

타임라인 정보로부터 SMIL 구조 정보를 생성하는 알고리즘의 개발

김태현^o 김경일 이규철
충남대학교 컴퓨터공학과
(thkim, kyung1, kclee)@ce.cnu.ac.kr

SMIL Structure Information Generation Algorithm based on Timeline Information

Tae-Hyun Kim^o Kyung -Il Kim Kyu -Chul Lee
Dept. of Computer Engineering, Chungnam National University

요 약

인터넷 미디어의 멀티미디어화 추세에 따라 W3C의 동기화 멀티미디어 워킹그룹(SYMM) [1]에서는 1998년 웹상에서의 멀티미디어 표현을 위한 XML 기반의 동기화 언어인 SMIL을 발표하였다. SMIL [2]은 순차관계(<seq>)와 병렬관계(<par>) 및 이들의 중첩관계를 사용하여 미디어간의 시간적 동기화를 설정한다. 이에 따라 기존의 SMIL 저작 도구들은 구조 편집기를 통하여 엘리먼트간의 중첩관계를 직접 편집함으로써 멀티미디어 시나리오의 시간적 동기화 정보를 기술하도록 하고 있다. 하지만 구조 편집기는 SMIL에 중심적인 편집 방법으로써 엘리먼트간의 중첩이 심화될수록 사용자에게 혼돈을 가져온다.

본 논문은 사용자 중심적인 시간적 동기화 정보 기술방법을 제공하기 위한 연구에서 출발하였다. 즉, 시간 축을 기준으로 미디어간의 시간정보를 기술하는 타임라인 편집기를 제공함으로써 사용자는 보다 직관적으로 프리젠테이션 시나리오를 작성할 수 있다. 이때, 타임라인 정보는 SMIL 문서의 구조 정보와는 상이한 데이터모델을 사용하므로 이를 SMIL 구조 정보로 자동 변환하기 위한 변환과정이 필요하게 된다.

1. 서론

SMIL은 웹상에서 TV와 유사한 형태의 멀티미디어 프리젠테이션을 쉽게 작성할 수 있도록 하기 위한 W3C의 표준 언어이다. 하지만 XML [3] 기반의 SMIL 문서는 반-구조적(semi-structured)인 문서 구조를 가지기 때문에 사용자가 단순한 텍스트 편집기를 사용하여 SMIL 멀티미디어 프리젠테이션을 작성하는 것은 쉽지 않은 일이다. 때문에 SMIL이 발표되면서 다양한 형태의 GUI를 사용한 SMIL 멀티미디어 문서 저작 도구들이 발표되었다[4, 5].

이러한 저작 도구들은 SMIL 문법 중심의 사용자 인터페이스를 제공하여 여전히 사용하기 어려운 점들을 가지고 있다. 즉, 대부분의 저작 도구들이 사용하는 구조 편집기(structure editor)는 SMIL이 가지는 문서구조를 직접 편집하도록 함으로써 엘리먼트간의 중첩관계가 심화될수록 편집에 어려움이 따른다.

본 논문은 사용자에게 보다 간편한 형태의 사용자 인터페이스를 제공하기 위한 연구이다. 이를 위해 단지 시간 축을 기준으로 하는 타임라인 에디터(timeline editor)만을 사용하여 멀티미디어 시나리오를 작성할 수 있도록 하기 위해 타임라인 정보(timeline information)

로부터 SMIL 구조 정보를 자동으로 생성하는 알고리즘을 개발하였다.

2. SMIL 구조 정보 생성

타임라인 정보는 시간에 대해 서로 독립적으로 표현되어지는 트랙(track)들의 집합으로 구성되며, 각 트랙은 순차적으로 표현되어지는 미디어들을 포함한다. 트랙은 미디어 객체들의 집합으로써, 다른 트랙에 속한 미디어 객체들에 대해 시간적인 독립성을 가진다.

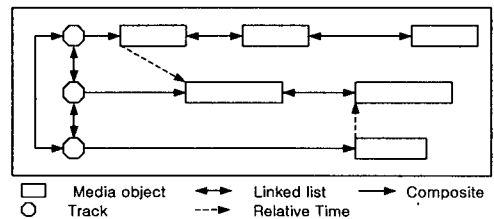


그림 1. 타임라인 정보를 구성하는 자료구조

그림 1은 타임라인 정보를 이루는 데이터들의 관계

를 표현한 것이다. 미디어 객체들은 트랙단위의 리스트 형태로 유지된다. 또한, 독립적으로 표현되어지는 각기 다른 트랙에 존재하는 미디어간의 연관성을 지정하기 위해 사용자는 상대시간 관계(Relative Time relationship)를 지정할 수 있다.

본 논문에서는 미디어들로 구성된 리스트들의 집합과 같은 형태로 처리되는 타임라인 정보를 반-구조적 특성을 가지는 SMIL 구조 정보로 변환하기 위한 중간 결과물로서 이진트리를 생성한다. 이때 변환을 쉽게 하기 위해 임시객체인 지연객체(delay object)를 사용하며, 지연객체를 포함하는 이진트리로부터 DOM [6] 트리를 생성함으로써 완전한 SMIL 문서를 생성하게 된다.

2.1 지연객체 (Delay object)

멀티미디어 표현에 있어서 두개의 미디어 객체간의 표현 시간관계에 대한 기본 이론은 Allen [6]에 의해 정의되어진 13 가지의 관계(before, meets, during, ...)이다. 하지만 이러한 관계는 두개 이상의 미디어간의 시간적 관계 표현에는 어려움이 있으며, 13 가지의 관계는 컴퓨터를 사용한 처리 알고리즘 개발에 어려움이 있다.

Allen의 13 가지 관계는 그 논리적 의미를 유지하면서 순차관계(<seq>)와 병렬관계(<par>) 및 이들의 중첩 관계를 사용하여 표현되어질 수 있다[8]. 이때 사용되는 개념이 지연객체 이다.

지연객체는 <seq>와 <par>를 기반으로 하는 시간적 동기화 표현을 단순하게 만들어주는 더미객체(dummy object) 이다. 본 논문에서는 타임라인 정보로부터 SMIL 구조 정보를 생성하기 위한 중간 단계로 지연객체를 적용한 이진트리를 구성하게 된다.

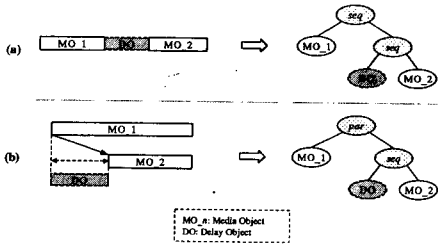


그림 2. 지연객체의 사용 예

그림 2는 지연객체를 사용하여 타임라인 정보로부터 이진트리를 생성하는 예를 보여준다. 이때, 그림에서와 같이 지연객체는 항상 <seq> 노드의 자식노드가 되며, 지연객체의 형제노드는 미디어객체 혹은 <seq> 나 <par>을 최상위 노드로 취하는 서브트리가 될 수 있다.

2.2 변환 알고리즘

타임라인 편집기를 통해 기술되어진 시나리오는 다음의 다섯 단계를 거쳐 완전한 SMIL 문서로 생성되어진다.

[1 단계] 상대시간 관계의 미디어객체들을 지연객체를 포함한 <par> 블록으로 변환(그림 3).

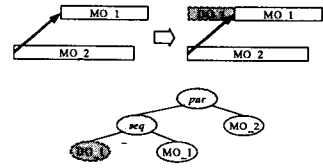


그림 3. <par>을 사용한 상대시간 관계 변환.

타임라인 정보에서 가장 밀접한 관계를 가진 객체들은 사용자에 의해 상대시간 관계로 설정되어진 미디어들이다. 따라서 이러한 관계에 있는 미디어들이 가장 먼저 처리되어야 한다. 상대시간 관계는 <par> 블록으로 변환되어지며, 변환된 <par> 블록은 하나의 객체로 취급할 수 있다.

그림 3에서 미디어객체 "MO_1"과 "MO_2"는 "DO_1"의 시간만큼 격차를 두고 같은 시간대에 표현되어지는 것을 볼 수 있다. 이러한 관계는 그림 하단의 이진트리와 같이 <par> 블록으로 변환되어진다.

[2 단계] 각 트랙의 객체들을 <seq> 블록으로 변환(그림 4).

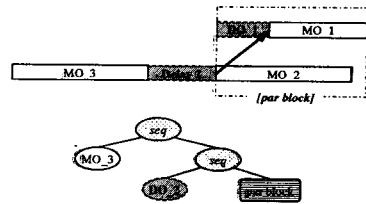


그림 4. 각 트랙의 객체들을 <seq> 블록으로 변환.

[1 단계]의 과정을 통해 트랙간의 시간적 동기화를 필요로 하는 상대시간 관계는 하나의 <par> 블록으로 처리되어진다. 따라서, 하나의 트랙에 존재하는 미디어들은 다른 트랙의 미디어에 독립적으로 표현되어질 수 있으며 이들은 순차적인 관계를 갖는다. 변환의 두 번째 단계에서는 각 트랙의 순차적인 미디어들을 <seq> 블록으로 묶는다.

그림 4에서 "par block"은 상대시간 관계를 가지는 미디어들의 집합체로써 하나의 객체로 처리되어질 수 있으므로 미디어객체 "MO_3"과 "par block"은 그림 2-(a)와 같은 방법으로 변환되어질 수 있다.

[3 단계] 남겨진 모든 객체들을 <par> 블록으로 변환(그림 5).

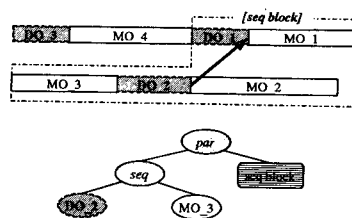


그림 5. 남겨진 객체들을 <par> 블록으로 변환.

[2 단계]의 과정을 마치면 타임라인 정보는 <seq> 블록으로 묶여진 몇 개의 이진트리들로 변환되어지며, 변환되어진 이진트리들은 동시에 표현되어질 수 있으므로 <par> 블록으로 묶여진다. 이 단계를 통해 타임라인 정보는 변환의 중간 단계인 자연객체를 포함하는 하나의 이진트리로 변환되어진다.

그림 5는 미디어객체 “MO_4”와 <seq> 블록 “seq block”을 <par> 블록으로 변환하는 것을 보여준다.

[4단계] 자연객체 제거 및 자연객체를 포함한 블록 처리: 자연객체를 SMIL 엘리먼트의 동기화 어트리뷰트로 변환(그림 6).

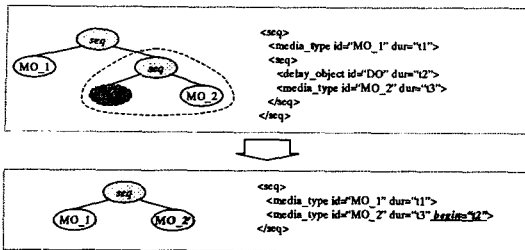


그림 6. 자연객체의 변환

SMIL 구조정보 생성 알고리즘의 네 번째 단계는 더 미객체인 자연객체를 변환하는 것이다. 자연객체는 데이터의 변환을 쉽게 하기 위한 임시객체 이므로 모든 자연객체들은 관련된 엘리먼트의 동기화 어트리뷰트로 변환되어져야 한다.

그림 6은 자연객체 “DO”가 형제 노드인 미디어객체 “MO_2”의 동기화 어트리뷰트 “begin”으로 변환되어지는 것을 보여준다.

[5 단계] 변환된 이진트리의 최적화를 통한 DOM 트리 생성(그림 7).

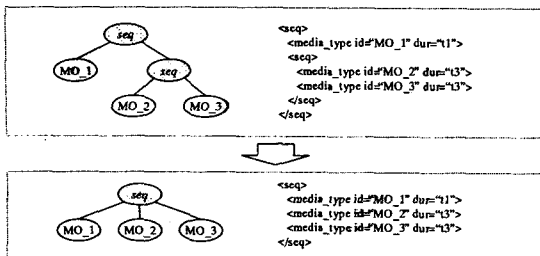


그림 7. 최적화를 통한 DOM 트리를 생성.

앞선 4 단계의 과정을 거쳐 생성되어진 이진트리는 불필요한 <seq>와 <par> 엘리먼트들을 다수 포함한다. 이 단계에서는 이진트리에 포함되어 있는 불필요한 <seq>와 <par> 엘리먼트를 제거함으로써 최적화된 DOM 트리를 생성한다. 이러한 과정은 보다 적은 수의 엘리먼트를 가지는 SMIL 문서를 생성하기 위해 필요하다.

그림 7과 같이 미디어객체 “MO_1”, “MO_2”, “MO_3”의 순차관계를 표현하기 위한 이진트리는 두 개의 <seq> 엘리먼트를 필요로 하지만, 그림의 하단과 같이 변환함으로써 하나의 <seq> 엘리먼트만을 사용하여 표현하는 것이 가능해진다.

3. 결론

SMIL은 기존의 서로 독립적인 미디어 객체들을 통합하여 TV와 같은 형태의 멀티미디어 프리젠테이션을 웹상에서 표현하는 것을 가능하게 한다. 또한, SMIL의 미디어 객체들에 대한 시간적 동기화 방법은 순차관계와 병렬관계 및 이들의 중첩관계만을 사용하여 기술하도록 하는 단순한 메커니즘을 제공함으로써 컴퓨터를 사용한 처리가 용이하다. 하지만 순차관계와 병렬관계의 중첩이 심화될수록 SMIL 문서의 작성 및 수정에 어려움이 있다. 우리는 기존의 SMIL 저작 도구들에 대한 조사를 통해 기존 시스템들이 제공하는 구조 편집기가 이러한 문제점을 가지고 있다는 것을 알았다.

본 논문에서는 타임라인 정보의 기술만으로 SMIL 구조 정보를 생성하는 알고리즘을 개발하였다. 사용자는 구조 편집기가 아닌 시간 축을 기준으로 하는 타임라인 편집기에 각 미디어들의 표현 시간을 배치함으로써 SMIL 문서의 시간적인 동기화 정보를 기술할 수 있다. 즉, SMIL 문서의 구조 정보 자체를 편집하는 것이 다분히 SMIL 지향적인 방법이라고 한다면 타임라인 편집은 사용자 지향적인 방법이라고 할 수 있다.

4. 참고 문헌

- [1] W3C Synchronized Multimedia Working Group, <http://www.w3.org/AudioVideo>
- [2] W3C SMIL Specification, <http://www.w3.org/TR/REC-smil>
- [3] W3C XML specification, <http://www.w3.org/TR/REC-xml>
- [4] D.C.A Bulterman, L. Hardman, J. Jansen, K.S. Mullender and L. Rutledge, “GRiNS: A Graphical Interface for Creating and Playing SMIL Documents,” Proc. Seventh International World Wide Web Conference (WWW7), April 1998.
- [5] Sausage Software Home page, <http://www.sausage.com/supertoolz/toolz/stsmil.html>
- [6] W3C DOM Level 1 Specification, <http://www.w3.org/TR/REC-DOM-Level-1>
- [7] J. F. Allen, “Maintaining Knowledge about Temporal Intervals,” Communications of the ACM, Vol.26, No.11, Nov. 1983, pp.823-843.
- [8] Kyuchul Lee, Woosaeng Kim, Jinho Hahm, Sukho Lee, “Temporal Specification and Synchronization for Multimedia Database Queries,” Proc. Of the International Symposium in Next Generation Database Systems and Their Applications, Sep.1993, pp.198-204.