

애니메이션 컴포넌트의 시간적 스크립트로부터 SMIL 문서 생성 알고리즘

하 양†

요 약

멀티미디어 내용을 웹에 전달해 주기 위해 W3C에서 제안한 SMIL은 현재 널리 사용되고 있으며, 이와 관련된 소프트웨어가 속속 등장하고 있다. 따라서, 본 연구는 시간의 흐름에 따른 상호작용 프로그래밍을 하기 위한 컴퓨터 애니메이션 툴 키트인 RASP의 시간 스크립트를 SMIL 문서 형태로 변환해 주는 알고리즘을 제안한다. 이것은 SMIL 문서의 활용가치가 높아짐에 따라 애니메이션 컴포넌트에 대해 재사용성을 향상시키며, UML의 순서 다이어그램으로부터 SMIL 문서를 생성해 내므로 기존에 제안된 SMIL 문서의 동기화를 위해 생성된 순서 다이어그램을 검증해 주는데, 큰 의의를 갖는다.

An Algorithm to generate SMIL Documents from Temporal Scripts of Animation Component

Yan Ha†

ABSTRACT

The SMIL specification that is recently approved by the W3C and is meant to help deliver multimedia content to the Web, is widely used in these days and they have continuously emerged tools or softwares related with that. In this paper, I propose an algorithm to convert temporal scripts of RASP that is an experimental toolkit for computer animation that promotes interaction-based programming over time into a SMIL document. For making better use of SMIL documents, we can improve reusability of animation components. And, main contribution of this paper is that it verifies the sequence diagram generated for synchronization of SMIL documents by reconverting SMIL from UML sequence diagram.

키워드 : SMIL, 애니메이션(animation), 컴포넌트(component), RASP, VML, 시간 스크립트(temporal scripts)

1. 서 론

XML(eXtensible Markup Language)은 특정 응용 목적에 맞게 엘리먼트, 애트리뷰트, 엔티티를 정의할 수 있는 융통성(flexibility)을 갖고 있으며, 이들 응용들은 사용목적에 따라 웹과 인터넷 응용, 메타데이터와 저장 응용, 멀티미디어 응용 등으로 구별된다. 특히 멀티미디어 XML 기반 응용은 그래픽, 비디오, 디지털화된 언어와 같은 정보를 웹에서 제시하기 위한 XML 언어와 문법을 사용한다[1]. 이에 대한 예로써 SMIL(Synchronized Multimedia Integration Language), PGML(Precision Graphics Markup Language), JSML(Java Speech Markup Language)이 있는데, 이 중에서 SMIL은 W3C가 추천하는 웹에서 사용될 TV와 같은 멀티미디어 표현(Presentation)을 작성해 주며[2], 일련의 개별적

멀티미디어 객체를 동기화된 멀티미디어 표현(presentation)으로 통합할 수 있도록 해주는 XML 응용이다. SMIL이 갖는 기능을 나열해 보면 다음과 같다. 첫째, 표현의 시간적 행동을 기술한다. 둘째, 화면상에서 표현의 배치를 기술한다. 셋째, 하이퍼링크를 미디어 객체와 결합한다[3].

현재 SMIL 문서가 늘어나고 있으며, 이와 관련된 소프트웨어나 툴이 활발하게 개발되고 있는 실정이다. 따라서, 앞으로 SMIL 문서의 활용 가능성은 더욱 높아지리라 예상된다.

한편, 컴퓨터 애니메이션의 소프트웨어 재사용 측면에서 개발되어 사용되고 있는 대표적인 것으로 RASP(Robotics Animation Simulation Platform) 툴 키트를 꼽을 수 있다. 이것은 시각적인 시뮬레이션을 개발하고자 할 때 연구자들을 편리하게 해주기 위해 공통적인 구조나 집합을 정의해 놓은 것으로 객체 지향 원리와 다양한 시뮬레이션 기법에 기반으로 두고 있으며, C++로 작성되어 있다[4].

† 준 회 원 : 경인여자대학 컴퓨터정보기술학부 교수
논문접수 : 2001년 10월 23일, 심사완료 : 2003년 1월 10일

그래서, 본 논문은 RASP 툴 키트의 애니메이션 컴포넌트들의 시간 상 관계를 나타내는 스크립트를 UML 순서 다이어그램으로 시각화하며, 이를 이용하여 SMIL 문서를 생성하고자 한다.

본 연구는 애니메이션 컴포넌트를 SMIL 문서로 재생성하여 이들 애니메이션 컴포넌트의 재사용성을 극대화시키고자 한다. 뿐만 아니라 기존에 연구한 바 있는 SMIL 문서의 동기화를 위한 알고리즘[5]에 대한 검증으로 순서 다이어그램으로부터 SMIL 문서를 역생성하는데 그 목적이 있다.

논문 구성은 다음과 같다. 2장은 관련 연구를 설명하고, 3장에서는 애니메이션 컴포넌트에서 사용되는 시간 스크립트, UML의 다이어그램들, 그리고, SMIL 문서에 대해 설명하며, 4장에서는 시간 스크립트로부터 SMIL 문서를 생성하는 시스템과 알고리즘, 이를 적용한 결과를 제시한다. 끝으로 5장에서는 결론 및 향후 연구 과제를 제시한다.

2. 관련 연구

소프트웨어 재사용은 컴퓨터그래픽 분야에서 관심이 집중되고 있으며, 많은 그래픽 응용에서 애니메이션 장면을 시각화하는데 있어서 코드와 설계를 재사용하여 시간과 노력, 비용을 절감시키고자 한다. 그러나, 시간에 따라 변하는 컴포넌트들간의 상호작용을 애니메이션하는 것이 쉽지가 않아, 대부분의 도구들이 제한적인 관계만을 지원하고 있다.

그런데, 최근에 이를 해결하기 위해 RASP 툴 키트이라는 새로운 방법이 제안되었다. 이것은 계층적으로 상호작용을 구성하고 제어하기 위한 도구로써, 각 단계에서 시간에 따른 상호작용을 구성하고 이를 적절하게 재사용하도록 해준다. 여기서, 시간에 따른(time-varying) 상호작용이란 시간적 속성(attribute)에 의해 재사용할 수 있는 상호작용을 증가시키는 것을 말한다. 즉, 속성은 언제, 얼마나 자주, 어떤 조건아래 상호작용이 일어나는지를 규정하며, 상호작용은 이 속성이 사용된 컴포넌트들간의 관계를 설정해주는 것이다[6].

한편, SMIL 문서의 시간적인 동기화를 표현하기 위한 방법으로 UML 사용사례와 순서 다이어그램을 이용한 방법 [5]이 있다. 이것은 UML의 사용사례 뷰(View)의 사용사례, 순서, 협력 다이어그램 그리고, 논리적 뷰의 클래스 다이어그램을 이용하여 SMIL 문서의 객체들을 추출, 그들 간의 동기화 및 협력 관계를 나타내고 클래스 다이어그램을 통해 논리적 관계를 객체 모델링하는 연구이다. 따라서, 본 연구에서는 위의 연구에서 SMIL 문서로부터 생성한 UML 순서 다이어그램에 대한 검증으로 순서 다이어그램으로부터

터 SMIL 문서를 역생성하는 알고리즘을 제안한다.

3. RASP, UML, SMIL 문서

본 장에서는 RASP 툴 키트(tool kit), UML 다이어그램들, 그리고 SMIL 문서의 기본적인 태그에 대해 설명한다.

3.1 RASP 툴 키트

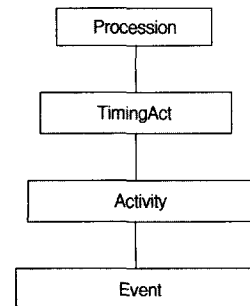
RASP는 상호작용기반 프로그래밍을 하는 컴퓨터 애니메이션을 위한 실험적인 툴 키트로, 기하학적인 모델을 구성, 렌더링, 그리고 애니메이션하기 위한 도구들로 구성되어 있으며, 재사용 할 수 있는 시간에 따른 상호작용을 지원함으로써 재사용을 촉진한다. 그리고, 시간에 따라 시스템적으로 컴포넌트를 상호 연결하기 위해 계층적 구조를 고용하는데, 이로 인하여 어떤 컴포넌트가 언제, 얼마나 자주 상호작용하는지 결정을 쉽게 하도록 한다.

3.1.1 프리미티브(primitive)

프리미티브 계층 구조는 그림과 같다. 이벤트(Events)는 하나의 상호작용을 관리하는 것으로 컴포넌트들 간의 통신하는데 있어서 시간적 바인딩(binding)을 형성한다. 활동(Activities)은 행동(behaviors)을 형성하는 상호작용을 관리하는데, 얼마나 자주, 어떤 순서로 상호작용이 일어나는지 제어하는 시간의 간격을 제한한다. 시간에 따른 활동(Timing-Acts)과 행렬(Processions)은 시간 스크립트를 형성하는 행동을 모은다. 여기서 시간 스크립트는 어떻게 행동이 관련이 있는지 언제 행위가 일어나는지를 결정한다.

3.1.2 재사용(Reuse)

소프트웨어의 재사용은 프리미티브의 계층에 따라 접근 형태가 다른데, fine-grain, medium-grain, coarse-grain이 있다. Fine-grain은 사건(events)에 관계된 것이고, medium-grain은 활동(activities), coarse-grain은 시간 스크립트(temporal scripts)에 관련된 재사용이다. 본 연구에서 사용되는 coarse-grain 재사용의 목적은 상호작용(interactions)이나 행동(behavior)이 아닌 순서의 재사용이다.



(그림 1) 프리미티브 계층

3.2 UML

UML의 여러 다이어그램에서 본 논문에서 적용하는 사용 사례 다이어그램과 순서 다이어그램에 대해 설명한다.

3.2.1 사용 사례 다이어그램

사용 사례 다이어그램은 시스템의 외부 행위자와 시스템이 제공하는 기능인 사용 사례(Use case)와의 관련성을 나타낸다. 하나의 사용 사례는 그 시스템이 제공하는 기능 중의 하나를 나타내며, 순서, 협력, 클래스 다이어그램 등을 삽입(insert)할 수 있다.

사용 사례 다이어그램의 표기는 외부 사용자와 사용 사례들의 집합, 그리고 그들간의 관계로 이루어진 그래프이며 객체를 사용자의 요구 사항으로부터 자연스럽게 유도하고 사용자와의 의사 소통을 정형화된 다이어그램 형식으로 나타내는 장점이 있다[7].

다음은 사용 사례 다이어그램의 구성요소와 이에 대한 설명이다.

<표 1> 사용사례 다이어그램의 구성 요소

구 성 요 소	설 명
액터(Actor)	시스템과 서브시스템, 혹은 클래스와 상호 작용을 하는 외부 사람, 프로세스 등을 말한다.
사용 사례(Use Case)	시스템의 내부구조를 나타내지 않는 하나의 연관성 있는 행동의 단위를 말한다.

3.2.2 순서 다이어그램

순서 다이어그램은 다른 객체들 사이의 메시지 흐름의 순서에 초점을 둔다. 표기는 2개의 축을 중심으로 수직축은 시간의 흐름을 나타내고 수평축은 해당 객체를 나타낸다. 객체는 직사각형으로 나타내고, 객체의 이름은 밑줄을 치고, 점선으로 된 수직선으로 객체의 생명선(lifeline)을 표시한다. 객체들 간의 상호작용은 메시지의 흐름을 통해 이루어지는데, 메시지는 객체의 생명선 사이의 수평선이 되며, 제어 초점(focus of control)은 길이가 긴 직사각형 모양으로 객체가 활동하는 시간대를 나타낸다.

다음은 순서 다이어그램에서 표현되는 요소들과 이에 대한 설명이다[8].

<표 2> 순서 다이어그램의 구성 요소

구 성 요 소	설 명	
사건 식별자 (event identifier)	메시지가 일어나는 사건을 참조할 때 필요한 식별자를 말한다.	
시간 제한 (time constraint)	사건들(events) 사이에 관련 있는 시간을 나타낼 때 시간 표시가 필요하다.	
메시지 (message)	Peer-to-peer	하나의 객체는 특별한 메시지를 오직 하나의 객체에 보낸다.
	Broadcast	하나의 객체에서 보내진 메시지는 하나 이상의 다른 객체에게 동시에 보낸다.

3.3 SMIL 문서

SMIL 문서는 기존의 HTML 등 하이퍼텍스트를 위한 마크업 언어에서 다루지 않았던 멀티미디어 데이터들 간의 시간적인 동기 관계를 명시할 수 있는 XML 기반 언어이다[9].

SMIL 문서는 <smil>, </smil> 태그 안에 <head> 부분과 <body> 태그 부분으로 이루어진다. 다음은 세부적인 태그들에 대해 간략하게 설명한다.

3.3.1 <head> 태그

순서에 따라 제시하는 것과 무관한 정보에 대한 태그이다. 여기에는 메타(meta) 엘리먼트와 <switch>나 <layout> 태그들이 포함된다[3].

(1) <layout> 태그

문서의 <body> 안에 엘리먼트가 어떤 형태로 놓이는지를 결정한다.

(2) <region> 태그

미디어 객체 엘리먼트의 위치, 크기 등을 제어한다,

(3) <switch> 태그

선택적인 엘리먼트의 집합을 규정하도록 하는데, 여러 개의 엘리먼트 중 하나만이 선택된다. 이 태그는 <body> 안에서도 사용 가능하다.

3.3.2 <body> 태그

문서의 시간상 연결 행동과 관련된 정보를 포함한 엘리먼트이다[9].

(1) 미디어 태그

미디어 객체를 나타내는 엘리먼트로서, <ref>, <animation>, <audio>, , <video>, <text>, <textstream>등이 이에 속한다. 이 중에서 본 논문에서는 <animation> 태그를 생성하도록 한다.

(2) 동기 태그

시간적인 동기화를 위해서 사용되는데, <par> 태그는 엘리먼트의 자식들이 같은 시간에 중복적으로 일어나도록 해주며, <seq> 태그는 엘리먼트의 자식들이 시간적인 순서를 갖도록 해준다.

(3) 하이퍼링크 태그

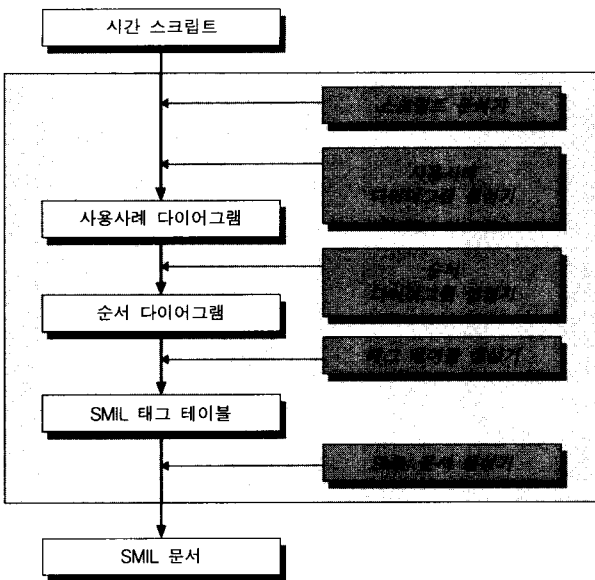
하이퍼링크와 관련된 태그는 <a>와 <anchor>가 있다. <a>는 HTML의 <a> 태그와 매우 유사한 것으로, 해당 미디어 리소스 전체에 대해 시작 시점부터 끝 시점까지 링크가 가능하도록 고정되나, <anchor> 태그는 <a>와 같이 하이퍼링크 엘리먼트이나 정의된 기간동안 화면 어느 곳, 특히 상연 중인 비디오 위에도 하이퍼링크 영역을 지정할 수 있다[12].

4. SMIL 문서 생성 시스템

본 장에서는 애니메이션 컴포넌트에 관한 시간 스크립트를 입력으로 했을 때, UML의 순서 다이어그램을 생성하고 이로부터 SMIL 문서를 생성해내는 시스템과 이를 위한 알고리즘, 적용결과를 제시한다.

4.1 시스템 구성도

먼저, 본 논문에서 제안하는 전체 시스템 구성은 그림 2와 같으며, 각 구성 요소에 대한 기능을 기술하면 다음과 같다.



(그림 2) SMIL 생성 시스템 구성도

(1) 스크립트 분석기

스크립트 분석기에서는 시간적 관계를 다음과 같은 메시지로 대응시켜 번역한다.

〈표 3〉 시간 스크립트의 관계에 대한 순서 다이어그램 메시지 대응표

관 계	메 세 지
MEETS	seq()
DELIMITS	par()
STARTS	par()
STOPS	$\{e_{end} - e_{end} = 0\}$ 단, e_i, e_j 는 사건 식별자
TimingAct(Object, begin, end)	$\{e_{end} - e_{begin}\}$
TimingAct(Object _i , begin, end) TimingAct(Object _j , begin, end)	$\{e_{begin} - e_{end}\}$

위의 대응표에 의해 다음 5가지 예제의 시간 스크립트를 분석한 결과를 각 테이블로 제시해본다.

메시지와 관련된 예문은 예제 1)과 예제 2)와 같다. 먼저, seq()와 관련된 예문이다.

예제 1) ObjectB → setRel(MEETS, ObjectC) ;

다음은 par()에 관한 예문이다.

〈표 4〉 예제 1)의 스크립트 분석 테이블

객 체		메 세 지
시 작	끝	
ObjectB	ObjectC	seq()

예제 2) ObjectC → setRel(DELIMITS, ObjectE) ;

〈표 5〉 예제 2)의 스크립트 분석 테이블

객 체		메 세 지
시 작	끝	
ObjectC	ObjectE	par()

다음은 제한조건에 관한 예제들이다. 2개의 객체들간의 끝나는 시간을 맞추기 위한 제한조건은 다음과 같다.

예제 3) ObjectF → setRel(STOPS, ObjectD) ;

〈표 6〉 예제 3)의 스크립트 분석 테이블

객 체	사건 식별자		제 한 조 건
	시 작	끝	
ObjectF	e_{1begin}	e_{1end}	$\{e_{end} - e_{end} = 0\}$
ObjectD	e_{2begin}	e_{2end}	

그리고, 하나의 객체의 생존 기간을 제한조건으로 표시하면 다음과 같다.

예제 4) proc → addTimingAct(ObjectA, 1, 10) ;

〈표 7〉 예제 4)의 스크립트 분석 테이블

객 체	사건 식별자		제 한 조 건
	시 작	끝	
ObjectA	e_{begin}	e_{end}	$\{e_{end} - e_{begin} = 9\}$

끝으로, 2개의 객체간의 시간차를 나타내는 경우는 다음과 같다.

예제 5) proc → addTimingAct(ObjectA, 1, 10) ;
proc → addTimingAct(ObjectB, 30, 40) ;

<표 8> 예제 5)의 스크립트 분석 테이블

객 체	사건 식별자		제 한 조 건
	시 작	끝	
ObjectA	e1begin	e1end	{e2begin - e1end = 20}
ObjectB	e2begin	e2end	

(2) 사용사례 다이어그램 생성기

다이어그램 생성 알고리즘에 의해 사용사례 다이어그램을 생성한다. 사용자와 SMIL 문서 생성을 위한 사용 사례를 생성한다.

(3) 순서 다이어그램 삽입기

스크립트 분석기에 의해 파싱된 메시지들을 이용하여 사용사례 다이어그램의 사용사례에 대한 순서 다이어그램을 삽입한다.

(3) 태그 테이블 생성기

순서 다이어그램으로부터 태그들을 추출하여 SMIL 태그 테이블을 생성한다. 태그 테이블에서는 동기, 미디어, 하이퍼링크 태그를 구별한다.

(4) SMIL 문서 생성기

SMIL 태그 테이블을 입력받아 SMIL 문서 인스턴스를 생성한다. 기본적으로 문서 인스턴스에서 생성되는 부분은 선언문, 주석문, <smil>과 <body>의 시작과 끝 태그이다.

4.2 알고리즘

시간 스크립트를 입력받아 순서 다이어그램, 태그 테이블을 통해 최종적으로 SMIL 문서를 생성하는 알고리즘은 다음과 같다.

```

입력 : 시간 스크립트
출력 : SMIL 문서

begin
{
  // 스크립트 분석 알고리즘
  while not (스크립트)
  {
    객체와 관계를 파싱한다.
    if (관계)
      메시지에 할당한다.
  }

  // 사용사례 다이어그램 생성 알고리즘
  액터를 생성한다.
  사용사례를 생성한다.
  액터와 사용사례 관계를 설정한다.
}
    
```

```

// SMIL 문서 생성을 위한 함수들 호출
순서 다이어그램 생성 함수를 호출한다.
태그 테이블 생성 함수를 호출한다.
SMIL 문서 생성 함수를 호출한다.

}
end
    
```

위의 주요 알고리즘에서 아래와 같은 서브 알고리즘을 각각 호출한다.

(1) 순서 다이어그램 생성

순서 다이어그램을 생성하는 함수의 알고리즘은 다음과 같다.

```

입력 : 메시지, UML 사용사례 다이어그램
출력 : UML 순서 다이어그램

// 순서 다이어그램 생성 함수의 알고리즘
begin
{
  순서 다이어그램을 삽입한다.
  if (순서 다이어그램)
  {
    객체를 생성한다.
    while not (메세지)
      if 하나의 객체에 대해 여러 메시지
        발생 par() 메시지 연결
      else seq() 메시지 연결
      사건 식별자 부여한다.
      제한조건을 부여한다.
    }
  }
end ;
    
```

(2) 태그테이블 생성

태그 테이블을 생성하는 함수의 알고리즘은 다음과 같다.

```

입력 : UML 순서 다이어그램
출력 : SMIL 태그 테이블

// 태그 테이블 생성 함수의 알고리즘
begin
{
  태그 테이블을 생성한다.
  while not (메세지)
  {
    if (메세지 == "애니메이션")
      애니메이션 태그를 인쇄한다.
    else if (메세지 == "하이퍼링크");
      하이퍼링크 태그를 인쇄한다.
  }
}
    
```

```

else
{
    동기태그를 인쇄한다.
    태그 추출 함수를 호출한다.
}
}
end
    
```

(3) SMIL 문서 생성

결과적으로 SMIL 태그 테이블로부터 SMIL 문서를 생성해 내는 알고리즘은 다음과 같다.

```

입력 : SMIL 태그 테이블
출력 : SMIL 문서

// SMIL 문서 생성 알고리즘
begin
{
    if (태그테이블)
    {
        선언문 삽입한다.
        while not (태그)
        {
            엘리먼트 시작태그 삽입한다.
            if (동기태그)
                태그테이블 생성 함수를 호출한다.
            else
            {
                애트리뷰트와 값을 삽입한다.
                엘리먼트 종료태그 삽입한다.
            }
        }
    }
}
end ;
    
```

4.3 적용 결과

본 시스템에서 시간적 스크립트를 입력으로 했을 때, 이에 대한 순서 다이어그램, 태그 테이블, 그리고, SMIL 문서는 다음과 같다.

4.3.1 시간 스크립트

- (1) ObjectB → setRel(MEETS, ObjectC) ;
- (2) ObjectC → setRel(MEETS, ObjectF) ;
- (3) ObjectC → setRel(DELIMITS, ObjectE) ;
- (4) ObjectC → setRel(MEETS, ObjectD) ;
- (5) ObjectF → setRel(STOPS, ObjectD) ;
- add timingActs to procession**
- (6) Procession * proc = new procession() ;

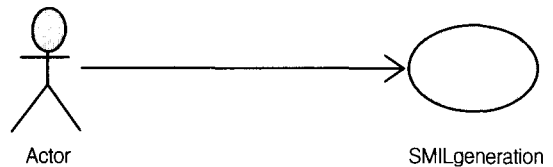
- (7) proc → addTimingAct(ObjectA, 1, 10) ;
- (8) proc → addTimingAct(ObjectB, 30, 40) ;

4.3.2 다이어그램

위의 시간 스크립트를 SMIL 문서 형태로 변환하기 위해 다이어그램 형태가 필요하다. 본 논문에서는 UML 사용 사례 다이어그램과 순서 다이어그램을 이용한다.

(1) 사용 사례 다이어그램

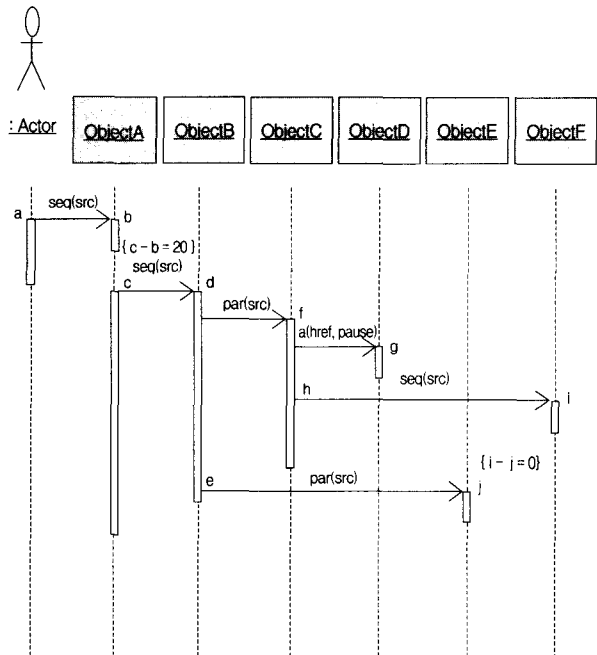
액터는 SMIL 문서를 생성하는 하나의 사용 사례와 상호 작용하며, 이에 대한 그림은 다음과 같다.



(그림 3) SMIL 문서 생성 사용 사례 다이어그램

(2) 순서 다이어그램

SMIL 문서 생성의 사용 사례에 순서 다이어그램을 생성하여 삽입(insert)한다. 순서 다이어그램은 객체와 객체들 간의 메시지 전달, 제한조건 등을 표시한다.



(그림 4) SMIL 문서 생성 순서 다이어그램

4.3.3 태그 테이블

순서 다이어그램으로부터 동기 태그와 하이퍼링크 태그, 그리고 미디어 객체를 구분하여 SMIL 태그를 위한 테이블을 생성한다.

<표 9> SMIL 태그 테이블

동기 태그	미디어 객체			
	동기태그	미디어 객체	하이퍼링크 태그	미디어 객체
seq	ObjectA			
	ObjectB			
	par	ObjectC	a	ObjectF
		ObjectE		
	ObjectF			

4.3.4 SMIL 문서

```

<?xml version = "1.0" encoding = "ISO-8859-1"?>
<!DOCTYPE smil PUBLIC "-// W3C//DTD
SMIL1.0//EN" "http:// www.w3.org/TR/REC-
smil/SMIL10.dtd">
<smil>
  <body>
    <seq>
      <animation src = "ObjectA", begin = 1, end = 10>
      <seq>
        <animation src = "ObjectB", begin = 30, end = 40>
        <par>
          <seq>
            <par>
              <a href = "D", show = "pause">
                <animation src = "ObjectC">
              </a>
              <animation src = "ObjectE">
            </par>
          <animation src = "ObjectF">
        </seq>
      </par>
    </seq>
  </body>
</smil>
    
```

5. 결론 및 향후 연구과제

웹에서 사용하는 멀티미디어 프레젠테이션을 작성해 주는 XML 응용으로 SMIL이 제안되어 현재 널리 사용되고 있으며, 이와 관련된 파서(parser), 에디터(editor), 플레이어(player) 등 소프트웨어 개발이 활발하게 이루어지고 있다. 이런 추세에 발 맞추어 컴퓨터 애니메이션에서 사용되는 컴포넌트들을 SMIL 문서로 관리할 필요성이 대두된다. 따라서, 본 연구는 시간의 흐름에 따른 상호작용 프로그래밍을 하기 위한 컴퓨터 애니메이션 툴 킷인 RASP의 시간 스크립트를 SMIL 문서 형태로 변환해 주는 알고리즘을 제안

하였다. 본 알고리즘은 UML의 순서 다이어그램으로부터 SMIL 문서를 생성해 내므로 기존에 제안된 SMIL 문서의 동기화를 위해 생성된 순서 다이어그램을 검증해 주는데, 큰 의의를 갖는다.

향후 연구 과제로는 컴퓨터 애니메이션을 위한 다양한 툴로부터 시간 개념을 포함한 컴포넌트들로부터 SMIL 문서를 생성하고 이를 통합해 주는 관리 시스템을 개발하는 것이다.

참 고 문 헌

- [1] Natanya Pitts-Moultis, Cheryl Kirk, "XML Black Book," The Coriolis Group Inc., 1999.
- [2] Elliotte Rusty Harold 저, 김용권 역, "XML Bible," 정보문화사, 1999.
- [3] <http://my.dreamwiz.com/gojirael/html/smil10spec/smil10spec.html>.
- [4] RASP-Robotics and Animation Simulation Platform, <http://www.cs.ubc.ca/spider/gslee/RASP/rasp.html>.
- [5] 채원석, 하 안, 김용성, "UML 사용사례 및 순서 다이어그램을 이용한 SMIL 문서 동기화", 정보과학회논문지(소프트웨어 및 응용), 제27권 제4호, 2000.
- [6] Gene S. Lee, "Reusable Interactions for Animation," The 5th International Conference on Software Reuse, June, 1998.
- [7] Bruce Power Douglass, "Real-Time UML Developing Efficient Objects for Embedded Systems," Addison-Wesley Longman Inc., 1998.
- [8] James Rumbaugh, Ivar Jacobson, Grady Booch, "The unified modeling language reference manual," Addison Wesley Longman Inc., 1999.
- [9] W3C, "Synchronized Multimedia Integration Language (SMIL) 1.0 Specification," W3C, June, 1998, <http://www.w3.org/TR/1998/REC-smil-19980615/>.
- [10] 김두현, 김지용, 황승구, "차세대 웹 상에서의 멀티미디어", 정보처리학회지, 제6권 제3호, pp.62-71, 1999.
- [11] Craig Larman, "Applying UML and PATTERNS : An Introduction to Object-Oriented Analysis and Design," Prentice-Hall, 1998.
- [12] M. J. Perez-Iuque, T. D. C Little, "A Temporal Preference Framework for Multimedia Synchronization," IEEE Journal on Select Areas in Communications, Vol.14, No.1m pp.36-51, 1996.
- [13] 류시원, 김창룡, 차광호, 정진호, "멀티미디어 데이터베이스 시스템에서의 멀티미디어 데이터 모델링에 대한 고찰", 1997, <http://www.multimedia.or.kr/newtech/NT14/nt2.htm>.
- [14] 채원석, 하 안, 김용성, "UML 클래스 다이어그램을 이용한

XML 문서 구조 다이어그램”, 정보처리논문지, 제6권 제10호, pp.2670-2679, 1999.

- [15] 하수철, 성해경, “게임공간의 분류와 시나리오의 시간 및 공간 동기화 표현법”, 정보처리학회논문지, 제6권 제10호, pp. 2630-2641, 1999.
- [16] 하 안, 황용주, 김용성, “SGML DTD로부터 UML 클래스 다이어그램으로의 사상 알고리즘”, 정보과학회논문지(B), 제26권 제4호, pp.508-520, 1999.
- [17] 홍도석, 하 안, 김용성, “UML 클래스 다이어그램을 XML DTD로의 변환 시스템 설계 및 구현”, 정보처리학회, 제7권 제12호, 2000.



하 안

e-mail : white@kic.ac.kr

1992년 덕성여자대학교 전산학과(학사)

1994년 이화여자대학교 전자계산교육전공
(석사)

2000년 전북대학교 대학원 전산통계학과
(이학박사)

2000년~2001년 중앙대학교 정보통신연구소 연구전담교수

2001년~현재 경인여자대학 컴퓨터정보기술학부 전임강사

관심분야 : XML 응용, 객체지향 모델링, 컴포넌트 모델링,
애니메이션, 멀티미디어