 결정된다. 만일 이 시작/종료 시간이 이벤트나 다른 요소에 의존적으로 결정되어야 할 경우에는 하나가 아니라 여러 개의 값이 만들어 질 수 있으므로 리스트를 사용하여 생성된다.

2.2.4 Interval Life Cycle 모델

각 멀티미디어 요소들은 재생시에 instance begin/end times 리스트에 저장된 시작 시간과 종료 시간을 이용하여 실시간으로 interval을 생성하여 이 값을 이용하게 된다. 요소에 시작/종료 시간이 여러 개 있다면 interval도 여러 개가 생성되지만, 하나의 시점에서 재생에 사용되는 interval은 하나이다. 이 값들은 의존하고 있는 요소의 이벤트 발생 여부에 따라 값이 변경될 수 있고, 현재 시점에서 재생에 이용되지 않는다면 삭제될 수도 있다.

2.2.5 Pruning/Cutting Off 개념

Pruning과 Cutting off는 상황이 발생되었을 때,

해당 요소를 화면에 나타내지 않는다는 점은 같지만 그 의미가 다르다. Cutting off가 발생하는 것은 시작 시간 또는 종료 시간이 음수 값을 가지고 있을 때이다. 이 때는 해당 요소가 화면에 보이지는 않으나, 논리적인 오류가 있는 것이 아니므로 시작/종료 시간에 의존하고 있는 요소들에게 이벤트를 전해줄 수 있다. 반면에 Pruning이 발생하는 경우는 종료 시간이 시작 시간보다 빠르거나 하는 의미적으로 오류가 발생했을 때이다. 이 때는 화면에 보이지 않을 뿐만 아니라 재생과 관련된 이벤트 또한 발생되지 않는다.

2.3 기존 SMIL 재생 도구들

아직도 일반 사용자들에게 SMIL은 낯설게 느껴지고 있지만, SMIL 표준안을 지원하는 저작 도구, 재생 도구들이 발표되고 있다[1]. SMIL을 지원하는 대표적인 회사로는 Adobe, Microsoft, RealNetworks 등이 있으며, 이 외에도 크고 작은 업체들이 SMIL을 지원하고 있다[10].

RealNetworks사는 SMIL를 가장 적극적으로 지원하고 있는 업체이다. *RealONE Player*[11]는 대표적인 SMIL 재생 도구이고, 2001년 9월에는 *SMILGen*이라는 SMIL 저작도구를 발표하였다. *RealOne Player*의 경우, 무료이기 때문에 쉽게 접하고 테스트 해 볼 수 있다.

마이크로소프트사는 *IE 5.0*부터 SMIL을 포함하기 시작하였고, 현재 *IE 6.0*에는 HTML+TIME 형태로 SMIL를 지원하고 있다. HTML+TIME이란 HTML 문서 내에 시간 속성을 더한 것으로 HTML 문서에 추가적으로 네임 스페이스를 지정하고, 새로운 요소와 속성을 사용한다. 이 때 화면의 영역 설정은 주로 CSS(Cascading Style Sheets)를 이용하여 나타내고, 시간 속성은 SMIL을 이용하여 나타낸다.

Oratrix사 또한 일반적인 SMIL 저작 도구인 *GRiNS Pro Editor for SMIL 2.0*, *RealOne* 재생을 위한 저작 도구인 *GRiNS Editor for RealOne*, 자체적인 SMIL 재생 도구인 *GRiNS Player for SMIL 2.0* 등 많은 SMIL 관련 도구들을 제공하고 있다[8,12].

이 밖에도 Adobe의 *GoLive Studio*, Confluent Technologies의 *Fluition* 등이 윈도우 기반의 SMIL 툴을 제공하고 있고, 자바 기반의 *X-Smiles*는 XHTML, SVG, XForms 등의 웹 표준들과 더불어 SMIL 2.0 Basic Profile을 지원한다. PDA와 같은 휴

대용 장치를 위한 SMIL 재생 틀인 *InterObject* 등도 있다.

2.4 다른 멀티미디어 표현 형식들과의 비교

웹에서 멀티미디어 데이터를 표현하는 것은 많은 잠재력을 가지고 있기 때문에 많은 연구와 개발이 이루어지고 있으며, 현재 SMIL 외에도 여러 형식들이 존재한다[13].

2.4.1 Flash

웹상에서 미디어들의 동적인 표현을 위해 현재 많이 사용되고 있는 Flash는 매크로미디어사의 독점적인 미디어 데이터 표현 형식이며[14], 공식적인 표준안은 아니다. SMIL은 미디어 데이터 표현 형식이 아니라 이미 존재하는 미디어들에 대해 동기화되는 시간 속성들을 추가하여 텍스트 문서로 기술한다는 점에서 Flash와 다르다. SMIL 문서는 Flash 파일 (*.swf)을 포함시켜 표현할 수 있다[15].

하나의 미디어 파일을 윈도우 또는 모바일과 같은 다른 디바이스 환경에서 보여주고자 할 경우, SMIL은 텍스트 문서의 수정만으로 가능하지만 Flash는 각 환경마다 다른 파일을 만들어줘야 하는 차이점이 있으며, SMIL은 다른 XML 호환 언어들과 통합이 가능하지만 Flash는 독점적인 형식으로 확장성이 부족하다는 단점이 있다.

2.4.2 MPEG-4

MPEG-4(Moving Picture Experts Group)는 미디어를 표현하는 것뿐만 아니라 미디어 객체의 내용과 상호 작용을 기술하기 위해 정의된 광범위한 프로토콜의 집합이다. MPEG-4는 이진 데이터 표현 형식이며 저급 레벨의 표현 방식인 반면에 SMIL은 XML 기반 언어로서 사람이 읽기 쉽게 고급 레벨로 기술한다는 차이가 있다. 또한 MPEG-4는 표현될 내용을 담고 있는 최종적인 데이터 형식인 반면에 SMIL은 실질적으로 데이터를 담고 있지 않다.

현재 XMT(eXtensible MPEG-4 Textual) 표준안을 통하여 SMIL뿐만 아니라 X3D(eXtensible 3D)와 MPEG-4 사이의 상호 운용을 가능하게 하기 위해 많은 연구가 진행 중이다[16].

3. 이벤트 처리기의 설계 및 구현

SMIL에는 각 미디어 원소들을 그룹지어 표현하

는 <par>, <seq>, <excl> 등의 시간 그룹 요소들이 있고, <video>, <audio>와 같은 시간기반 미디어들이 혼합되어 존재하므로 이들을 동기화시킬 방법이 필요하다. 또한, SMIL 2.0에서는 DOM 메소드를 이용한 이벤트 기반으로 미디어 원소들을 동기화하는 방법을 제공함으로써 인해 시간 모델이 더욱 복잡하게 되었다. 따라서 SMIL 재생기에서는 미디어들을 올바르게 표현하기 위한 타이밍 및 동기화 기능(Timing and Synchronization)을 지원하는 스케줄러의 제작이 필수적으로 요구된다.

스케줄러에서는 시간이 고정된 미디어들의 재생 시간뿐만 아니라, 재생 중 발생하는 이벤트들로 인한 미디어들의 재생 시간 변화까지 인식하여야 한다. 따라서 본 논문에서는 두 단계로 나누어 스케줄링을 수행하고 두 번째 단계에서 이벤트를 처리하도록 하는 방법을 제시한다. 제 1단계인 Prefetch Schedule 단계는 재생이 시작되기 전에 수행되는 것으로 미디어 원소들마다 알려진 시간들을 기초로 스케줄을 작성한다. 알려진 시간이란 SMIL 문서 작성자가 문서 내에 속성으로 기술한 시간을 말한다. 제 2단계인 Main Schedule 단계는 재생 중에 수행되는 것으로 해결되지 않았던 값들이나 이벤트에 의해 발생하는 시간들을 실시간으로 계산하여 스케줄을 작성하고 다시 재생에 이용하게 된다[17].

3.1 Prefetch Schedule 단계

이 단계에서는 각 미디어 요소들마다 상수로 정의된 값이나 오프셋으로 정의된 속성 값들을 이용하여 시작/종료 시간을 구하고, 재생시 필요한 시간을 재구성한다. 또한 이벤트 의존 시간들에 대한 내용도 저장된다.

본 논문에서는 시간 재구성시 필요한 상태를 표현하기 위하여 “UNRESOLVED”, “INDEFINITE”, “isLock”이라는 새로운 용어를 정의하였으며, 이벤트 의존 요소에 대한 정보를 저장하기 위하여 새로운 XML 문서인 “Event”를 정의하였다.

“UNRESOLVED”와 “INDEFINITE”는 현 단계에서 구할 수 없는 값들을 표현한다. “UNRESOLVED”의 경우 이벤트에 의존하는 시간을 나타내고, “INDEFINITE”는 이벤트에 의존하고 있지 않지만 부모 요소 등에 의한 제약과 같은 여러 가지 이유로 인해 아직 확정되지 않은 시간을 의미한다.

“isLock”은 실행 중 시스템에 의해 값이 변할 수 있는지의 여부를 체크한다. 이것은 시간 그룹 요소들을 염두에 두고 정의한 속성이다. 고유의 simple duration, 즉 얼마동안 재생되어야 하는 지에 대한 정의가 있을 경우 자식 요소에 의해 영향을 받지 않으므로 isLock=“T”로 지정하게 된다. 만일 고유의 재생 시간이 정의되어 있지 않다면 자식 요소들의 재생 시간에 의해 자신의 재생 시간이 결정되므로 isLock=“F”로 지정하게 된다. isLock=“F”일 경우 자식 요소의 재생이 시작되거나 끝날 때마다 부모 요소의 재생 시간이 수정된다.

Event 문서는 스케줄링시 확정되지 않는 시간들 중에서 이벤트에 의존하는 시간일 경우 어느 요소가 어떤 이벤트에 의존하고 있는지에 대한 정보를 담고 있게 된다. Event 문서의 DTD는 예 3과 같다. 루트 요소는 <event>이고 이벤트 의존 요소가 발견될 때마다 하나의 <item> 요소가 생성된다. 이 요소는 from, fromKind, to, toKind 라는 네 가지의 속성을 가지는데 이 속성들을 통하여 어느 요소의 어떤 값이 무슨 이벤트에 의존하고 있는지를 알 수 있다.

```
<!ELEMENT event ( item* ) >
<!ELEMENT item ( #PCDATA ) >
<!ATTLIST item
    from CDATA #REQUIRED
    fromKind (beginEvent | endEvent | activatesEvent) "beginEvent"
    to CDATA #REQUIRED
    toKind (begin | end) "begin"
>
```

예 3. Event 문서를 위한 DTD

- ① from : 의존하고 있는 요소 명을 기록
- ② fromKind : 요소의 어떤 시간에 의존하고 있는지를 기록
- ③ to : 영향 받는 요소 명을 기록
- ④ toKind : 요소의 어떤 시간이 영향을 받는지를 기록

그 결과, 본 단계의 수행 뒤에는 본래의 SMIL 문서 외에 수정된 SMIL 문서와 Event 문서가 새롭게 생성된다.

본 단계에서 수행하는 일은 다음과 같은 속성을 참고하여 시간을 재구성한다.

- ① 기본적인 시간 제어 속성 : begin, end, dur

- ② 반복 속성 : repeatCount, repeatDur
- ③ 추가적인 제어 속성 : min, max

이 단계에서는 각 요소에서 기술되어 있는 값을 기반으로 계산이 가능한 시간만 구하고, 구할 수 없는 시간들 중에서 이벤트에 의존하는 시간일 경우에는 “UNRESOLVED”, 이벤트에 의존하지는 않지만 현재로서는 명확하게 알 수 없는 시간들에는 “INDEFINITE” 값을 지정해 준다. 예 4(a)는 원본 SMIL 문서이며 예 4(b)는 Prefetch Schedule 단계를 수행하고 난 뒤 재구성된 문서이다.

```
<par id="group">
  <img id="smile" dur="5s" repeatCount="2" />
  <video id="vid" begin="smile.endEvent" />
  <img id="frown" dur="3s" />
</par>
```

예 4(a). 이벤트 의존 시간이 기술된 SMIL 문서의 예

```
<par id="group" begin="0" end="INDEFINITE" dur="INDEFINITE" isLock="F">
  <img id="smile" dur="5s" repeatCount="2" begin="0" end="10" isLock="T" />
  <video id="vid" begin="UNRESOLVED" dur="5" end="UNRESOLVED" isLock="T" />
  <img id="frown" dur="3s" begin="0" end="3" isLock="T" />
</par>
```

예 4(b). Prefetch Schedule를 수행하고 난 뒤, 재구성된 SMIL 문서

이러한 방식으로 재생할 때 기본적으로 이용하는 속성인 begin, end, dur, repeatCount, repeatDur에 해당하는 시간을 생성한다. 이벤트 의존 시간이 발견됐을 경우에는 따로 저장하여 재생시 이용하게 된다.

예 5는 예 4(a)의 SMIL 문서가 Prefetch Schedule 단계를 끝내고 난 뒤 생성되는 이벤트 문서이다. 원본 SMIL 문서에 이벤트 의존 요소가 하나밖에 없기 때문에 <item>요소는 하나만 생성된다. 속성은 의미는 “smile” 요소(from)가 종료(fromKind)되면, 그것은 “vid” 요소(to)의 시작 시간(toKind)에 영향을 준다는 것을 나타낸다.

```
<event>
  <item from="smile" to="vid" fromKind="endEvent" toKind="begin" />
</event>
```

예 5. Prefetch Schedule를 수행하고 난 뒤 생성되는 Event 문서

3.2 Main Schedule 단계

Main Schedule 단계는 SMIL 문서의 재생이 진행되면서 다시 스케줄링되는 단계이다. 재생을 하면서 Prefetch Schedule 단계를 수행하여 생성된 SMIL 문서를 이용하여 해결되지 못한 값을 계산하고 다시 재생에 이용하는 과정을 거치게 된다.

여기서는 <body> 요소의 첫 번째 자식, 즉 시간 그룹 요소부터 재생을 시작한다. 이 때 begin 속성이 "UNRESOLVED"라면 시작 시간을 알 수 없는 것이기 때문에 재생을 할 수 없게 된다. 다른 요소의 경우도 마찬가지로 시작 시간이 해결되지 못한 요소의 경우는 재생될 수 없다. 재생될 수 있는 요소의 차례가 오면 가지고 있는 속성을 이용하여 시간 간격(interval)을 생성해 낸다. 시간 간격이란 미디어 요소의 시작 시간과 종료 시간을 의미하며 시간 간격 표는 현 시점까지 생성된 시간 간격들을 시작 시간 순서대로 저장해 놓은 것이다. 표 1은 예 4 (b)의 문서가 재생되면서 생성된 시간 간격을 나타낸 것이다.

재생이 진행되면서 동시에 생성되는 시간 간격을 문서 타이머(document timer)가 참고하면서 재생에 이용하게 된다. 이벤트 의존 요소의 시간을 해결하기 위하여 시간 간격을 생성하자마자 바로 Event 문서를 검사한다. Event 문서에서 from, fromKind 속성에 현재 생성한 시간 간격에 해당하는 요소가 나타나 있으면 바로 to, toKind 속성에 나타난 미디어 요소에 시간을 전달한다. 시간을 전달받은 요소는 자신의

표 1. 재생 시작과 동시에 생성된 시간 간격(interval) 표

요소 명(아이디)	시작 시간(초)	종료 시간(초)
par (group)	0	INDEFINITE
img (smile)	0	5
img (frown)	0	3
img (smile)	5	10

표 2. 재생이 진행되면서 생성된 시간 간격(interval) 표

요소 명(아이디)	시작 시간(초)	종료 시간(초)
par (group)	0	INDEFINITE
img (smile)	0	5
img (frown)	0	3
img (smile)	5	10
video (vid)	2	7

시간 간격을 수정하게 되고 이것은 다시 재생에 영향을 준다. 따라서 "vid" 비디오 요소의 시작 시간이 해결되면 표 1의 시간 간격 표에 "vid"에 관한 시간 간격이 추가되어 다음과 같이 시간 간격 표가 수정되는 것을 볼 수 있다.

4. 실험 결과

본 논문에서 제안한 이벤트 처리기가 올바르게 동작하고 있는지를 알아보는 실험은 고정된 시간을 이용해 미디어 요소들을 동기화 시킨 경우와 이벤트 의존 시간을 이용해 동기화 시킨 경우로 나누어서 진행하였다. 고정된 시간을 이용한 동기화란 재생되는 미디어 요소들의 시간 관련 속성 값이 이벤트에 의존하지 않는 속성 값들로만 구성된 문서를 의미한다. 반대로 이벤트 의존 시간을 이용한 동기화란 재생되는 미디어 요소 중에 이벤트 발생에 의존하여 재생 여부와 시간이 결정되는 요소를 포함하고 있는 문서를 말한다.

4.1 실험 시스템의 구성 및 개발 환경

본 논문에서 제안한 이벤트 처리기를 이용하여 SMIL에서 지정한 여러 호스트 순응 언어(Host Conformance Language) 중에서 Basic Profile을 지원하고, 추가적으로 시간과 동기화(timing and synchronization)에 관련된 속성을 처리할 수 있는 SMIL 재생기를 구현하였다. 본 시스템의 개발환경은 Windows 2000 Professional에서 Visual C++ 6.0과 DirectShow 8.0을 개발 도구로 사용하였고, 파서로는 MSXML 3.0을 이용하였다. 구현한 SMIL 재생기 시스템의 전체 구성도는 그림 1과 같다. SMIL 2.0에서 정의하고 있는 각 기능별 그룹에 따라 그 기능을 담당하는 매니저(Manager)를 두어 관리하도록 하였다.

각 매니저들이 하는 역할은 다음과 같다.

- ① Parse Manager : 불러들인 SMIL 문서가 유효한지 검사하고, 새롭게 추가되는 요소와 속성들에 대해 검사
- ② Layout Manager : 문서에 정의된 영역(region)들을 생성하고, 화면 영역이 올바르게 사용될 수 있도록 관리

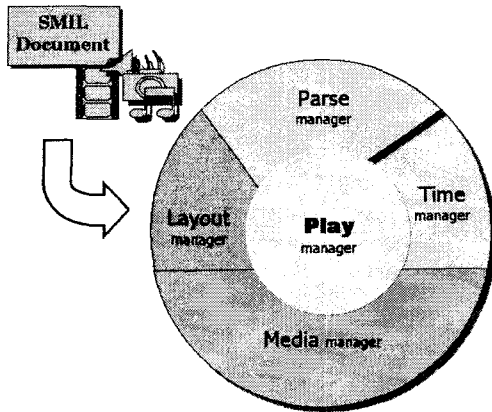


그림 1. 시스템 구성도

- ③ Time Manager : 각 미디어 요소들이 기본적으로 사용하는 시간과 재생 중 변화하는 시간을 관리
- ④ Play Manager : 타이머를 관리하며, 주어진 시간 간격을 이용하여 실제 재생을 진행
- ⑤ Media Manager : 개별적인 미디어 요소를 관리

본 시스템을 이용하여 SMIL 문서가 재생되는 과정을 간단히 살펴보면 다음과 같다. 재생하고자 하는 SMIL 문서를 선택하면, 그 문서는 Parse Manager에 의해 읽혀지고, 문서의 오류 여부[18, 19]를 검사하게 된다. 올바른 문서라고 판단되면 재생이 시작되기 전에 세 가지 작업이 이루어진다. 첫째, 재생에 이용할 각 매니저들을 생성한다. 둘째, Layout Manager를 통해 문서에 정의된 region 요소들을 생성하고 사용하게 될 공간 정보를 생성한다. 마지막으로 Time Manager를 통해 각 미디어, 시간 그룹 요소에서 사용하게 될 시간을 계산한다[1]. 이 과정들은 재생이 이루어지기 전에 수행된다. 사용자가 재생 버튼을 누르면 Play Manager를 통해 문서가 재생된다. Play Manager는 문서 전체에서 사용하는 문서 타이머를 가동시키고 각 시간별로 알맞은 요소를 재생하며 중요한 작업을 수행하게 된다.

사용자 인터페이스는 화면은 메뉴와 프리젠테이션 영역으로 구분되고, 메뉴의 선택은 사용자의 편의를 위하여 팝업과 툴바를 이용한 두 가지 선택 방법을 제공한다.

4.2 고정된 시간을 이용한 동기화

실험에 사용한 SMIL 문서는 예 6과 같다.

```

<smil xmlns="http://www.w3.org/2001/SMIL20/Language">
<head>
<layout id="a">
<root-layout id="SMIL_Example" width="640" height="480" />
<region id="left" top="0" left="0" height="240" width="320" />
<region id="right" top="0" left="320" height="240" width="320" />
<region id="bottom" top="240" left="160" height="240" width="320" />
</layout>
</head>
<body>
<seq id="first" begin="2s" end="20s">



</seq>
</body>
</smil>
    
```

예 6. 고정된 시간을 이용한 동기화 실험에 사용한 SMIL 문서

문서가 실행될 화면에 대한 정의는 <layout> 요소 내에 나타나 있는 것과 같이, 전체 화면 크기는 640×480 이고, 각 영역은 <region> 요소에 따라 배치될 것이다.

문서의 실행 순서는 “first” 요소에 의해 자식 요소들인 세 개의 이미지 객체들이 순차적으로 화면에 보인다. “first” 요소의 속성으로 begin과 end 시간이 명시되어 있으므로, 이 값에 의해 자식 요소들의 재생 시간이 제약을 받아서 18초 내에 세 개의 자식 요소들이 모두 수행되어야 한다. “copy” 요소의 경우에는 불연속적인 객체이기 때문에 “dur”이 명시되지 않는다면 재생될 수 없다. 그러나 부모 요소에 영향을 받아서 두 개의 객체가 수행되고 남은 시간인 14초에서 20초까지 6초 동안 화면에 보이게 된다. 이 문서의 경우, 이벤트 기반 시간 요소를 포함하고 있지 않으므로 Prefetch Schedule 단계가 수행되고 난 뒤 이벤트 문서는 생성되지 않고 수정된 SMIL 문서만이 생성되며 이 수정된 문서를 이용하여 재생이 진행되게 된다.

본 시스템을 이용하여 재생한 결과 재생시간에 따른 화면의 변화를 그림 2에 보였는데, 이는 올바른 재생 결과 화면이 보인다는 것을 알 수 있다.

4.3 이벤트 의존 시간을 이용한 동기화

이벤트에 의존하는 시간 속성을 갖는 요소를 이용하여 진행한 실험에 사용한 SMIL 문서는 예 7과 같다. 이 문서는 이벤트 의존 속성 값이 존재하기 때문

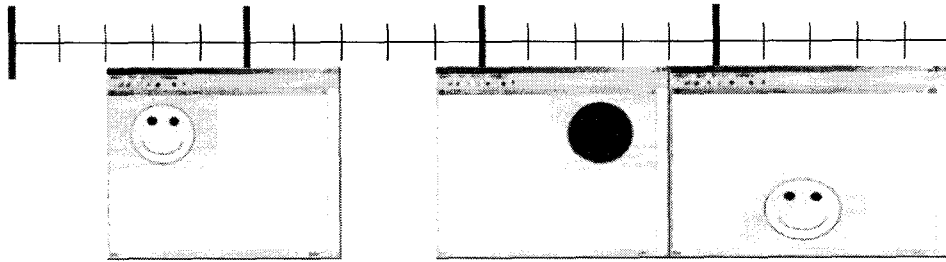


그림 2. 고정된 시간에 의한 동기화 실험의 실행 결과

```
<smil xmlns="http://www.w3.org/2001/SMIL20/Language">
<head>
<layout id="a">
<root-layout id="SMIL_Example" width="640" height="480" />
<region id="left" top="0" left="0" height="240" width="320" />
<region id="right" top="0" left="320" height="240" width="320" />
<region id="bottom" top="240" left="0" height="240" width="320" />
</layout>
</head>
<body>
<par dur="20s">
<img id="foo" end="10s" />
<video id="bar" begin="foo.beginEvent" />
<img id="copy" begin="foo.activateEvent" />
</par>
</body>
</smil>
```

예 7. 이벤트 의존 시간을 이용한 동기화 실험에 사용한 SMIL 문서

예 Prefetch Schedule 단계가 수행된 후에 SMIL 문서가 재구성될 뿐만 아니라 이벤트 문서도 생성된다.

실행되는 과정을 살펴보면 “foo” 요소는 0초부터 10초까지 재생이 진행된다. 이 때 “foo” 요소가 시작되는 이벤트가 발생하면 “copy” 요소의 재생이 시작된다. 또한, “foo” 요소가 재생되고 있는 화면에 마우스 클릭이 발생했을 경우에는 activateEvent가 발생하여 “copy” 요소가 실행된다. 이벤트 처리기를 이용하여 재생을 한 결과 예 8과 같은 Event 문서가 중간 단계에 생성된다.

그림 3은 본 시스템을 이용하여 재생하는 과정에서 사용자가 “foo” 요소인 화면의 왼쪽 위에 있는

```
<event>
<item from="foo" to="bar" fromKind="beginEvent" toKind="begin" />
<item from="foo" to="copy" fromKind="activateEvent" toKind="begin" />
</event>
```

예 8. 이벤트 의존 시간을 이용한 동기화 실험에서 생성된 이벤트 문서

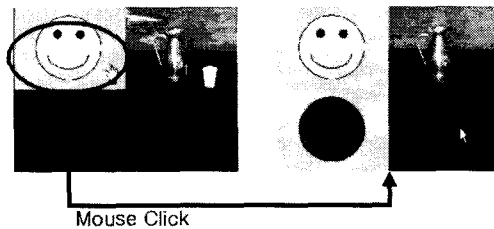


그림 3. 이벤트 의존 시간을 이용한 동기화 실험의 실행 화면.

그림을 마우스로 클릭한 상황을 보여주고 있다. 그림을 마우스로 클릭할 경우 해당하는 그림에 “activateEvent”가 발생하고 이 그림과 이벤트로 연결되어 있던 “copy” 요소가 화면의 왼쪽 아래 부분에 나타나게 된다. 따라서 이벤트 의존 시간이 포함된 문서의 재생에서도 이벤트 처리기가 올바르게 작동하여 재생된다는 것을 알 수 있다.

4.4 성능 비교

현재 발표되어 있는 대표적인 재생 도구인 RealNetworks사의 RealOne player와 Oratrix사의 GRiNS player를 이용하여 SMIL의 시간 모델을 제대로 만족하는지 몇 가지 항목으로 나누어 비교해 보았다.

첫 번째로 시간 그룹 요소들이 혼합되어 구성되어 있는 예 9와 같은 문서를 실행하여 보았다. 이 문서의 실행 과정과 결과를 예상해 보면 <seq> 요소의 두 자식 요소 중 먼저 <par> 요소의 실행이 끝나고 이어서 “image2” 요소가 실행되어 3초 뒤에 끝날 것이다. 이 문서를 이용하여 실험을 수행해 본 결과, RealOne player는 예상한 것과 같은 결과를 보여주었지만 GRiNS player는 그러한 결과를 보여주지 못했다. GRiNS player에서는 <par> 요소의 수행이 끝난 뒤 바로 전체 문서의 재생이 종료되어 “image2”의 출력을 볼 수 없었다.

```
<body>
  <seq id="cont1">
    <par id="cont2">
      <video id="video1" region="left" src="videos/GLOBE.AVI" />
      
    </par>
    
  </seq>
</body>
```

예 9. 시간 그룹 요소들이 혼합되어 구성된 SMIL 문서

기본적으로 요소는 불연속 객체이므로 dur 속성을 명시하지 않으면 "묵시적인" 재생 시간은 0초로 인식하게 된다. 그러나 테스트 문서의 경우, dur 속성은 나타나지 않았지만 end 속성을 이용하여 "image2"의 재생 시간을 3초로 나타내고 있기 때문에 3초 동안 재생이 진행되어야 한다. GRiNS player의 경우, 시간 그룹을 혼합하여 구성하지 않고 단독으로 구성하여 요소를 테스트하였을 경우에는 end 속성에 의한 재생 시간 제어를 제대로 인식한 것으로 보아서 혼합된 시간 그룹에서 불연속 객체의 재생 시간처리가 올바르게 되었음을 알 수 있다.

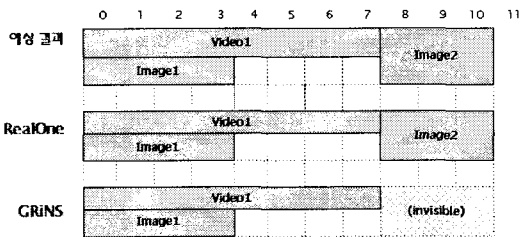


그림 4. 시간 그룹 요소들이 혼합되어 구성된 SMIL 문서의 실행 결과

두 번째로 테스트해 본 것은 예 10의 문서에서 보는 바와 같이 이벤트 의존 시간을 포함하고 있는 경우이다. 올바른 실행 과정은 0초에 재생이 시작되면서 바로 "image1"의 시작 동기화에 기반을 둔 비디오 요소인 "video1"이 실행되어야 한다. 하지만 이 시점에 "video1"의 부모 요소인 <par> 요소가 아직 실행되지 않았기 때문에 "video1"은 화면에 나타날 수 없다. 3초 후에 "par00"의 실행이 시작되면 비로소 "video1"도 화면에 보일 수 있게 되며 비디오 내부에서 3초가 지난 시점부터 시작되어야 한다.

이렇듯 다른 요소의 실행 시간에 동기화되어 의존하고 있는 시간 속성은 이벤트에 의존하고 있는 시간

속성과 다르게 처리되어야 한다. 부모 요소가 활성화 되었을 경우만 다른 요소의 재생 시간에 영향을 받는 것이 이벤트 의존 시간이라면 부모 요소의 활성화 여부에 관계없이 의존하고 있는 요소의 재생 시간과 동기화되는 것은 동기화 의존 시간이다.

테스트를 수행해 본 결과, 이번에는 GRiNS player에서 올바른 결과를 보여 주었고 RealOne player에서는 다른 결과를 보여주었다. RealOne player의 경우, 비디오 요소인 "video1"은 아예 재생되지 않았다. 그 이유는 "image1"이 시작되는 시점에 부모 요소인 "par00"이 실행되지 않고 있기 때문에 그 자식 요소인 "video1"이 동기화된 시간을 인식하지 못하도록 처리하고 있는 것으로 생각된다.

따라서 RealOne player의 경우는 올바르게 동기화 의존 시간을 처리하지 못하였으므로 이벤트 의존 시간과 동기화 의존 시간을 구별하여 수행하는 처리가 올바르게 되었음을 알 수 있다.

```
<body>
  <seq id="seq00">
    
    <par id="par00">
      
      <video id="video1" src="nist.mpg" region="bonus" begin="image1.begin" dur="5s" />
    </par>
    
  </seq>
</body>
```

예 10. 이벤트 의존 시간을 포함하고 있는 SMIL 문서

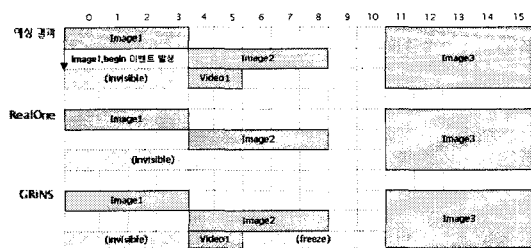


그림 5. 이벤트를 포함하고 있는 SMIL 문서의 실행 결과

표 3은 SMIL 문서 작성시에 나타날 수 있는 대표적인 타이밍 및 동기화 설정 형태를 토대로 그룹화하여 설정하고 해당 항목의 기능을 포함하는 SMIL 문서를 작성하여 본 논문에서 구현한 시스템과 RealOne player, GRiNS player에서 각각 실행,

표 3. 재생기별 성능 비교

항목	재생기	<i>RealOne</i>	<i>GRiNS</i>	제안한 시스템
단순 시간 그룹 내 미디어 객체 처리		○	○	○
혼합된 시간 그룹 내 불연속적인 미디어 객체 처리		○	×	○
혼합된 시간 그룹 내 연속적인 미디어 객체 처리		○	×	○
비활성화 된 부모 요소 내에서 동기화 의존 시간 처리		×	○	○
비활성화 된 부모 요소 내에서 이벤트 의존 시간 처리		○	×	○

비교한 결과를 보여준다.

SMIL의 시간 모델은 상당히 복잡하고 여러 가지 애매한 상황에 대해서 재생기가 임의로 처리할 수 있도록 허용하고 있어서 재생기들 사이에서도 다른 결과를 보이는 문제점을 내포하고 있다.

5. 결 론

SMIL의 장점은 개별적으로 존재하는 수많은 멀티미디어 데이터들을 통합하여 새로운 멀티미디어 데이터를 만들 수 있다는 것과 기존 웹 브라우저에서는 불가능했던 멀티미디어 데이터들의 동기화를 수행할 수 있다는 것이다. 이러한 SMIL은 웹뿐만이 아니라 방송, 광고, 교육 등 여러 분야에서 활용될 수 있고 다른 마크업 언어들과 통합되어서 사용될 수 있을 것으로 기대된다.

그러나 여러 가지 장점에도 불구하고 현재는 SMIL이 대중화되지 못하고 있는 것이 사실이다. 그 원인으로서는 아직 사람들에게 SMIL의 개념이 명확히 인식되지 못한 점과 SMIL이 여러 기능 그룹으로 분류되어 지원되는 재생기마다 지원할 수 있는 범위가 달라 활용하기가 어려운 점 등을 들 수 있다[18]. 따라서 SMIL이 활성화될 수 있도록 하기 위해서는 저작 도구의 개발과 더불어 재생 도구의 개발이 필수적으로 요구된다.

본 논문에서는 SMIL 2.0 시간 모델을 만족하는 이벤트 처리기를 설계 및 구현하였다. 그리고 이벤트 처리기를 이용하여 SMIL Basic 프로파일을 만족하는 재생기를 개발하였다. 이것은 SMIL 콘텐츠들을 좀 더 정확하게 재생할 수 있도록 하여 SMIL의 활성화에 기여할 수 있을 것으로 예상된다. 또한 SMIL 표준안에 정의된 여러 종류의 프로파일에서 재사용될 수 있고, XHTML+SMIL이나 SMIL Animation과 같은 SMIL을 통합하는 다른 표준들에 응용될 수 있을 것으로 예상된다.

향후 연구 과제로는 이벤트 처리기를 이용하여 애니메이션 관련 그룹과 화면 변환 효과 관련 그룹 등에 대한 지원이 가능하게 하여 SMIL에 제시된 모든 모듈을 포함하는 프로파일인 SMIL Language 프로파일을 만족하는 재생기를 구현하고자 한다. 그리고 PDA와 같은 제한된 성능을 가진 휴대용 장치에서 SMIL 문서를 효과적으로 재생할 수 있도록 하는 부분에 대해 연구할 계획이다.

참 고 문 헌

- [1] J. Ayers, et al., "Synchronized Multimedia Integration Language(SMIL) 2.0," World Wide Web Consortium Recommendation, <http://www.w3.org/AudioVideo>, 2001.
- [2] Lloyd Rutledge, "Multimedia Standards: Building Blocks of the Web," *IEEE MultiMedia*, Vol. 8, No. 3, pp. 13-15, 2001.
- [3] F. Arciniegas A., "A Realist's SMIL Manifesto", <http://www.xml.com>, 2002.
- [4] D.C.A. Bulterman, "SMIL 2.0 Part 1: Overview, Concepts, and Structure," *IEEE MultiMedia*, Vol. 8, No. 4, pp. 82-88, 2001.
- [5] F. Arciniegas A., "A Realist's SMIL Manifesto, Part II," <http://www.xml.com/pub/a/2002/07/17/smil.html>, 2002.
- [6] D.C.A. Bulterman, "SMIL 2.0: Repurposing Broadcast Content for the Web", *XML Technologies in Broadcasting*, 2001.
- [7] <http://www.mobilemms.com>.
- [8] <http://www.oratrix.com/Products/G2P/index.html>.
- [9] Jourdan, M., "A Formal Semantics of SMIL: A Web Standard to Describe Multimedia Documents," *Computer Standards & Inter-*

faces, Vol. 23, No. 5, pp. 439-455, 2001.

[10] A. Zeigler, "Get Up to Speed with SMIL 2.0. An XML-based Approach to Integrating Mlti-media into Web Content," *IBM developerWorks*, 2002.

[11] <http://www.realnetworks.com/solutions/leadership/realone.html>.

[12] D.C.A. Bulterman, Hardman, L., Jansen, J. Mullender, K.S. and Rutledge, L. "GRiNS: A GRaphical INterface for Creating and Playing SMIL Documents," *Proc. of Seventh International World Wide Web Conference, Melbourne*, Australia, 1998.

[13] D.C.A. Bulterman, "SMIL 2.0 Part 2: Examples and Comparisons," *IEEE MultiMedia*, Vol. 9, No. 1, pp. 74-84, 2002

[14] <http://www.macromedia.com>.

[15] *Inside QuickTime - Interactive Movies*, Apple Computer, Inc., 2001.

[16] M. Kim et al., "eXtensible MPEG-4 Textual Format," Contribution to ISO-IEC JTC1/SC29/WG11 MPEG00/6110, Int'l Organization for Standardization, Geneva, 2000.

[17] 김혜은, 채진석, 우요섭, "SMIL 2.0 문서 재생을 위한 실시간 이벤트 스케줄러의 설계 및 구현," 2002 춘계학술발표논문집(B), 한국정보과학회, Vol. 29. No. 1, 한양대학교, 2002.

[18] Tae-Hyun Kim, Kyung-II Kim, Kyu-Chul Lee, "Simple and Consistent SMIL Authoring: No More Structure Editing and No More Errors," *Multimedia Computing on the World Wide Web 2000*, IEEE, 2000.

[19] P.N.M. Sampaio, C.A.S. Santos, J.P. Courtiat, "About the Semantic Verification of SMIL

Documents," *Proc. of IEEE International Conference on Multimedia and Expo (ICME 2000)*, New York, USA, 2000.



김혜은

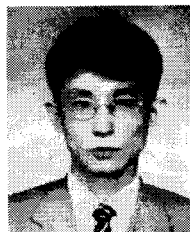
2001년 인천대학교 전자계산학과(공학사)
 2003년 인천대학교대학원 컴퓨터공학과(공학석사)
 2003년~현재 (주)에니스틸닷컴 연구원

관심분야 : XML, SMIL, 데이터베이스



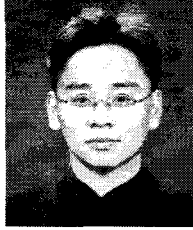
채진석

1990년 서울대학교 컴퓨터공학과(공학사)
 1992년 서울대학교 대학원 컴퓨터공학과(공학석사)
 1998년 서울대학교 대학원 컴퓨터공학과(공학석사)
 1992년~1997년 서울대학교 공학연구소 조교
 1997년~1998년 한국학술진흥재단 부설 첨단학술정보센터 선임연구원
 1998년~현재 인천대학교 컴퓨터공학과 조교수
 관심분야 : 인터넷 소프트웨어, 마크업 언어, 디지털 도서관



이재원

1990년 서울대학교 컴퓨터공학과(공학사)
 1992년 서울대학교 컴퓨터공학과(공학석사)
 1998년 서울대학교 컴퓨터공학과(공학박사)
 1999년~2002년 성신여자대학교 컴퓨터정보학부 강의전담교수
 2002년~현재 성신여자대학교 컴퓨터정보학부 전임강사
 관심분야 : 기계학습, 전산금융, 자연언어처리



김 성 동

1991년 서울대학교 컴퓨터공학과 (학사)

1993년 서울대학교 컴퓨터공학과 (석사)

1999년 서울대학교 컴퓨터공학과 (박사)

1999년~2001년 서울대학교 컴퓨터신기술공동연구소 특별연구원

2001년~현재 한성대학교 컴퓨터공학부 조교수

관심분야: 자연언어처리, 기계번역, 데이터마이닝, 기계학습, 전산 금융



이 종 우

1990년 서울대학교 컴퓨터공학과 졸업(학사)

1992년 서울대학교 컴퓨터공학과 졸업(석사)

1996년 서울대학교 컴퓨터공학과 졸업(박사)

1996년~1998년 현대전자산업 과장

1998년~1999년 현대정보기술 책임연구원

1999년~2002년 한림대학교 정보통신공학부 조교수

2002년~2003년 광운대학교 컴퓨터공학과 조교수

2003년~현재 (주)아이닉스 소프트 개발이사

관심분야: 전산금융, 군집 컴퓨팅, 분산 병렬 시스템, 시스템 소프트웨어