

UML 사용 사례 다이어그램과 순서 다이어그램을 이용한 SMIL 문서 동기화

(Synchronization of SMIL Documents using UML Use Case Diagrams and Sequence Diagrams)

채원석^{*} 하안^{**} 김용성^{***}
(Won-Seok Chae) (Yan Ha) (Yong-Sung Kim)

요약 SMIL(Synchronized Multimedia Integration Language)는 멀티미디어 객체의 집합으로 시간적인 동기화를 표현할 수 있는 마크업 언어이다. 본 연구에서는 SMIL 문서를 시간에 따라 동기화 표현을 명확히 할 수 있도록 객체 지향 기반의 시각화 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 SMIL 문서 개발자들이 손쉽게 문서 구조 파악 및 개발을 할 수 있으며, 형식 모델과 다이어그램 모델링 함수는 객체지향 데이터베이스 문서 관리의 기반이 된다는 것이다.

Abstract SMIL(Synchronized Multimedia integration Language) allows integrating a set of independent 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). 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 easily generate them using this rules and algorithm. And, the formal models and modeling functions provide an environment for processing object-oriented documents.

1. 서론

이 기종 시스템간의 문서의 교환을 목적으로 1986년에 국제 표준 기구(International Standard Organization)에서 SGML(Standard Generalized Markup Language)을 제정하였다[1]. SGML은 문서의 논리적 구조와 내용을 기술하기 위한 마크업 언어로 CALS/EC 등 업계의 표준으로 사용되어 왔다. 그러나, SGML의 기능은 너무 복잡하기 때문에 이를 지원하는 소프트웨

어 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)이 등장하였다. 1996년에 W3C(World Wide Web Consortium)에서 제정한 XML은 기존의 HTML에 사용자가 새로운 태그를 정의할 수 있는 기능을 추가한 SGML의 부분집합(subset)이다[2].

이러한 XML은 특정 응용 목적에 맞게 엘리먼트, 애트리뷰트, 엔티티를 정의할 수 있는 융통성(flexibility)을 갖는데, 그 예로 CDF(Channel Definition Format), OSD(Open Software Description), RDF(Resource Description Framework), WIDL(Web Interface Definition Language) 등이 있다. 그 중에서 멀티미디어 XML 기반 응용으로 SMIL(Synchronized Multimedia

* 정회원 : 원광보건대학 컴퓨터응용개발과 교수
wschae@sky.wkhc.ac.kr

** 학생회원 : 전북대학교 전산통계학과
yanha@cs.chonbuk.ac.kr

*** 종신회원 : 전북대학교 컴퓨터과학과 교수
yskim@moak.chonbuk.ac.kr

논문접수 : 1999년 7월 20일

심사완료 : 2000년 1월 21일

Integration Languages)이 있다. SMIL은 멀티미디어 내용을 웹에 전송할 수 있도록 1998년 6월 W3C에서 제정하였는데[3], 인터넷에서 TV와 같은 품질의 내용을 전송할 뿐만 아니라 하이퍼링크 기능을 추가하여 빠르게 웹 기반 내용으로 이동하며, 표현을 동기화시킨다[4].

앞으로 SMIL의 사용이 급격히 늘어날 것으로 예상되나 아직까지 SMIL 문서를 개발하기 위한 보조적인 도구나 소프트웨어가 개발되지 않았기 때문에 복잡한 SMIL 문서를 개발하기가 무척 어려운 실정이다. 특히 문서 개발자들은 일일이 시간에 따른 미디어들간의 관계를 파악해야 하며, 개발자들간에 공통된 도식화를 사용하지 않아 문서 교환이 용이하지 않다. 따라서, 본 연구는 UML을 이용하여 SMIL 문서에 대한 비주얼(visual) 모델링 방법을 제안한다.

최근 객체지향적인 모델링 언어의 표준이 되고 있는 UML(Unified Modeling Language)은 사용자 관점의 사용 사례 뷰(View)를 중심으로 여러가지 뷰를 제시하고 있다. 사용 사례 뷰에는 사용사례, 순서, 그리고 협력 다이어그램이 포함되며, 사용사례 다이어그램은 사용자 관점에서 사용사례를 나타내며, 각 사용사례에 순서, 협력 다이어그램 등을 추가할 수 있다. 순서, 협력 다이어그램은 시간의 흐름에 따른 객체 사이의 관계를 잘 나타내 주는 상호작용 다이어그램[5]으로, 기타 다른 뷰에 속하는 다이어그램들의 생성을 위한 기반이 된다.

따라서, 사용사례 다이어그램을 SMIL 문서에 적용하면 사용자 관점에서 사용사례를 구별할 수 있으며, 순서 다이어그램에서 객체를 추출하고 시간적 동기화를 표현하고, 협력 다이어그램에서 객체들의 협력 관계를 파악할 수 있다. 이것은 문서 개발자로 하여금 복잡한 SMIL 문서의 개발을 용이하게 하며, 객체로부터 클래스 추출, 클래스 다이어그램을 생성하며, 객체지향 스키마로 변환과 객체지향 시스템 개발을 손쉽게 하도록 한다.

또한, 본 연구에서는 SMIL 문서의 객체지향 데이터베이스 스키마 형태인 형식 모델과 다이어그램 모델링 함수를 제안하므로써 SMIL 문서에 대한 객체지향적인 시각화 뿐만 아니라 데이터베이스 문서로의 저장, 관리를 용이하도록 한다. 아울러, 본 연구의 모델링의 평가를 위해 기존의 멀티미디어 모델들의 일반적 기준에 의거하여 비교, 분석해본다.

2. 관련 연구

지금까지 SGML과 XML, 그리고 HyTime의 모델링에 대한 연구는 꾸준히 진행되어 왔으나 SMIL에 대한

연구는 이루어지지 않았다.

SGML DTD를 모델링하는 방법으로는 구조도(Structure Diagram)[6]와 XOMT(eXtended Object Modeling Technique) 다이어그램[7], UML 클래스 다이어그램(Class Diagram)[8]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 SMIL를 모델링하게 되면 시간적인 동기화와 하이퍼링크는 표현할 수가 없다.

XML에 관한 모델링으로 UML 클래스 다이어그램을 이용한 방법이 있는데, 이것은 SGML과 다른 XML 문법이나 링크 표현을 지원한다[9]. 그러나, 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.

그리고, XML 문서를 위한 표준화된 API를 제공하여 웹 문서에 접근하고 조작하기 위한 방법으로 DOM(Document Object Model)[2][4]이 있는데, 이것은 문서의 논리적 구조를 정의하는 객체 기반 구조이기는 하나 트리 계층 구조의 형태이므로 시간의 흐름에 따른 객체 관계 등을 나타낼 수가 없다.

한편, HyTime 문서를 모델링하는 연구로는 HOMT 다이어그램[10], [11] 등이 있다. HOMT는 XOMT 다이어그램을 확장하여 HyTime 문서 표현을 위한 노테이션을 추가하였으나 단지 별도의 부 원도우를 삽입하였을 뿐 클래스들 간의 하이퍼링크 관계는 도식화하지 못하는 단점이 있으며, [11]에서 제안한 HyTime 문서 구조 다이어그램은 HyTime 애트리뷰트가 포함되며, 참조할 엘리먼트에 IDREF 애트리뷰트를 연결하는 점선은 존재하나, 어느 일정 시간 동안 링크를 지원하는 등 시간의 흐름에 따른 하이퍼링크 관계는 표현하지 못하는 단점이 있다.

이와 같이 기존의 연구들을 살펴본 결과, 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 사

용사례, 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. 그리고, 기존의 SGML/XML 문서에 관해 제시한 바 있는 형식 모델과 모델링 함수[8][9]를 확장하여 SMIL 문서의 속성에 맞게 제안한다. 즉, 시간적 동기와 하이퍼링크에 대해 형식 모델을 제안하여 데이터베이스로 저장을 용이하게 하며, 객체의 발생 순서 등에 관한 모델링 함수를 제안하여 문서 구조 파악, 통합, 변형 등을 가능하게 한다.

3. 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.

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

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>, , <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[12].

3.2 UML

UML(Unified Modeling Language)는 Rumbaugh의 OMT 방법론, Booch의 Booch 방법론, 그리고, Jacobson의 OOSE 방법론을 통합하여 만든 것으로 현재 많은 CASE 도구들이 이를 지원하고 있다[13]. UML은 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].

사용사례 뷰는 외부 사용자에게 의해서 파악되는 시스템의 기능적인 측면을 묘사한 것으로 일반적인 사례를 표현한다. 따라서, 다른 관점에 비해 쉽게 시스템을 기술하며 다른 뷰를 통한 시스템 개발의 기본이 된다. 이에 해당하는 다이어그램으로는 사용사례, 순서, 협력 다이어그램이다. 이러한 사용사례 뷰와는 대조적으로 논리적 뷰는 시스템의 내부 구조를 분석하고 정적 구조와 동적인 구조를 나타낸다. 정적인 구조를 나타내는 다이어그램으로는 클래스 다이어그램 등이 있다.

3.2.1 사용사례 다이어그램

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

사용사례 다이어그램의 표기는 외부 사용자와 사용사례들의 집합, 그리고 그들 간의 관계로 이루어진 그래프이며 객체를 사용자의 요구 사항으로부터 자연스럽게 유도하고 사용자와의 의사 소통을 정형화된 다이어그램 형식으로 나타내는 장점이 있다.

3.2.2 순서 다이어그램

순서 다이어그램은 다른 객체들 사이의 메시지 흐름

의 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 표현 가능한 항목들을 나열하고 설명한다[14].

(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 메시지(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. 객체의 표현은 직사각형으로 하며 이들 간의 관계를 메시지를 통해 표시한다. 이 때, 메시지는 일련 번호를 갖는다.

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 표기하며, 서브클래스(subclass)는 슈퍼클래스(superclass)로부터 애트리뷰트와 오퍼레이션 리스트를 상속(inheritance)받는다.

4. SMIL 문서 동기화

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

4.1 사상 규칙

SMIL 문서로부터 UML 사용사례 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>

```

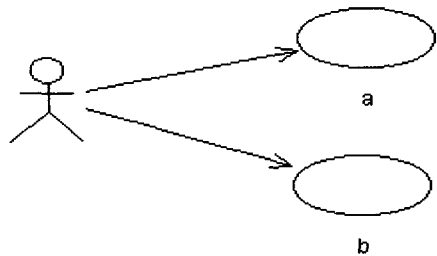


그림 1 예 1)의 사용 사례 다이어그램

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) `<text region="a" src="text.html">`

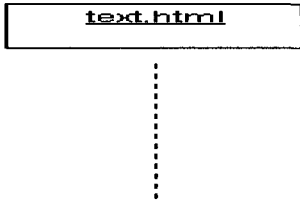


그림 2 예 2)의 순서 다이어그램

예 3) ``

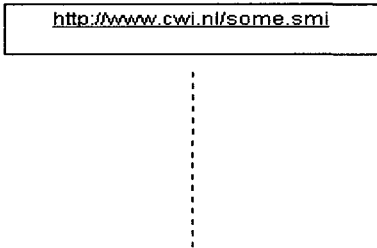


그림 3 예 3)의 순서 다이어그램

[규칙 3] 속성 “dur”는 시간 제한이 된다. 단, 속성 “begin”, “end”가 나오는 경우 시간 제한은 “{dur = end - begin}”이다.

예 4) `<text region id="a" src="text.html" dur="10s">`

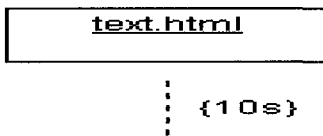


그림 4 예 4)의 순서 다이어그램

(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 5) `<par>`

```

<audio src="audio.rm"/>
<video src="video.rm"/>
<textstream src="stockticker.rtx"/>
</par>
    
```

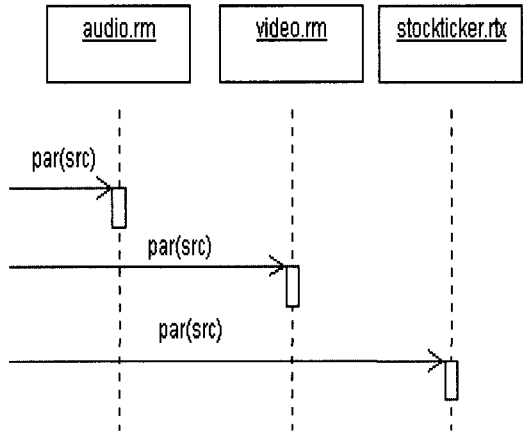


그림 5 예 5)의 순서 다이어그램

예 6) `<seq>`

```

<audio src="joe-audio"/>
<audio src="tim-text"/>
<audio src="sam-audio"/>
</seq>
    
```

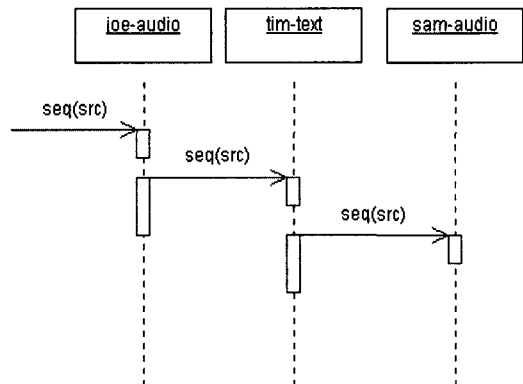


그림 6 예 6)의 순서 다이어그램

(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 7) <switch>
      <audio src="joe-audio-english"
            system-language ="en" />
      <audio src="joe-audio-korean"
            system-language ="kr" />
</switch>
    
```

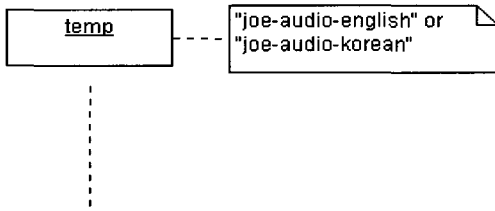


그림 7 예 7)의 순서 다이어그램

(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 8) <a href="http://www.cwi.nl/som.smi">
      <video src="rtsp://foo.com/graph.imf">
    
```

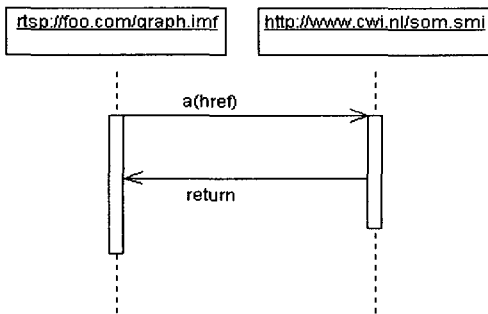


그림 8 예 8)의 순서 다이어그램

```

region="1_window"/n>
    
```


[규칙 7] <anchor> 태그는 해당 객체에 메시지를 보내고 받는다. 속성 "begins"과 "end"가 존재하는 경우 "{end - begin}" 제한조건을 갖는다.

```

예 9) <video src="http://www.w3.org/CoolStuff">
      <anchor href="http://www.w3.org/
            AudioVideo" begin="0s" end="5s"/>
</video>
    
```

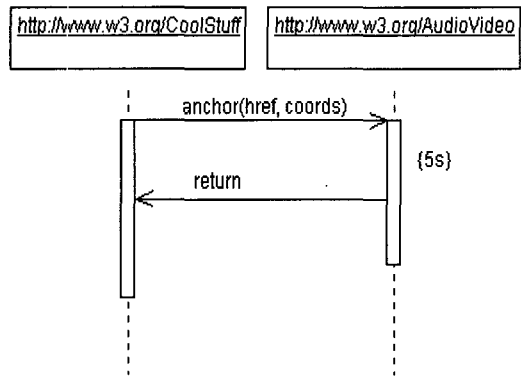


그림 9 예 9)의 순서 다이어그램

4.2 형식 모델

본 절은 4.1.2절의 규칙들에 의해 모델링한 순서 다이어그램을 데이터베이스 스키마로 표현하기 위해 일반적인 객체지향 데이터베이스에서 지원하지 않는 확장된 스키마 형태를 정의한다. 이를 위해 몇 가지 가정을 한다.

[가정 1] 기본 타입(integer, text 등)의 집합은 dom 이고, 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 O 의 값들은 $V = \{ v_1, \dots, v_i, \dots, v_n \}$ 이다. 단, V 의 원소 v_i 는 nil 이거나 dom 혹은 O 의 각 원소들이 되고, 튜플(tuple) $[a_1 : v_1, \dots, a_n : v_k]$ 도 V 의 원소들이 된다.

[가정 3] 클래스의 타입 τ 의 언어(interpretation)는 $dom(\tau)$ 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, $dom([a_1 : \tau_1 \parallel \dots \parallel a_n : \tau_n]) = \{ [a_1 : v_1, \dots, a_n : v_n, \dots, a_{n+l} : v_{n+l}] \mid v_i \in dom(\tau_i), i = 1, \dots, n; l \geq 0 \}$ 이다.

예 10) $[a_1 : \text{audio.rm} \parallel a_2 : \text{video.rm} \parallel a_3 : \text{stockticker.rtx}]$: 애트리뷰트는 각각 a_1, a_2, a_3 이며, 타입은 각각 'audio.rm', 'video.rm', 'stockticker.rtx'이다. 이 때, 모든 애트리뷰트는 동시에 발생한다.

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

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

예 11) $[a_1 : \text{joe-audio}, a_2 : \text{tim-text}, a_3 : \text{sam-audio}]$: 애트리뷰트는 각각 a_1, a_2, a_3 이고, 타입은 각각 'joe-audio', 'tim-text', 'sam-audio'이며 애트리뷰트는 순서적으로 발생한다.

4.2.2 하이퍼링크 태그

다음은 하이퍼링크 $\langle a \rangle$ 나 $\langle \text{anchor} \rangle$ 태그와 관련된 정의이다.

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

예 12) $[\text{HREF} : \text{http://www.cwi.nl/som.smi}]$: 해당 튜플은 링크 타입이 되고 링크된 자원은 객체의 타입이 된다

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.

예 13) $\langle \text{head} \rangle$

```
<layout>
  <region id="a" top="50" />
  <region id="b" top="100" />
</layout>
```

$\langle \text{head} \rangle$

이면 $n(\text{use_case}) = 2$ 이다.

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.

예 14) $\langle \text{par} \rangle$

```
<audio src="audio.rm"/>
<video src="video.rm"/>
<textstream src="stockticker.rtx"/>
</par>
```

이면 $\text{par}_o(\text{audio.rm}) = \{ \text{video.rm}, \text{stockticker.rtx} \}$ 이다.

[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.

예 15) $\langle \text{seq} \rangle$

```
<audio src="joe-audio"/>
<audio src="tim-text"/>
<audio src="sam-audio"/>
</seq>
```

이면 $\text{seq}_o(\text{joe-audio}) = \{ \text{tim-text} \}$ 이고, $\text{seq}_o(\text{tim-text}) = \{ \text{sam-audio} \}$ 이다.

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

예 16) $\langle a \text{ href}="http://www.cwi.nl/som.smi">$

```
<video src="rtsp://foo.com/graph.imf"
  region="1_window"/n>
</a>
```

이면 $\text{href}_o(\text{rtsp://foo.com/graph.imf}) = \{ \text{http://www.cwi.nl/som.smi} \}$ 이다.

[정의 9] 함수 $\text{anchor}_n : \tau \rightarrow \text{Integer} : \forall t \in \tau$ 는 하이퍼링크 $\langle \text{anchor} \rangle$ 태그에서 해당 객체가 수행

되는 동안 참조될 객체의 수를 나타낸다.

```

예 17) <video src="http://www.w3.org/CoolStuff">
      <anchor href="http://www.w3.org/
      AudioVideo" begin="0s" end="5s">
      <anchor href="http://www.w3.org/Style"
      begin="5s" end="10s">
    </video>
  
```

이때 anchor_n(http://www.w3.org/CoolStuff) = 2 이다.

4.4 알고리즘

SMIL 문서를 입력으로 하여 UML 사용 사례와 순서 다이어그램을 생성하는 알고리즘은 다음과 같다.

```

입력 : SMIL 문서
출력 : UML 사용 사례 및 순서 다이어그램

begin
{
  for ( 각 channel 이나 region 태그 )
  { // 영역 개수만큼 사용 사례 생성한다.
    use_case_diagram 생성
    sequence_diagram 삽입
  }
  for (각 sequence_diagram)
  { // 각 사용 사례에 해당하는 순서 다이어그램을
    // 불러온다.
    for ( 각 미디어나 동기 태그 )
    { // 태그의 종류에 따라 객체 또는 임시
      // 객체를 생성한다.
      if ( 태그 속성에 channel(or region)이 없거나
        channel(or region) 속성 값이 use_case 이름과
        같으면 )
        then 객체 생성
      else if ( switch 태그 )
        then 임시(temp) 객체 생성
      }
    for ( 각 객체 )
    { // 생성된 객체에 제한조건 주거나
      // 메시지 처리를 한다.
      if ( dur 속성 )
        then 시간 제한 표시
      if ( par 태그 )
        then broadcast_message 처리
        else if ( seq 태그 )
          then peer_to_message 처리
        else if ( a 태그 )
          then a_send_receive 처리
        else if ( anchor 태그 )
          then {
            anchor_send_receive 처리
            if ( begin, end 속성 )
              then 시간 제한 표시
          }
      }
    }
  }
}
end ;
  
```

4.5 시스템 구현

본 절에서는 전체 시스템 구성과 기능에 대해 설명하고 사상 결과를 제시한다.

4.5.1 시스템 구성

SMIL 문서에 대한 각종 다이어그램과 객체지향 코드 생성에 관한 시스템 구성도는 그림 10과 같다.

그림 10의 각 구성 요소의 기능과 역할을 알아보면 다음과 같다.

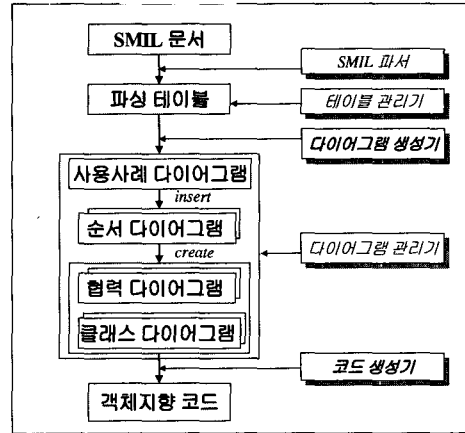


그림 10 시스템 구성도

- (1) SMIL 파서

SMIL 문서를 입력받아 문법의 오류 등을 검사하고, 파싱 테이블을 생성한다.
- (2) 테이블 관리기

생성된 파싱 테이블(HEAD 및 BODY 부분)을 항목 별로 저장, 관리한다.
- (3) 다이어그램 생성기

파싱 테이블을 읽어 들여 다이어그램을 생성한다. HEAD 테이블을 참조하여 사용사례 다이어그램을 생성한다. 각 사용 사례는 BODY 테이블을 이용하여 순서 다이어그램을 삽입하며, 협력 다이어그램은 순서 다이어그램을 통해 자동 생성된다. 그리고, 클래스 다이어그램은 순서 다이어그램에서 클래스를 추출하고 애틀리뷰트와 오퍼레이션을 삽입하여 생성한다.
- (4) 다이어그램 관리기

사용사례, 순서, 협력, 그리고, 클래스 다이어그램 등의 요소들 즉, 사용사례, 클래스, 객체, 메시지 등의 정보 등을 저장, 관리한다.
- (5) 코드 생성기

클래스 다이어그램으로부터 각 클래스에 대해 객체지향 코드를 생성한다.

4.5.2 사상 결과

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

(1) 입력

다음은 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="atstp://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>
    
```

표 1 HEAD 부분의 파싱 테이블

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

(2) 파싱 테이블

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

표 2는 BODY 부분의 파싱 테이블을 나타낸다.

표 2 BODY 부분의 파싱 테이블

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

(3) 다이어그램

다이어그램 생성기에 의해 생성된 사용사례, 순서, 협력, 그리고, 클래스 다이어그램을 나타낸다. 파싱 테이블의 HEAD 부분에 의해 생성되는 사용사례 다이어그램은 그림 11과 같다.

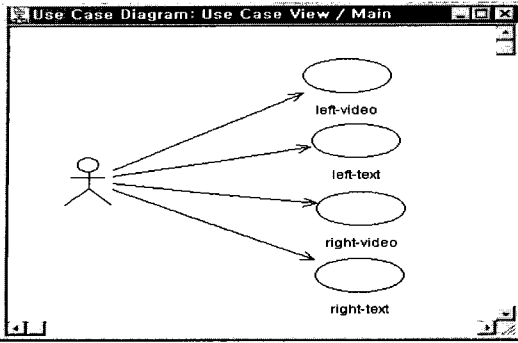


그림 11 사용사례 다이어그램

한 클래스 다이어그램이다.

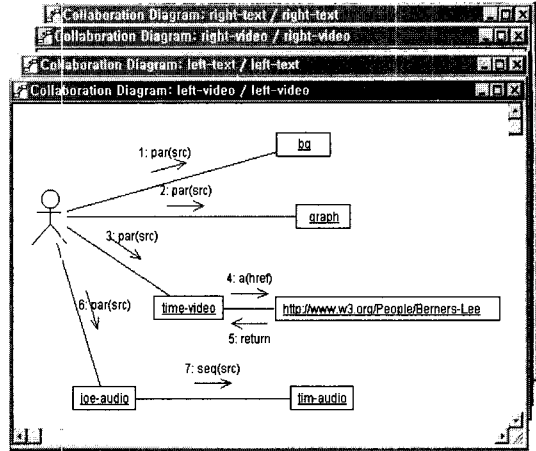


그림 13 협력 다이어그램들

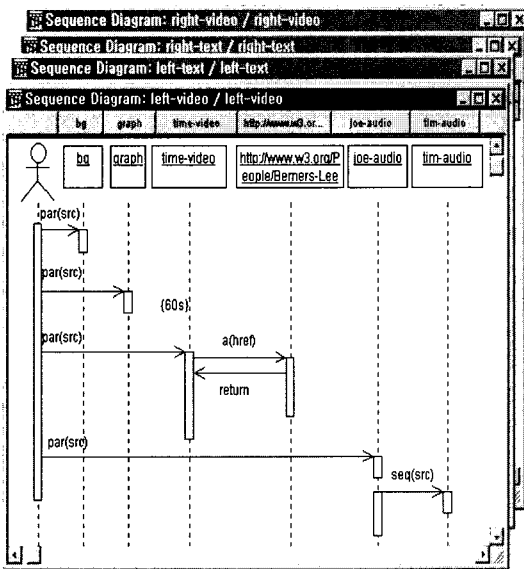


그림 12 순서 다이어그램들

그림 11의 4개의 사용 사례에 대해 각각 삽입된 순서 다이어그램은 그림 12와 같으며, 순서 다이어그램들은 파싱 테이블의 BODY 부분에 의해 생성된다.

그림 12의 각 순서 다이어그램에 해당하는 협력 다이어그램은 그림 13과 같다. 하나의 순서 다이어그램은 하나의 협력 다이어그램을 자동 생성한다.

그림 12의 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 14는 각 순서 다이어그램에 대

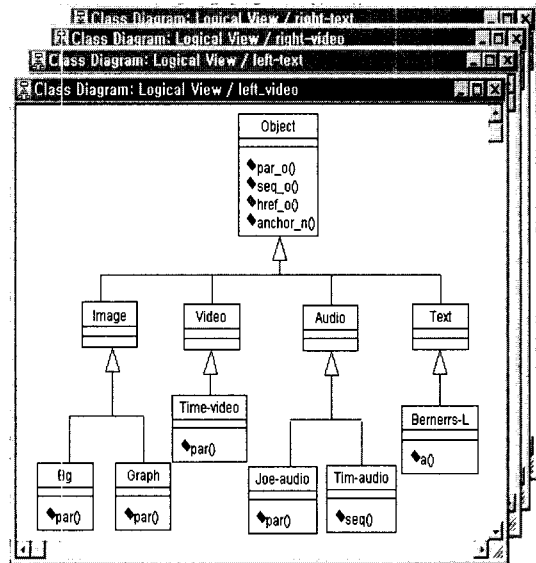


그림 14 클래스 다이어그램들

(4) C++ 코드

본 시스템은 클래스 다이어그램의 각 클래스에 대해 객체지향 코드인 C++ 코드(.cpp)와 헤더파일(.h)을 생성한다. 그림 14의 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 ()
```

위의 클래스에 대한 헤더파일(Bg.h)은 다음과 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%384F11160334
void par ();
};
#endif
```

4.6 비교 분석

본 연구의 모델링을 평가하기 위해 기존에 제안된 바 있는 대표적인 멀티미디어 모델링 방법들에 대해 살펴

보면 다음과 같다. 시간축(Time-line) 모델은 모든 사건(event)을 하나의 절대적인 시간축 상에 표현하는 것이며, Firefly 모델은 사건을 그래프 상의 사각형 노드로 표현하고 미디어 상연(play)과 사건 사이의 관계는 라벨(label)된 간선(edge)으로 표현한다. 시간(Timed) 페트리네트(Petri net)는 미디어 간의 시간 정보를 직접 표현하는 것으로 페트리네트를 이용한다. 객체 합성(Object Composition) 페트리네트는 시간 페트리네트에 공간 정보를 표현하며, 미디어 사이의 시간 관계를 다차원 관계로 표현하도록 확장한 모델이다[15].

이러한 모델에 대해 멀티미디어 데이터 모델링의 일반적 기준[15][16]을 중심으로 본 연구와 비교해 보면 표 3과 같다.

표 3 멀티미디어 모델들 비교

기준 \ 모델	시간 축	Fire fly	시간페트리네트	객체합성 페트리네트	본 연구
시간관계 표현 능력 (A:사건기반 B:시간간격기반)	△ (A)	O (A)	O (B)	O (B)	O (A)
추상화 정도	X	X	O	O	X
동적 객체 표현 능력	X	X	X	X	O
공간정보 표현 능력	X	X	X	O	O
사용자와의 상호작용	X	X	X	X	△
멀티미디어 동기화 모델을 위한 참조 기준에 의한 표현능력 (C:정량적날짜, D:정성적시점, E:정성적시간간격)	△ (C)	O (D)	O (E)	O (E)	O (D)
태그 분석 능력	X	X	X	X	O

* 'O'는 좋음(또는 높음), '△'는 보통, 'X'는 낮음(또는 없음)

표 3에서 제시한 각 기준을 살펴보면 다음과 같다. 첫째, 시간 관계 표현 능력이다. 보통 시간은 하나의 시점 형태로 표현되는 사건 기반(event-based)과 두 시점 사이의 관계에 의해 표현되는 시간 간격기반 모델이 있다. 본 연구는 사건 기반 모델이나, 시간 제한조건을 사용하여 시간 관계 표현이 가능하므로 시간 관계 표현 능력이 좋다. 둘째, 추상화 정도인데, 시간 간격기반 모델은 미디어 간의 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 페트리

네트 등은 객체의 동적인 행위를 구분하지 못한다. 예를 들어, 2개의 객체가 서로 동기화 정보를 교환할 때, 시나리오의 다음을 구별하지 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 사례로 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 상호 작용이 부분적으로 가능하다. 여섯째, Perez-Luque의 멀티미디어 동기화 모델을 위한 참조 기준[17]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. 예를 들어 정량적 정보는 기본적인 시간 단위에 의해 표현되는 시간 정보이고, 정성적 정보는 시점들이나 시간 간격들 간에 설정되는 시간 관계 등을 나타낸다. 그리고, 시간 표현의 유형은 날짜 기법(dating), 시간 제한(constraint propagation), 지속 기간(duration) 기법이 있다. 이런 시간 모델의 구성요소의 조합에 의해 5가지 유형이 나오는데, 그 중 정량적 날짜형은 직접적인 시간 관계와 비결정적인 시나리오를 표현할 수가 없는 단점이 있고, 정성적 시간 간격은 정성적 정보를 이용하여 기술하고 필요에 따라 정량적 정보를 이용한다. 본 연구는 정성적 시점으로, 정량적 정보와 정성적 정보를 함께 갖는 특징이 있으며, 모델은 사건의 전개 순서와 제한시간에 대한 정보를 가지고 있으며, 정성적 정보로 인해 비결정적 시나리오 표현이 가능하다. 끝으로, 태그 분석 능력이다. 기존의 연구들이 멀티미디어 데이터베이스 시스템에 저장된 문서를 위한 모델링인데 반하여 본 연구는 XML 형태로 된 SMIL 문서에 대한 모델링을 지원한다.

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.

5. 결론 및 향후 연구과제

현재 인터넷을 통해 문서의 교환이 활발히 이루어짐

에 따라 마크업 언어에 관한 관심이 높아졌으며 최근에 멀티미디어 문서 교환을 위해 SMIL 문서가 제정되었다. 그러나, 아직까지 SMIL 문서를 모델링하는 도구(tool)가 개발되지 않아 SMIL 문서 개발 시에 복잡한 멀티미디어간의 시간적인 동기화 구조를 일관성있게 표현하고 문서화하기 어렵다. 따라서, 본 연구에서는 UML 사용사례, 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 사용사례 다이어그램과 순서 다이어그램에 적절하게 사상시키는 규칙과 알고리즘을 제안하였으며, SMIL 문서를 위한 객체지향 데이터베이스 스키마 형태인 형식 모델과 다이어그램 모델링 함수를 제안하였다. 그리고, 본 연구를 평가하기 위해 기존의 멀티미디어 데이터를 위한 대표적인 모델링 방법들과 비교, 분석해보았다.

본 연구의 사상 규칙과 알고리즘을 이용하면 SMIL 문서 개발시 UML 사용사례, 순서, 협력, 그리고, 클래스 다이어그램을 이용하여 객체지향적으로 모델링 해주므로 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.

향후 연구 과제로는 UML 사용사례, 순서, 협력, 그리고, 클래스 다이어그램으로부터 SMIL 문서를 자동 생성하는 역변환 알고리즘을 제안하는 것이며, 각종 OODB 저장을 위한 스키마 생성과 검색을 위한 질의어도 개발해야 할 것이다.

참고 문헌

- [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-1998-0615/>.
- [4] Natanya Pitts-Moultis, Cheryl Kirk, "XML black book," The Coriolis Group, Inc., 1999.
- [5] James rumbaugh, Ivar Jacobson, Grady Booch, "The unified modeling language reference manual," Addison-Wesley, 1999.
- [6] Eric van Herwijnen, "Practical SGML," Kluwer

Academic Publishers, 1994.

- [7] 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.
- [8] 하얀, 황용주, 김용성, "SGML DTD로부터 UML 클래스 다이어그램으로의 사상 알고리즘", 정보과학회 논문지, 제 26권, 제4호, pp. 508-520, 1999. 4.
- [9] 채원석, 하얀, 김용성, "UML 클래스 다이어그램을 이용한 XML 문서 구조 다이어그램", 정보처리학회 논문지, 제6 권, 제10호, pp. 2670-2679, 1999. 10.
- [10] 장원호, 임혜정, 박인호, 강현석, "HOMT: HyTime DTD 설계를 지원하기 위한 XOMT의 확장", 정보처리학회 논문 지, 제5권, 제9호, pp. 2213-2223, 1998. 9.
- [11] Lloyd Rutledge, John F. Buford, John L. Rutledge, Modeling Techniques for HyTime, 1996, <http://www.sil.org/sgml/rutledgeModelingHy.htm>.
- [12] 김두현, 김지용, 황승구, "차세대 웹 상에서의 멀티미디어", 정보처리학회지, 제6권, 제3호, pp. 62-71, 1999. 5.
- [13] Bruce Power Douglass, "Real-Time UML Developing Efficient Objects for Embedded Systems," Addison-Wesley, 1998.
- [14] Grady Booh, James Rumbaugh, Ivar Jacobson, "The Unified Modeling Language User Guide," Addison-Wesley, 1999.
- [15] 류시원, 김창룡, 차광호, 정진호, "멀티미디어 데이터베이스 시스템에서의 멀티미디어 데이터 모델링에 대한 고찰", 1997. <http://www.multimedia.or.kr/newtech/NT14/nt2.htm>.
- [16] 하수철, 성해경, "게임공간의 분류와 시나리오의 시간 및 공간 동기화 표현법", 정보처리학회 논문지, 제6권, 제10호, pp. 2630-2641, 1999. 10.
- [17] 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.
- [18] Craig Larman, "Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design," Prentice Hall PTR, 1998.
- [19] E. Akpotsui, V. Quint, C. Roisin. "Type Modelling for Document Transformation in Structured Edition Systems," August 1994, <http://www.oasis-open.org/cover/>.
- [20] James Rumbaugh, Ivar Jacobson, Grady Booch, "The Unified Modeling Language Reference Manual," Addison-Wesley, 1999.
- [21] V. Christophides, S. Abiteboul, S. Cluet, M. Scholl, "From Structured Documents to Novel Query Facilities," In Proc. ACM SIGMOD Intl. Conf. Management of Data, pp.313-324, May 1994.



채 원 석

1984년 원광대학교 전자공학(공학사). 1986년 인하대학교 전자계산학과(이학석사). 1998년 전북대학교 전산통계학과 박사과정 수료. 1989년 ~ 현재 원광보건대학 컴퓨터응용개발과 교수. 관심분야는 멀티미디어 응용, 영상 처리, 전자 도서

관, 정보 검색 등



하 안

1992년 덕성여자대학교 전산학과 졸업(학사). 1994년 8월 이화여자대학교 교육대학원 전자계산교육 전공 졸업(석사). 2000년 2월 전북대학교 대학원 전산통계학과 졸업(이학박사). 관심분야는 SGML/XML, 정보 검색, 객체지향 모델링, 지

능형 교육시스템 등

김 용 성

정보과학회논문지 : 소프트웨어 및 응용 제 27 권 제 3 호 참조