

## SMIL 문서 편집기 개발을 위한 객체 모델링

채 원 석\*

### Object Modeling for SMIL documents Editor Development

Won Seok Chae\*

#### 요 약

SMIL(Synchronized Multimedia Integration Language)는 멀티미디어 객체들을 시간적인 동기를 표현할 수 있는 마크업 언어이다. 본 연구에서는 SMIL 문서의 편집기 개발을 위해 시간에 따라 동기화 표현을 명확히 할 수 있는 도구인 UML의 사용 사례(Use case), 순서(Sequence), 다이어그램을 이용한 객체 모델링 방법을 제안한다. 이를 위해 SMIL 문서에서 UML 사용사례와 순서 다이어그램으로의 사상 규칙과 알고리즘, 형식 모델, 그리고 모델링 함수를 제안한다.

본 연구의 의의는 손쉽게 SMIL 문서 편집기 개발을 할 수 있고, 형식 모델과 다이어그램 모델링 함수는 객체지향 데이터베이스 문서 관리의 기반이 된다는 것이다.

#### Abstract

SMIL(Synchronized Multimedia Integration Language) allows multimedia objects into a synchronized multimedia presentation. In this paper, I propose modeling rules, formal models, modeling function and an algorithm for SMIL documents using use case diagram and sequence diagram of UML(Unified Modeling Language)to Develop for SMIL Documents Editor. It will be used the use case diagram and collaboration diagram for object-oriented visualizing tool to describe the temporal behavior of the presentation. The main contribution of this paper is that developers of SMIL documents editor easily generate them using this rules and algorithm. And, the formal models and modeling functions provide an environment for processing object-oriented documents.

▶ Keyword : SMIL, 객체 모델링, 사상규칙, 형식모델, 모델링함수.

• 제1저자 : 채원석

• 접수일 : 2005.06.11, 심사완료일 : 2005.07.15

\* 원광보건대학 정보컨텐츠과 교수,

※ 이 논문은 2003년 원광보건대학 학술연구비 지원(해외파견)에 의하여 연구되었음.

본 논문의 모델링 평가를 위해 기존의 멀티미디어 데이터 모델들의 일반적인 기준에 의거 비교, 분석하였다.

## I. 서론

1986년에 제정된[1] SGML(Standard Generalized Markup Language)의 기능은 너무 복잡하고 이를 지원하는 소프트웨어 개발이 용이하지 않을 뿐만 아니라 인터넷을 목적으로 하지 않아 인터넷 상의 서비스를 제대로 지원하지 못한다. 그리하여, SGML의 복잡한 기능을 축약시키고, HTML의 장점을 추가한 XML (eXtensible Markup Language)[2]은 특정 응용 목적에 맞게 엘리먼트, 애트리뷰트, 엔티티를 정의할 수 있는 융통성(flexibility)을 갖는데, 그 중에서 멀티미디어 XML 기반 응용으로 SMIL이 있다. SMIL은 인터넷에서 고품질의 내용을 전송할 뿐만 아니라 하이퍼링크 기능을 추가하여 빠르게 웹 기반 내용으로 이동하며, 표현을 동기화시킨다[4].

그러나 SMIL 문서를 개발하기 위한 보조적인 도구나 소프트웨어가 개발되지 않았기 때문에 복잡한 SMIL 문서를 개발하기가 무척 어려운 실정이다. 특히 문서 개발자들은 일일이 시간에 따른 미디어들 간의 관계를 파악해야 하며, 개발자들 간에 공통된 도식화를 사용하지 않아 문서 교환이 용이하지 않다. 따라서, 본 연구는 UML을 이용하여 SMIL 문서 편집기를 개발하는데 필요한 비주얼(visual) 모델링 방법을 제안한다.

본 논문은 UML을 이용하여 SMIL 문서를 객체지향적으로 동기화 모델링을 하기 위하여 SMIL 문서의 요소를 UML의 사용 사례, 순서, 협력, 클래스 다이어그램 요소에 사상시키는 사상 규칙들을 정의하고, 이 규칙들에 의해 모델링한 순서 다이어그램을 데이터베이스 스키마로 표현하기 위해 SMIL 문서의 동기 태그, 선택 태그, 하이퍼링크 태그에 대한 각각의 형식 모델을 정의하고, SMIL 문서의 사용 사례 다이어그램, 순서 다이어그램에 관련된 모델링 함수를 제안함으로써 문서의 구조를 비교, 변형시킬 수 있도록 하였다.

또한 사상 규칙들에 따라 SMIL 문서를 UML 사용 사례, 순서, 협력 다이어그램으로 생성하는 사상 알고리즘을 제안한다. 그리고, SMIL 문서의 순서 다이어그램으로부터 클래스를 추출하여 클래스 다이어그램을 생성하고 데이터베이스 스키마를 생성하고자 형식 모델을 제안하였다. 아울러

## II. 관련 연구

지금까지 SGML DTD를 모델링하는 방법으로는 구조도 (Structure Diagram)[5]와 XOMT(eXtended Object Modeling Technique) 다이어그램[5], UML 클래스 다이어그램(Class Diagram)[6]이 있다. 구조도는 가장 널리 사용되는 방법으로 트리 구조에 엘리먼트들간의 관계를 표현한다. XOMT와 UML 클래스 다이어그램은 정적인 구조를 나타내는 객체 모델링으로 DTD를 클래스 다이어그램으로 사상시킨다. 따라서, 이들을 이용하여 SMIL을 모델링하게 되면 시간적인 동기화와 하이퍼링크는 표현할 수가 없다.

XML에 관한 모델링으로 UML 클래스 다이어그램을 이용한 방법이 있지만[7] 이를 이용하여, XML 문법을 따르지만 특정 목적에 의해 태그를 정의하여 쓰는 SMIL을 모델링하게 되면 시맨틱상 적절한 모델링을 생성하지 못한다. 왜냐하면, XML에 대한 클래스 다이어그램에서는 SMIL에 포함된 시간의 개념을 표현한 모델링을 지원하지 않기 때문이다.

[10]에서 제안한 HyTime 문서 구조 다이어그램은 HyTime 애트리뷰트가 포함되며, 참조할 엘리먼트에 IDREF 애트리뷰트를 연결하는 점선은 존재하나, 어느 일정 기간 동안 링크를 지원하는 등 시간의 흐름에 따른 하이퍼링크 관계는 표현하지 못하는 단점이 있다.

또한 멀티미디어 데이터의 데이터베이스화는 먼저 다양한 멀티미디어 데이터의 시·공간적 특성에 부합되는 데이터 모델링과 사용자의 다양한 질의 형태에 대응하기 위한 검색에 대한 연구가 우선적으로 필요하다. 각 미디어들의 동기화 정보는 절대적인 시간 축 상에 표현된 사건들의 시점에 의해 표현되며 미디어들은 해당 시점에서 화면상에 적절히 표시된다. Firefly 모델은 사건을 그래프 상의 사각형 노드로 표현하고 매체의 상연과 사건 사이의 관계는 라벨(labeled)된 간선(edge)으로 표현하는 방법으로써 시간 관계를 표현한다. TPN 모델은 Allen이 정의한 13가지의 시간 관계에 의해 미디어간의 시간 정보를 직접 패트리네트를 이용하여 표현한 모델이며, 미디어간의 시간 정보를 직접

기술할 수 있기 때문에 시간 축 모델보다 상위 관계의 추상화 수단을 제공해 준다.

이와 같이 기존의 연구들은 SMIL의 가장 중요한 특징인 시간적인 동기화를 표현하는 모델링 방법은 제안된 바 없으며, 하이퍼링크의 경우에도 정해진 시간에 일어나는 등 다양한 형태를 구별하여 지원하지 못하고 있는 실정이다. 따라서, SMIL 문서의 시간적인 동기화는 물론 다양한 경우에 대한 하이퍼링크 관계를 표현할 수 있는 새로운 도구가 필요하다.

따라서, 본 연구는 UML의 사용사례 뷰(View)의 사용 사례, 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 모델링한다. 즉, SMIL 문서의 속성에 맞게 시간적 동기화 및 하이퍼링크에 대해 형식 모델을 제안하며, 객체의 발생 순서 등에 관한 모델링 함수를 제안하여 문서 구조 파악, 통합, 변형 등을 가능하게 한다.

### III. SMIL 문서와 UML

본 장에서는 SMIL 문서의 기본적인 태그와 UML에 대해 설명한다.

#### 3.1 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.

##### 3.1.1 <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 제한한다.

- ③ <switch> 태그 : 선택적인 엘리먼트의 집합을 규정하도록 하는데, 여러 개의 엘리먼트 중 하나만이 선택된다. 이 태그는 <body> 안에서 사용 가능하다.

#### 3.1.2 <body> 태그

문서의 시간상 연결 행동과 관련된 정보를 포함한 엘리먼트이다[10].

- ① 미디어 태그 : 미디어 객체를 나타내는 엘리먼트로서, <ref>, <animation>, <audio>, <img>, <video>, <text>, <textstream>등이 이에 속한다.
- 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 [8].

#### 3.2 UML

UML(Unified Modeling Language)는 시스템의 서로 다른 관점에서 보여주는 뷰들을 정의하는데, 각각의 뷰들은 시스템을 서로 다른 관점에서 보여주며 그것을 표현할 수 있는 다이어그램이 존재한다. 그 중에 대표적인 것으로는 사용사례 뷰와 논리적 뷰를 꼽을 수 있다[2].

##### 3.2.1 사용 사례 다이어그램

사용사례 다이어그램은 시스템의 외부 행위자와 시스템이 제공하는 기능인 사용사례(Use case)와의 관련성을 나타낸다. 사용사례 다이어그램의 표기는 외부 사용자와 사용사례들의 집합, 그리고 그들 간의 관계로 이루어진 그래프이다.

##### 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 이루어 지는데, 메시지는 객체의 생명선 사이의 수평선이 되며, 제어 초점(focus of control)은 길이가 긴 직사각형 모양으로 객체가 활동하는 시간대를 나타낸다.

다음은 순서 다이어그램에서 표현 가능한 항목들을 나열하고 설명한다[11].

- ① 사건 식별자(event identifier) : 메시지가 일어나는 사건을 참조할 때 필요한 식별자이다.
- ② 시간 제한(time constraint) : 사건들(events) 사이에 관련 있는 시간을 나타낼 때 시간 표시가 필요하다.
- ③ Peer-to-peer 메세지(message) : 하나의 객체는 특별한 메시지를 오직 하나의 객체에 보낸다.
- ④ Broadcast 메시지 : 하나의 객체에서 보내진 메시지는 하나 이상의 다른 객체에게 동시에 보낸다.

### 3.2.3 협력 다이어그램

협력 다이어그램은 순서 다이어그램과 같은 정보를 갖으나 메시지의 순서보다는 객체들 간의 정적 구조에 초점을 맞추고 있다. 따라서, 협력 다이어그램은 객체들, 그들 사이의 링크들, 그리고, 객체들 간의 주고받는 메시지를 순서대로 나열해 준다.

### 3.2.4 클래스 다이어그램

클래스 다이어그램은 클래스와 객체, 그리고 그들 간의 관계를 정적인 구조로 나타낸다. 클래스는 클래스 이름, 애트리뷰트 리스트, 오퍼레이션 리스트 부분으로 구성된다. 그리고, 클래스들 간의 관계 중 대표적인 것으로 일반화 관계가 있는데, 이것은 ‘△’로 표기한다.

## IV. SMIL의 객체 모델링

본 장은 SMIL 문서를 객체 지향적으로 동기화하기 위해 SMIL 문서로부터 UML로의 사상 규칙과 형식 모델, 사상 알고리즘, 그리고 사상 결과를 나타낸다.

### 4.1 사상 규칙

SMIL 문서로부터 UML 사용 사례 및 순서 다이어그램을 생성하기 위해서 다음과 같이 정의한다.

[정의 1] SMIL 문서의 <head>와 <body> 부분(태그, 속성)은 각각 UML 사용 사례 다이어그램과 순서 다이어그램의 요소로 사상된다.

#### 4.1.1 사용 사례 다이어그램

문서의 <head> 부분에 속하는 태그나 속성들은 사용 사례 다이어그램의 요소들이 된다. 이에 대한 규칙은 다음과 같다.

[규칙 1] <channel>이나 <region> 태그의 속성 id는 사용 사례(Use case)의 이름이 된다. 즉, 영역(region)의 개수만큼 사용 사례(Use case)를 만들어서 각 영역을 별도의 객체로서 관리한다.

```

예 1) <head>
<layout>
<region id="a" top="50" />
<region id="b" top="100" />
</layout>
</head>
    
```

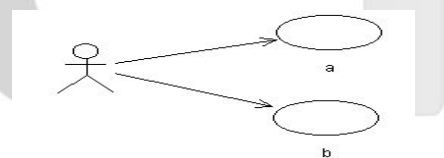


그림 1. 예 1)의 사용 사례 다이어그램  
Fig. 1 Use case diagram of example 1)

#### 4.1.2 순서 다이어그램

4.1.1절의 각 사용 사례에 대해 순서 다이어그램을 삽입할 수 있다.

다음은 순서 다이어그램에 대한 규칙들이다.

- ① 미디어 태그 : 미디어 태그나 하이퍼링크 태그는 순서 다이어그램의 객체를 생성한다. 이에 대한 규칙은 다음과 같다.

[규칙 2] 속성 “src”나 “href”는 객체를 생성한다.

예 2) <a href="http://www.cwi.nl/some.smi">

http://www.cwi.nl/some.smi

그림 2 예 2)의 순서 다이어그램  
Fig. 2 Sequence diagram of example 2)

[규칙 3] 속성 “dur”는 시간제한이 된다. 단, 속성 “begin”, “end”가 나오는 경우 시간제한은 “(dur = end - begin)”이다.

② 동기 태그 : 동기 태그는 메시지의 형태를 결정하는데 이에 대한 규칙은 다음과 같다.

[규칙 4] <par> 태그는 broadcast 메시지가 되고 <seq> 태그는 peer-to-peer 메시지 형태가 된다. 여기서, broadcast, peer-to-peer 메시지는 각각 par(src), seq(src)가 된다.

예 3) <par>  
<audio src="audio.rm"/>  
<video src="video.rm"/>  
<textstream src="stockticker.rtx"/>  
</par>

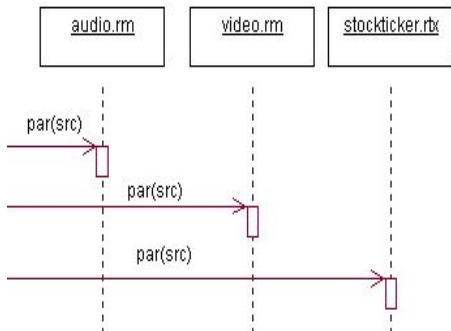


그림 3 예 3)의 순서 다이어그램  
Fig. 3 Sequence diagram of example 3)

예 4) <seq>  
<audio src="joe-audio"/>  
<audio src="tim-text"/>  
<audio src="sam-audio"/>  
</seq>

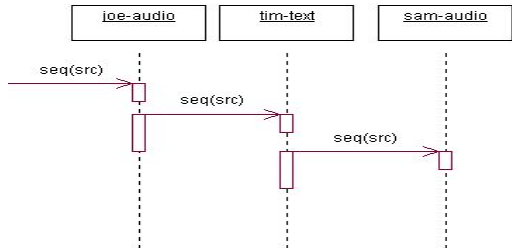


그림 4 예 4)의 순서 다이어그램  
Fig. 4 Sequence diagram of example 4)

③ 선택 태그 : <layout>이나 미디어 태그를 선택할 수 있는 것으로 다음과 같이 정의한다.

[규칙 5] <switch> 태그는 temp(임시) 객체를 생성한다. 임시 객체는 <switch>, </switch> 사이에 나열되는 객체들 중 하나가 되므로 해당 객체들을 주석에 덧붙인다.

예 5) <switch>  
<audio src="joe-audio-english"  
system-language="en" />  
<audio src="joe-audio-korean"  
system-language="kr" />  
</switch>

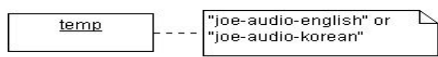


그림 5 예 5)의 순서 다이어그램  
Fig. 5 Sequence diagram of example 5

④ 하이퍼링크 태그 : 하이퍼링크 태그 <a>나 <anchor>의 경우는 유사한 형태를 갖으나 <anchor>의 경우 속성 “coords”와 “begin”, “end” 값에 대한 처리가 필요하다.

[규칙 6] <a> 태그는 해당 객체에 메시지를 보내고 다시 받는다.

```

예 6) <a href="http://www.cwi.nl/som.smi">
<video src="rtsp://foo.com/graph.imf:
region="1_window"/>
</a>
    
```

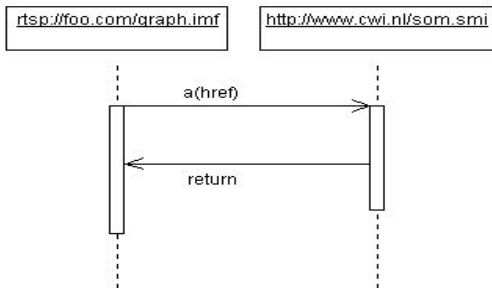


그림 6. 예 6)의 순서 다이어그램  
Fig. 6 Sequence diagram of example 6)

[규칙 7] <anchor> 태그는 해당 객체에 메시지를 보내고 받는다. 속성 “begins”과 “end”가 존재하는 경우 “(end - begin)” 제한조건을 갖는다.

```

예 7)
<video src="http://www.w3.org/CoolStuff">
<anchor href="http://www.w3.org/
AudioVideo" begin="0s" end="5s"/>
</video>
    
```

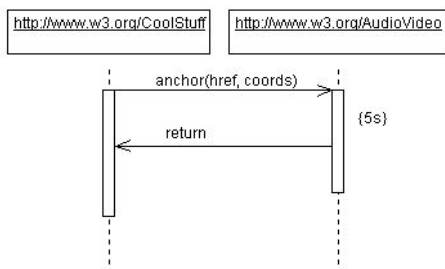


그림 7. 예 7)의 순서 다이어그램  
Fig. 7 Sequence diagram of example 7)

4.2 형식 모델

본 절은 4.1.2절의 규칙들에 의해 모델링한 순서 다이어그램을 데이터베이스 스키마로 표현하기 위해 일반적인 객

체지향 데이터베이스에서 지원하지 않는 확장된 스키마 형태를 정의한다. 이를 위해 몇 가지 가정을 한다.

[가정 1] 기본 타입(integer, text 등)의 집합은 dom이고, 애트리뷰트의 집합은  $A = \{ a_1, a_2, \dots, a_n \}$ , 객체의 집합은  $O = \{ o_1, o_2, \dots, o_n \}$ 이다.

[가정 2] 객체 집합 O의 값들은  $V = \{ v_1, \dots, v_i, \dots, v_n \}$ 이다. 단, V의 원소  $v_i$ 는 nil이거나 dom 혹은 O의 각 원소들이 되고, 튜플(tuple)  $[ a_1 : v_1, \dots, a_n : v_k ]$ 도 V의 원소들이 된다.

[가정 3] 클래스의 타입τ의 언어(interpretation)는 dom(τ)이다.

위와 같은 가정을 기반으로 형식 모델을 정의하고, 각 정의에 대해 SMIL 문서의 예를 든다.

4.2.1 동기 태그

먼저, 동기 태그 <par>, <seq>에 관한 정의들이다.

[정의 2] 튜플  $[ a_1 : \tau_1 \parallel \dots \parallel a_n : \tau_n ]$ 은 병렬(parallel) 타입이다. 즉,  $\text{dom}([ a_1 : \tau_1 \parallel \dots \parallel a_n : \tau_n ]) = \{ [ a_1 : v_1, \dots, a_n : v_n, \dots, a_{n+1} : v_{n+1} ] \mid v_i \in \text{dom}(\tau_i), i = 1, \dots, n; l \geq 0 \}$ 이다.

[정의 3] 튜플  $[ a_1 : \tau_1, \dots, a_n : \tau_n ]$ 은 순차(sequence) 타입이다.

즉,  $\text{dom}([ a_1 : \tau_1, \dots, a_n : \tau_n ]) = \text{dom}([ a_1 : \tau_1, a_2 : \tau_2 ]) + \text{dom}([ a_2 : \tau_2, a_3 : \tau_3 ]) + \dots + \text{dom}([ a_{n-1} : \tau_{n-1}, a_n : \tau_n ]) = \cup \{ \text{dom}([ a_i : \tau_i ]) \mid 1 \leq i \leq n \}$ 이다

4.2.2 하이퍼링크 태그

다음은 하이퍼링크 <a>나 <anchor> 태그와 관련된 정의이다.

[정의 4] 튜플  $[ a : \tau ]$ 에서 a가 ‘HREF’이면 링크 타입이다. 즉,  $\text{dom}([ a : \tau ]) = \{ [ a : v ] \mid v \in \text{dom}(\tau) \}$ 이다

### 4.3 다이어그램 모델링 함수

SMIL 문서에 대한 사용 사례와 순서 다이어그램에 관련된 함수는 다음과 같이 정의된다. 이 함수들은 문서의 구조를 비교, 변형시키기 위한 목적으로 사용할 수 있다. 이에 대한 가정과 정의는 다음과 같다.

[가정 4] 사용 사례 다이어그램의 집합은  $U$ 이며, 순서 다이어그램은  $(T, f)$ 의 쌍으로 이루어진다. 단,  $T$ 는 순서 다이어그램의 집합이고, 함수  $f : T \rightarrow T : \forall t \in T$ 이다.

#### 4.3.1 사용 사례 다이어그램 함수

다음은 사용 사례 다이어그램과 관련된 함수이다.

[정의 5] 함수  $n : U \rightarrow \text{Integer} : \forall u \in U$  는 사용 사례의 개수를 나타낸다.

#### 4.3.2 순서 다이어그램 함수

순서 다이어그램에서 시간적 동기와 하이퍼링크 개념을 포함한 객체의 모델링 함수는 다음과 같이 정의한다.

[정의 6] 함수  $\text{par}_o : \tau \rightarrow f(\tau) : \forall t \in \tau$  는 동기 태그  $\langle \text{par} \rangle$ 에서 동시에 발생하는 객체들을 나타낸다.

[정의 7] 함수  $\text{seq}_o : \tau \rightarrow f(\tau) : \forall t \in \tau$  는 순서 태그  $\langle \text{seq} \rangle$ 에서 다음에 발생할 객체들을 나타낸다.

[정의 8] 함수  $\text{href}_o : \tau \rightarrow f(\tau) : \forall t \in \tau$  는 하이퍼링크 태그  $\langle \text{a} \rangle$ 와  $\langle \text{anchor} \rangle$ 에서 해당 객체가 수행되는 동안 참조될 객체를 나타낸다.

### 4.4 사상 결과

SMIL 문서를 시스템에 적용시켜 생성되는 파싱 테이블과 각종 다이어그램(사용사례, 순서, 협력, 클래스 다이어그램), 그리고, 객체지향 코드를 제시한다.

#### 4.4.1 입력

(그림 8)는 시스템에 입력이 되는 SMIL 문서 예제이다.

```

<smil>
<head>
  <layout type="text/smil-basic">
    <channel id="left-video" left="20" top="50" />
    <channel id="left-text" left="20" top="120" />
    <channel id="right-video" left="150" top="50" />
    <channel id="right-text" left="150" top="120" />
  </layout>
</head>
<body>
  <par>
    <image src="bg" />
    <seq>
      <par>
        
        <text src="graph-text" channel="left-text" />
      </par>
      <par>
        <a href="http://www.w3.org/People/Berners-Lee">
          <video src="tim-video" channel="left-video" />
          <text src="tim-text" channel="left-text" />
        </a>
      </par>
    </seq>
    <seq>
      <audio src="joe-audio" />
      <audio src="tim-audio" />
    </seq>
    <video id="jv" src="atso://www.w3.org/video/joe-video.mpg" channel="right-video" />
    <text src="http://www.w3.org/video/joe-text.txt" channel="right-text" />
  </par>
</body>
</smil>
    
```

그림 8. 예제 프로그램  
Fig. 8 Example program

#### 4.4.2 파싱 테이블

파싱 테이블은 HEAD 부분과 BODY 부분으로 나누어 생성된다. <표 1>은 HEAD 부분의 파싱 테이블이다.

표 1. HEAD 부분의 파싱 테이블  
Table1. Pathing table of head part.

| 태그      | 속성   | 값               |
|---------|------|-----------------|
| layout  | type | text/smil-basic |
| channel | id   | left-video      |
|         | left | 20              |
|         | top  | 50              |
| channel | id   | left-text       |
|         | left | 20              |
|         | top  | 120             |
| channel | id   | right-text      |
|         | left | 150             |
|         | top  | 50              |
| channel | id   | right-text      |
|         | left | 150             |
|         | top  | 120             |

<표 2>는 BODY 부분의 파싱 테이블을 나타낸다.

표 2 BODY 부분의 파싱 테이블  
Table 2 Pathing table of body part

| 동기   | 태그        |           | 속성         | 값                                     |                                      |            |
|------|-----------|-----------|------------|---------------------------------------|--------------------------------------|------------|
|      | 미디어 또는 동기 |           |            |                                       |                                      |            |
|      | 동기        | 미디어 또는 동기 |            |                                       |                                      |            |
| par  | img       |           | src        | bg                                    |                                      |            |
|      | par       | img       | src        | graph                                 |                                      |            |
|      |           |           | channel    | left-video                            |                                      |            |
|      |           |           | dur        | 60s                                   |                                      |            |
|      |           |           | src        | graph-text                            |                                      |            |
|      | seq       | text      | channel    | left-text                             |                                      |            |
|      |           |           | a          | href                                  | http://www.e3.org/People/Berners-Lee |            |
|      |           |           | par        | video                                 | src                                  | tim-video  |
|      |           |           |            |                                       | channel                              | left-video |
|      | seq       | audio     | src        | joe-audio                             |                                      |            |
|      |           |           | src        | tim-audio                             |                                      |            |
|      | video     | text      | id         | jv                                    |                                      |            |
|      |           |           | src        | rtsp://www.w3.org/video/joe-video.mpg |                                      |            |
|      |           |           | channel    | right-video                           |                                      |            |
|      |           |           | src        | https://www.w3.org/video/joe-text.txt |                                      |            |
| text | text      | channel   | right-text |                                       |                                      |            |

#### 4.4.3 다이어그램

다이어그램 생성기에 의해 생성된 사용사례, 순서, 협력, 그리고, 클래스 다이어그램을 나타낸다. 파싱 테이블의 HEAD 부분에 의해 생성되는 사용 사례 다이어그램은 그림 9와 같다.

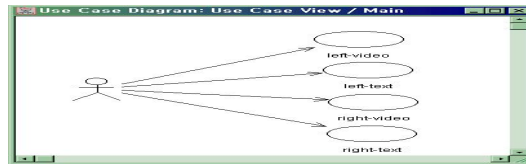


그림 9. 사용 사례 다이어그램  
Fig. 9 Use Case diagram

(그림 9)의 4개의 사용 사례에 대해 각각 삽입된 순서 다이어그램은 (그림 10)과 같으며, 순서 다이어그램들은 파싱 테이블의 BODY 부분에 의해 생성된다.

(그림 10)의 각 순서 다이어그램에 해당하는 협력 다이어그램은 (그림 11)과 같다. 하나의 순서 다이어그램은 하나의 협력 다이어그램을 자동 생성한다.

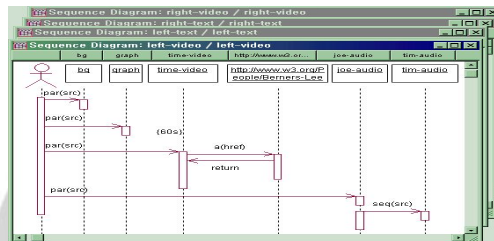


그림 10. 순서 다이어그램들  
Fig. 10 Sequence Diagrams

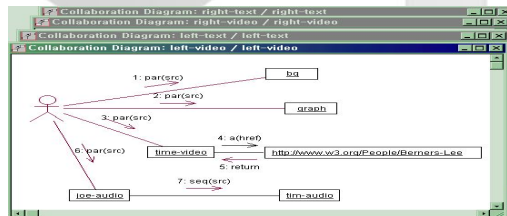


그림 11. 협력 다이어그램들  
Fig. 11 Collaboration diagrams

(그림 10)의 순서 다이어그램으로부터 클래스를 추출, 클래스 다이어그램을 생성한다. 필요에 따라 순서 다이어그램 상의 메시지는 오퍼레이션이 되며, 각 클래스는 기본 타입 클래스(Image, Video, Audio, Text 등)로부터 상속을 받는다.

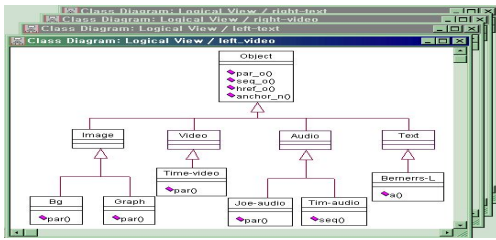


그림 12. 클래스 다이어그램들  
Fig. 12 Class diagrams

(그림 10)의 순서 다이어그램으로부터 클래스를 추출, 클래스 다이어그램을 생성한다. 필요에 따라 순서 다이어그램 상의 메시지는 오퍼레이션이 되며, 각 클래스는 기본 타입 클래스(Image, Video, Audio, Text 등)로부터 상속을 받는다. (그림 12)는 각 순서 다이어그램에 대한 클래스 다이어그램이다.

#### 4.4.4 C++ 코드

본 시스템은 클래스 다이어그램의 각 클래스에 대해 객체지향 코드인 C++ 코드(.cpp)는 (그림 13)과 같고 헤더 파일(.h)을 (그림 14)과 같이 생성되었다. (그림 12)의 클래스 다이어그램에서 “Bg”클래스에 대한 C++ 코드(Bg.cpp)는 다음과 같다.

```
// Bg
#include "Bg.h"
// Class Bg
Bg::Bg()
Bg::Bg(const Bg &right)
Bg::~Bg()
Bg & Bg::operator=(const Bg &right)
int Bg::operator==(const Bg &right) const
int Bg::operator!=(const Bg &right) const
###Other Operations (implementation)void Bg::par ()
```

그림 13. C++ 코드  
Fig. 13 C++ code

위의 클래스에 대한 헤더파일(Bg.h)은 (그림 14)와 같다.

```
#ifndef Bg_h
#define Bg_h 1
// Image
#include "Image.h"
class Bg : public Image
### Inherits: <unnamed>:%384F11FC026C
{
public:
### Constructors (generated)
Bg();
Bg(const Bg &right);
### Destructor (generated)
~Bg();
### Assignment Operation (generated)
Bg & operator=(const Bg &right);
### Equality Operations (generated)
int operator==(const Bg &right) const;
int operator!=(const Bg &right) const;
### Other Operations (specified)
### Operation: par%384F1160334
void par ();
};
#endif
```

그림 14. 헤더파일  
Fig. 14 Header file

#### 4.5 비교 분석

본 연구의 모델링을 평가하기 위해 기존에 제안된 바 있는 대표적인 멀티미디어 모델링 방법들인 시간축 (Time-line) 모델, Firefly 모델, 시간(Timed) 페트리네트(Petri net), 객체 합성(Object Composition) 페트리네트과 본 연구와 비교 분석하였다. 이러한 모델에 대해 멀티미디어 데이터 모델링의 일반적 기준[9][10]을 중심으로 본 연구와 비교해 보면 다음과 같다.

첫째, 시간 관계 표현 능력이다. 본 연구는 사건 기반 모델이나, 시간 제한조건을 사용하여 시간 관계 표현이 가능하므로 시간 관계 표현 능력이 좋다.

둘째, 추상화 정도인데, 시간 간격기반 모델은 미디어간의 시간 관계를 직접 표현하기 때문에 사건기반 모델보다 추상화 단계가 높다. 따라서, 본 연구는 사건기반 모델이므로 비교적 추상화 단계가 낮다.

셋째, 동적 객체 표현 능력이다. 그래프 방법인 페트리네트 등은 객체의 동적인 행위를 구분하지 못한다. 이에 비해 본 연구는 동기화를 주고받는 객체의 구별이 가능하므로 동적 표현 능력이 좋다.

넷째, 공간 정보 표현 능력이다. 본 연구는 공간에 해당하는 속성을 인식하여 공간을 분할하여 사용 사례

로 표현하므로 객체 합성 페트리네트와 함께 공간 정보 표현 능력이 좋다.

다섯째, 사용자와의 상호 작용인데, 본 연구는 사용자가 해당 객체에서 클릭하면 새로운 객체로 분기 가능한 하이퍼링크 부분을 지원하므로 다른 모델에서 지원되지 않는 사용자와의 상호작용이 부분적으로 가능하다.

여섯째, Perez-Luque의 멀티미디어 동기화 모델을 위한 참조 기준[11]이다. 시간 모델 특성의 구성요소는 기본 시간 단위, 이와 관련된 문맥 정보, 그리고, 시간 표현 방법의 유형이다. 즉, 기존 시간 단위는 시점(instant)과 시간(interval)으로 분류되고, 문맥 정보는 정량적(quantitative)과 정성적(qualitative) 정보로 분류된다. 본 연구는 정성적 시점으로, 정량적 정보와 정성적 정보를 함께 갖는 특징이 있으며, 모델은 사건의 전개의 순서와 제한시간에 대한 정보를 가지고 있으며, 정성적 정보로 인해 비결정적 시나리오 표현이 가능하다.

위의 내용을 종합해보면 본 연구의 모델링 방법은 추상화 정도는 낮으나 시간 관계와 공간 정보, 그리고 동적 객체 표현 능력이 좋으며, 사용자와의 상호 작용이 가능하다. 그리고, 정성적 시점으로 멀티미디어 동기화 참조기준에 의한 표현능력이 좋으며, 태그를 분석할 수 있는 장점을 갖는다.

## V. 결론 및 향후 연구과제

현재 인터넷을 통해 문서의 교환이 활발히 이루어짐에 따라 멀티미디어 문서 교환을 위해 SMIL 문서가 제정되었다. 그러나, 아직까지 SMIL 문서를 모델링하는 도구(tool)가 개발되지 않아 SMIL 문서 개발 시에 복잡한 멀티미디어 간의 시간적인 동기화 구조를 일관성있게 표현하고 문서화하기 어렵다. 따라서, 본 연구에서는 UML 사용 사례, 순서, 협력, 클래스 다이어그램을 이용하여 SMIL 문서의 객체들간의 시간적 동기화와 협력 관계, 논리적 관계를 표현하였다. 이를 위해 본 연구에서는 SMIL 문서의 태그나 속성 등을 UML 사용 사례 다이어그램과 순서 다이어그램에

적절하게 사상시키는 규칙과 알고리즘을 제안하였으며, SMIL 문서를 위한 객체지향 데이터베이스 스키마 형태인 형식 모델과 다이어그램 모델링 함수를 제안하였다.

본 연구의 사상 규칙과 알고리즘을 이용하면 SMIL 문서의 객체들간의 상호 관계를 쉽게 파악할 수 있으며, 각종 OODB 스키마 생성의 기반이 된다. 그리고, 형식 모델과 모델링 함수는 SMIL 문서를 객체지향 데이터베이스에 저장, 관리하기 위한 기반이 될 것이다.

향후에 각종 OODB 저장을 위한 스키마 생성과 검색을 위한 질의어도 개발해야 할 것이다.

## 참고문헌

- [1] International Standards Organization(ISO), "Information Processing-Text and Office Systems -Standard Generalized Markup Language(SGML)," ISO 8879, 1986.
- [2] W3C, "Extensible Markup language(XML) 1.0," 1998, <http://www.w3.org/TR/1998/REC-xml-19980210>
- [3] W3C, "Synchronized Multimedia Integration Language(SMIL) 1.0 Specification," W3C, June 1998, <http://www.w3.org/TR/1998/REC-smil-19980615/>
- [4] Natanya Pitts-Moulis, Cheryl Kirk, "XML black book", The Coriolis Group, Inc., 1999
- [5] 박인호, 한에노, 정은주, 김은정, 배종민, 강현석, 김완석, "XOMT : SGML DTD 설계를 위한 객체 다이어그램 기법", 정보과학회 논문지(C), 제3권, 제3호, pp. 228-237, 1997. 6
- [6] 하얀, 황용주, 김용성, "SGML DTD로부터 UML 클래스 다이어그램으로의 사상 알고리즘", 정보과학회 논문지, 제26권, 제4호, pp. 508-520, 1999. 4.
- [7] 채원석, 하얀, 김용성, "UML 클래스 다이어그램을 이용한 XML 문서 구조 다이어그램", 정보처리학회 논문지, 제6권, 제10호, pp. 2670-2679, 1999. 10.
- [8] Lloyd Rutledge, John F. Buford, John L. Rutledge, Modeling Techniques for HyTime, 1996, <http://www.sil.org/sgml/rutledgeModelingHy.htm>
- [9] 류시원, 김창룡, 차광호, 정진호, "멀티미디어 데이터베이스 시스템에서의 멀티미디어 데이터 모델링에 대

한 고찰”, 1997, <http://www.multimedia.or.kr/newtech/NT14/nt2.htm>

- [10] 허수철, 성혜경, “게임공간의 분류와 시나리오의 시간 및 공간 동기화 표현법,” 정보처리학회 논문지, 제6권, 제10호, pp. 2630-2641, 1999. 10.
- [11] 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.



**채 원 석**

1984년 원광대학교 전자공학과 (공학사)  
 1986년 인하대학교 대학원 전산학과 (이학석사)  
 2000년 전북대학교 대학원 전산통계과(이학박사)  
 1989년~현재 원광보건대학 정보컨텐츠과 교수  
 <관심분야> SGML/XML, 데이터베이스, 멀티미디어시스템 등

