

---

# 확장된 시간 구간 모델을 이용한 SMIL2.0 문서의 시간 관계 검증

김경덕\*

Verification of Temporal Relations on SMIL 2.0 Document  
using an Extended Temporal Interval Model

Kyungdeok Kim\*

## 요 약

본 논문에서는 다양한 상호작용을 지원하는 SMIL 2.0 멀티미디어 문서를 확장된 시간 구간 모델에 기반하여 문서의 시간 관계를 검증하는 방법을 제안한다. SMIL 2.0 문서는 기존 SMIL 문서보다 비동기적인 시간 관계의 기술 방법을 추가되어 객체간 다양한 상호작용 관계를 제공하지만, 이러한 상호작용 관계에 의한 시간 관계에서 모순이 발생할 가능성이 높다. 그러므로 제안한 검증 방법은 SMIL 2.0 문서를 기존 시간 구간 모델을 확장하여 상호작용 관계를 함수적 관계로 표현하고 객체간 상호작용 관계와 시간 구간을 이용하여 SMIL 문서를 검증한다. 제안한 검증 방법의 적용 예로는 다자간 상호작용 콘텐츠, 온라인 교육 등이다.

## ABSTRACT

This paper suggests a method for verification of temporal relations on SMIL 2.0 document using an extended temporal interval model. The SMIL 2.0 document has more asynchronous temporal relations and supports more interactive relations between objects than traditional SMIL document. But it is higher to produce temporal inconsistency by various interactive relations between objects. So the suggested verification method expresses SMIL 2.0 document as functional relations based on an extended temporal interval model and verifies SMIL 2.0 document using interactive relations and temporal intervals between objects. Applications to the verification method are as follows; multiuser interactive contents, online-learning, etc.

## 키워드

SMIL2.0, 시간 구간, 시간 관계 검증, 함수적 표현

## I. 서 론

상호작용 가능한 멀티미디어 문서는 다양한 분야에서 현재 널리 이용되고 있다. 특히 상호작용 관계를 쉽게 표현할 수 있는 SMIL(Synchronized Multimedia Integration Language) 문서는 데스크톱 및 모바일 장치에

서 그 활용도 점점 높아지고 있다. SMIL 문서는 웹상에서 분산된 다양한 미디어 객체들의 복합체와 사용자간 상호작용을 표현하고, XML기반의 문서로 사용자의 작성 또한 편리하다[1, 2]. SMIL 2.0 문서는 기존 SMIL 1.0 사양에 비동기적 시간관계 사양을 추가하여 더욱 융통성 있는 시간 관계 모델을 지원한다. 하지만 다양한 시간

관계 기술이 가능해짐에 따라 SMIL 문서를 저작할 때 모순된 시간 관계의 기술 가능성도 더욱 높아진다. 이러한 모순된 시간 관계를 가진 SMIL 문서는 사용자의 저작 의도와 다른 프레젠테이션을 제공할 수 있다. 이러한 시간적 관계의 모순을 해결하기 위한 기존 연구들은 주로 SMIL 1.0 문서의 시간 관계 모순을 해결하기 위한 연구들이며, 복잡한 시간 관계를 제공하는 SMIL 2.0 문서의 모순 검증, 진단, 스케줄링 방법에 대한 연구는 미흡한 편이다.

기존 멀티미디어 문서의 시간 관계를 검증하는 방법으로는 페트리넷(petri-net) 기반 모델, 페트리넷의 응용 모델, LOTOS 등의 주로 형식적인 모델을 이용하고 있다. 기존 페트리넷 기반 모델은 논리적이고 양적인 시간 관계를 표현할 수 있지만 작성 및 구현이 어렵다[3, 4, 5, 6, 7]. 페트리넷 응용 모델인 OCPN[4]은 시간 구간을 기반으로 하여 객체들 간의 시간 관계를 명확히 나타낼 수 없으며, 특히 이벤트 기반의 객체 관계는 더욱 나타내기 어렵다[5]. 특히 객체의 배타적인 재생 관계를 표현하지 못한다. 동적 페트리넷 DPN[6]은 반복적이고 상호작용적인 모델을 기술하기 위하여 스크립트를 기존 페트리넷 모델의 플레이스(place)에 연관하여 표현하고 있으나, 객체간 상호작용 관계가 복잡할수록 문제의 기술과 분석이 쉽지 않으며 스크립트내 공유 변수의 사용으로 해석이 어렵다. LOTOS는 상호작용 기술을 위하여 확장 모델인 RT-LOTOS를 주로 이용하며, 주어진 문서내 각 객체들의 관계를 도달(reach) 가능 그래프로 변환하고 비도달(reachless) 노드의 탐색을 통하여 모순을 검증한다[8]. 하지만 그래프 크기가 문제 규모에 비하여 상대적으로 크고 주어진 그래프로부터 비도달 노드의 탐색이 쉽지 않다. 그러므로 본 논문에서는 확장된 시간 구간 모델을 사용하여 SMIL2.0 문서를 확장된 시간 구간 모델에 기반을 둔 구조화된 함수 표현으로 변환하여 시간 관계 모순을 검증하는 방법을 제안한다.

본 논문의 구성은 다음과 같다. 제 2장에서는 확장된 시간 구간 모델과 SMIL 문서의 함수 표현으로 변환하는 방법에 대하여 설명한다. 제 3장에서는 SMIL 문서 검증 방법을 설명하고, 제 4장에서는 구현 및 검증 사례를 보이고, 마지막 제 5장에서는 결론 및 향후 연구 방향을 기술한다.

## II. 시간 구간 모델과 SMIL문서의 표현

### 2.1 기존 시간 구간 모델

기존 시간 구간 모델은 Allen의 시간 표현 방법을 응용하여 객체의 재생 시간 구간을 표현하고 있으나, 객체간 또는 사용자의 상호작용으로 발생하는 관계는 표현하기 어렵다. 그러므로 본 논문에서는 기존 시간 구간 모델에 비결정적으로 발생할 수 있는 객체간 상호작용 관계를 추가하여 기존 시간 구간 모델을 확장하여 사용한다. 본 장에서는 먼저 기존 시간 구간 모델[8]을 설명한 후, 상호작용 관계를 위하여 2.2절에서 확장된 모델[9]을 설명한다.

시간 구간 모델에서 객체  $o$ 의 시간 구간은 식 (1)과 같이 정의한다. 여기서,  $t$ 는 시간이고  $o_s$ 와  $o_e$ 는 시간 구간의 시작과 종료를 나타낸다.

$$o = \{t \mid o_s \leq t \leq o_e\} = (o_s, o_e) \quad \dots(1)$$

시간 구간 모델에서 SMIL 문서에 포함된 다양한 멀티미디어 객체들의 재생 시간은 시간 구간 연산자의 계산 결과이며, 이 연산자의 입력과 출력은 각 객체들의 시간 구간을 나타낸다. 즉, 시간 구간 연산  $ti_3 = op(ti_1, ti_2)$ 는  $op$ 를 시간 구간 연산자로 하고,  $ti_1$ 과  $ti_2$ 는  $op$  연산자에 입력으로 주어지는 객체의 시간 구간이며,  $ti_3$ 는 출력되는 시간 구간이다. 기존 시간 구간 모델에서 사용하는 연산자로는  $seq()$ ,  $follow()$ ,  $par\_begin()$ ,  $par\_end()$ ,  $par\_min()$ ,  $par\_max()$ ,  $equal()$ ,  $alt()$ ,  $loop()$ 가 있으며, 연산자  $\mapsto$ 는 시간 구간의 출력을 나타내며,  $a$ ,  $b$ 는 피연산자인 객체이고,  $a_s$ ,  $a_e$ ,  $b_s$ ,  $b_e$ 는 각 객체의 시작 및 종료 시각을 의미하며, 각 연산의 결과는 다음과 같다.

$$\begin{aligned} seq(a,b) &\mapsto (a_s, b_e), follow(a,b) \mapsto (a_s, b_e), \\ par\_begin(a,b) &\mapsto (a_s, a_e), \\ par\_end(a,b) &\mapsto (a_s, a_e), \\ par\_min(a,b) &\mapsto (a_s, \min(a_s, a_e)), \\ par\_max(a,b) &\mapsto (a_s, \max(a_e, a_e)), \\ equal(a,b) &\mapsto (a_s, a_e), ident(a,b) \mapsto (a_s, b_e), \\ alt((a_1, b_1), \dots, (a_n, b_n)) &\mapsto (a_{is}, b_{ie}), \\ loop(a, \infty) &\mapsto (a_s, \infty). \end{aligned}$$

각 연산자의 의미는 다음과 같다. 먼저,  $seq(a,b)$  연산은 객체  $a$ 의 재생 종료 시간이 객체  $b$ 의 재생 시작을 정의하고, 객체  $b$ 의 재생 종료 시간이 전체 연산의 종료를 의미한다.  $follow(a,b)$  연산은 객체  $a$ 의 재생 종료 시간은 객체  $b$

의 재생 시작으로 정의되고, 객체 b의 재생 종료는 전체 연산의 종료를 의미한다.  $par\_min(a,b)$  연산은 객체 a의 재생과 객체 b의 재생을 동시에 시작하고 둘 중 먼저 재생이 종료하는 객체가 나머지 객체의 재생 종료와 전체 연산의 종료를 의미한다.  $par\_max(a,b)$  연산은 객체 a의 재생과 객체 b의 재생을 동시에 시작하고 둘 중 재생 시간이 큰 객체의 종료 시간이 나머지 객체의 재생 종료를 정의하고 전체 연산의 종료를 의미한다.  $equal(a,b)$  연산자는 객체 b의 재생 시작 시간 및 종료 시간이 객체 a의 시작과 종료에 종속되며, 객체 a의 재생 시간이 전체 연산의 시간 구간을 의미한다.  $ident(a,b)$  연산자는 객체 a의 재생 시작이 객체 b의 재생을 시작하게 하고 객체 b의 재생 종료는 객체 a의 종료를 정의하며, 전체 연산의 시간 구간은 객체 a의 재생 시작이 전체 연산의 시작 시간되고 객체 b의 종료 시간이 전체 연산의 종료 시간으로 정의된다.  $alt((a1,b1),..., (an,bn))$  연산자는  $a1, \dots, an$ 의 객체가 병렬로 재생 중에 선택된 객체  $a_i$ 와 그에 종속된 객체인  $b_i$ 의 재생 종료로 전체 연산 시간이 정의된다.  $loop(a, \infty)$  연산자는 객체 a의 재생을 무한 반복하고 외부 이벤트에 의하여 전체 연산의 종료가 정의된다. 기존 SMIL1.0 문서는 객체간 단순한 상호작용 행위를 기술함으로 기존 시간 구간 모델로도 적용이 용이하나, SMIL2.0 문서의 경우는 재생 객체에 시작과 종료 속성이 추가되어 적용이 어렵다. 그러므로 본 논문에서는 SMIL 2.0문서를 시간 구간 모델을 적용하기 위하여 기존 시간 구간 모델의 표현 방법을 확장하여 문서 검증에 이용한다.

## 2.2 확장된 시간 구간 모델과 SMIL문서의 표현

확장된 시간 구간 모델에서는 시간 구간을 가지는 객체로서 재생 객체 외에 내부 및 외부 이벤트를 피연산자 객체로 확장하여, SMIL 2.0문서의 다양한 상호작용 관계를 표현한다. SMIL 2.0 문서는 기존 시간구간 연산자 중에서  $seq()$ ,  $par\_end()$ ,  $par\_begin()$ ,  $par\_max()$ ,  $loop()$ 와 새로운  $excl()$  연산자를 추가하여 시간 구간을 표현한다. 그리고  $loop()$  연산자는 객체의 반복 재생 속성을 나타내는 연산자로 피연산자로서 반복 재생할 객체와 반복 재생 횟수나 반복 재생 시간을 피연산자로 정의한다.  $excl()$  연산자는 기존 시간 구간 모델의  $alt()$  연산자와 유사하지만, 시작 속성에 따라 활성화된 각 객체는 자신의 종료 속성이거나 다른 피연산자 객체가 활성화될 때까지 재생된다. 즉  $excl()$  연산자는 사용자와 상호작용에 따라 배

타적인 연산을 수행하며, 피연산자인 n개의 객체중 선택된 하나의 객체만 재생한다. SMIL 2.0 문서를 확장된 시간 구간 표현으로의 변환은 문서의 태그 및 속성을 이용하여 확장 정의된 시간 구간 연산자들의 조합인 함수적 관계로 변환한다. 변환으로 표현되는 함수적 관계는 트리로 표현할 수 있어 문서의 검증이 용이하다. SMIL문서를 나타내는 DOM(Document Object Model) 트리의 각 노드는 문서의 내용에 따라 다수의 부 노드를 가지며 각 노드가 포함하는 속성에 의한 상호작용 관계를 분석하기 어렵다. 그러나 DOM 트리를 확장된 시간 구간으로 표현하면 상호작용에 따른 시간 관계 분석이 용이하다. 여기서, 변환된 표현은 이진 트리 구조로 나타낼 수 있으며, 단  $excl()$  연산자만 두개 이상의 자식 노드로 표현될 수 있다.  $excl()$  연산자 이외의 모든 노드의 자식 노드는 연산자 객체, 이벤트 객체, 재생 객체를 자식 노드로 가질 수 있으며,  $excl()$  연산자 노드는 하나 이상의 연산자 객체를 자식 노드로 가진다. 모든 이벤트 객체 노드는 항상 좌측 노드에 위치하며, 형제 노드의 하위 계층 노드들의 재생 및 종료를 표현한다. 확장된 시간 구간 모델에서 객체의 시작과 종료 속성을 나타내는 이벤트 객체는 식 (2)와 같은 시간 구간으로 정의한다. 이벤트 객체의 시간 구간은 시작과 종료는 동일하므로 시간 구간은  $\emptyset$ 이다.

$$e = (e_s, e_e) = \{t | e_s = t = e_e\} \quad \dots(2)$$

그림 1은 SMIL 2.0 문서의 시간 구간 표현을 나타낸다. 여기서, (a)는 사용된 SMIL 문서이며, (c)는 시간 구간 트리로 루프 A, B, C는 (b)의 SMIL 문서 DOM 트리에서 속성을 가진 노드가 확장된 시간 구간 모델로 변환된 상태를 나타낸다. 또한, (c)의 시간 구간 트리의 노드에서 이름 확장자가  $begin$ ,  $end$ ,  $activate$ 인 객체는 이벤트 객체로서 객체 재생의 시작 및 종료와 활성화를 나타내고, 각 노드의 좌 노드에 위치하여 형제 노드의 하위 계층 노드들의 활성화 및 종료를 제어하기 위한 이벤트임을 나타낸다. (b)의 DOM 트리에서 좌측  $\langle par \rangle$  노드는 속성으로  $endsync=a1$ 을 가지며, 이것은 시간 구간 트리에서 루프 A로 변환되며, 루프 B와 C는 각각 DOM 트리의 v1 노드와 우측  $\langle par \rangle$  노드의 시간 구간 트리로 변환된다.

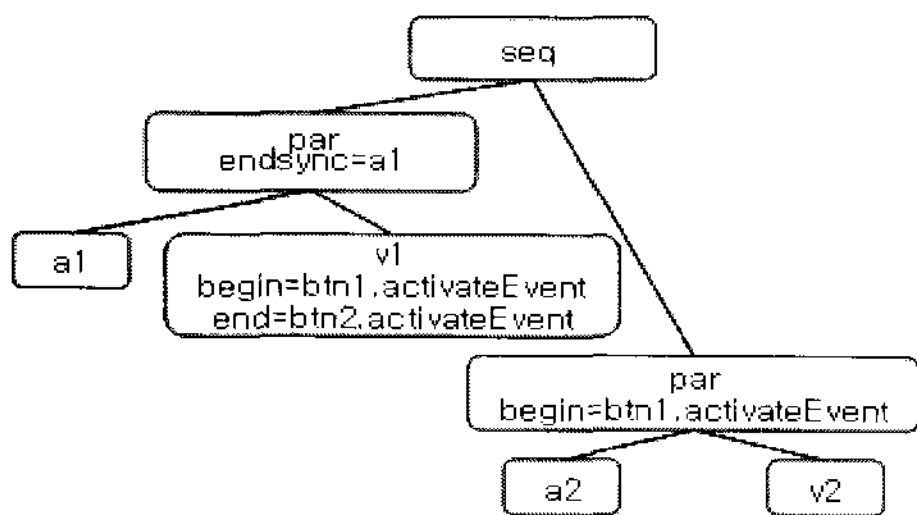
변환시 시작 속성은  $\langle par\_begin \rangle$  태그로, 종료 속성은  $\langle par\_end \rangle$  태그로 변환되며, 속성의 값은 이벤트 객체로서 각 태그의 좌측 자식 노드에 배정된다. 그리고  $\langle par \rangle$  태그는  $\langle par\_max \rangle$  태그로 변환되며, 그림 1의 시간 구간

트리(c)의 함수적 표현은 다음과 같다.

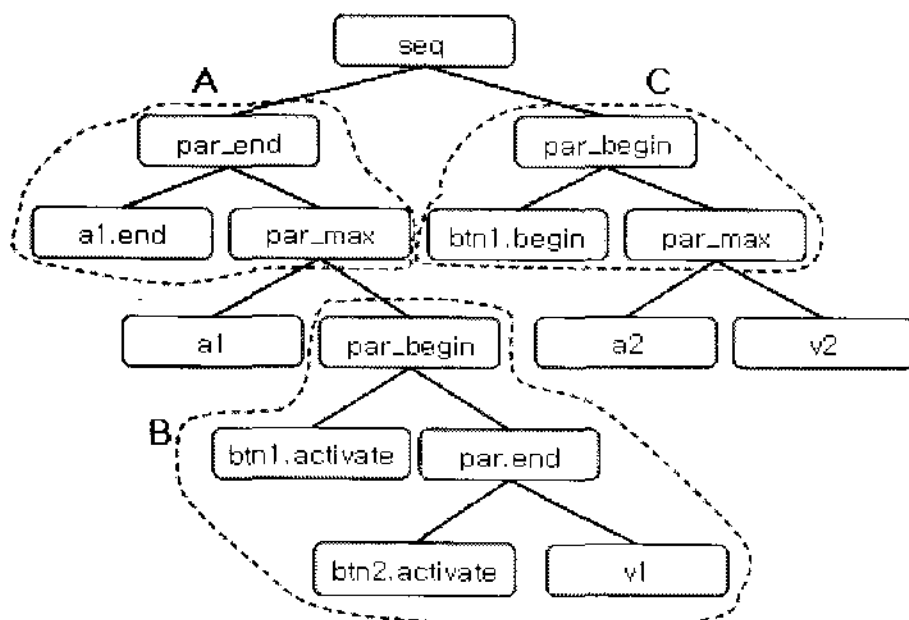
```
seq(par_end(a1.end,
    par_max(a1,
        par_begin(btn1.activate,
            par_end(btn2.activate,
                v1),
            par_begin(btn1.begin,
                par_max(a2,
                    v2)
            )
        )
    )
).
```

```
<seq>
<par endsync="a1">
  <audio id="a1" src="..." />
  <video id="v1" src="..."
    begin="btn1.activateEvent"
    end="btn2.activateEvent" />
</par>
<par begin="btn1.activateEvent">
  <audio id="a2" src="..." />
  <video id="v2" src="..." />
</par>
</seq>
```

(a) SMIL 문서



(b) DOM 트리



(c) 시간 구간 트리

그림 1. SMIL 2.0 문서의 시간 구간 표현  
Fig. 1 Expression of an Extended Temporal Interval on a SMIL 2.0 Document

다음 함수 Transformation\_of\_SMIL(tree)는 SMIL 문서 DOM 트리의 선순회(pre-order) 탐색에서 각 노드를 시간 구간 트리로 변환하는 함수이다. 함수는 태그와 각 태그의 속성에 따라 시간구간 트리로 변환한다.

```
tree function Transformation_of_SMIL(tree NT)
{ // input : tree
  // output : tree for functional expression
  switch (NT's tag type) {
    <seq> : the node is not changed;
    if (node's sibling has more than 3)
      T ← call_transform(NT, <seq>);
    break;
    <object begin ...> :
    if (the node has repeat attribute) then
      it is changed to a tree T for an expression
      "par_begin(begin's attribute,
        loop(repeat's attribute,object))";
    else
      node's tag attribute is changed to a tree T
      for an expression "par_begin(begin's attribute,
        object)";
    break;
    <object endsync ...> :
    if (the node has repeat attribute) then
      it is changed to a tree T for an expression
      "par_end(endsync's attribute,
        loop(repeat's attribute,object))";
    else
      node's tag attribute is changed to a tree T for
      an expression "par_end(endsync's attribute,
        object)";
    break;
    <object begin ... endsync ...> :
    if (the node has repeat attribute) then
      it is changed to a tree T for an expression
      "par_end(endsync's attribute,
        loop(repeat's attribute,object))";
    else
      node's tag attribute is changed to a tree T for
      an expression "par_begin(begin's attribute,object)";
    break;
    <par> :
    node's tag attribute is changed to a tree T for
    an expression "par_max()";
    if (node's sibling has more than 3)
      T ← call_transform(NT, <par>); break;
    <par ... begin...>:
    node's tag attribute is changed to a tree T for
```

```

    an expression
    "par_begin(begin's attribute,par_max(...))";
    break;
<par ... endsync ...> :
    node's tag attribute is changed to a tree T for
    an expression
    "par_end(endsync's attribute,par_max(...))";
    break;
<par ... begin ... endsync ...> :
    node's tag attribute is changed to a tree T for
    an expression to an expression
    "par_begin(begin's attribute,
    par_end(endsync's attribute,...))";
    break;
<excl ...> :
    node's tag attribute is changed to a tree T for
    an expression "excl(...)";
    break;
}
mark NT and newly created node;
NT's child nodes is attached to child nodes
of T;
NT ← T;
return NT;
}

tree function call_transform(tree O, type t)
{
    Except for the leftmost child node, rest node is
    attached under type t node;
    // the newly created t node is right child
    // node of O;
    if (the number of the t node's child is more than
        3) O ← call_transform(O, t);
    return O;
}

```

### III. SMIL 문서의 검증

SMIL 문서를 확장된 시간 구간 표현인 함수 관계로 변환하면 발생 가능한 논리적 및 시간적 모순의 탐색이 용이하다. SMIL 문서에서 발생 가능한 모순으로는 내부적으로 모순된 시간 관계 설정에서 발생하는 논리적인 모순과, 외부적으로 사용자와 상호작용시 이벤트에 의한 시간 관계 설정에서 발생하는 의미적 모순 및 시간적 모순이 있다[8].

내부 객체들의 시간 관계 설정에서 발생하는 모순은

객체의 재생시간 구간과 객체간 참조 관계로부터 모순을 탐색할 수 있으나, 외부 객체로부터 시간 관계 설정이 발생하는 모순은 관련 이벤트의 발생 시점에 따라 모순 여부가 결정됨으로 탐색이 쉽지 않다. 논리적인 모순은 객체간 참조가 적절하지 못한 경우에 발생하는 것으로, 예로는 이미 재생이 끝난 비활성 객체를 참조할 때 발생한다. 의미적 모순은 부적절한 사용자 상호작용에 의하여 발생하는 것으로, 예로는 이벤트 객체의 참조로 재생을 시작하는 객체가 해당 이벤트 객체의 발생 시간이 재생 종료 시점이 지난 시점에 발생할 경우, 해당 객체의 재생은 의미 없는 기술이 될 수 있다. 시간적 모순은 객체의 재생이 진행 중일 때 발생하는 이벤트에 의하여 발생하며, 예로는 객체의 재생 동안 사용자 상호작용이나 이벤트의 참조가 필요한 객체가 그 이벤트 발생 시점이 부적절할 때 발생할 수 있다[8]. 이러한 모순은 모두 내부 및 외부 이벤트 객체를 잘못 참조하여 발생한다. 그러므로 본 논문에서는 SMIL 문서를 함수적 표현으로부터 객체들의 재생 시간 구간과 객체간 참조 관계를 이용하여 모순을 검증한다. 예로서 다음 SMIL 문장과 같이 병렬 재생 관계에서 재생되는 미디어 객체가 객체간 상호 참조 이벤트 객체에 의하여 재생될 때 재생 순서에서 모순이 발생할 수 있다. 즉, 객체 a와 v중 어느 객체가 먼저 재생될지 알 수가 없거나 재생이 되지 않을 수 있다. 순차 재생 관계에서는 이런 상호참조 이벤트 객체가 발생한다면 재생은 불가능하다.

```

<par>
  <audio id="a" src="..." begin=id(v)(begin)" />
  <video id="v" src="..." begin=id(a)(begin)" />
</par>

```

주어진 문장의 함수적 표현은  $par\_max(par\_begin(v.begin,a),par\_begin(a.begin,v))$ 이 되며, 주어진 시작 속성에 의하여 두 미디어 객체의 시작과 관련된 이벤트 객체가 상호 의존성을 가짐으로서 병렬 재생에서 모순이 발생한다. 병렬 재생 관계에서는 상호참조가 가능하나 재생이 교착 상태에 빠질 수 있으므로 참조에 따른 해결책이나 재생 순서의 수정이 필요하다. 다음 예는 재생 시간상 후순위로 활성화되는 객체를 참조하는 경우로서, 병렬 재생 시에는 문제가 되지 않는다. 하지만, 아래 순차 재생 예에서 객체 a가 후순위로 재생되는 객체 v를 참조함으로써 객체 a의 재생은 불가능하다.

```
<seq>
<audio id="a" src="..." begin=id(v)(begin)" />
<video id="v" src="..." />
</seq>
```

주어진 예를 함수적으로 표현하면 seq (par\_begin (v.begin,a),v)이며, a와 v의 재생 시간 구간이 0보다 클 경우, 객체 a는 이벤트 객체 v.begin 보다 앞설 수 없으므로 순차 재생에서 모순이 발생한다. 즉, a = (a.begin,a.end)이고, v = (v.begin,v.end)일 경우, seq(a,v)에서 a.begin ≥ v.begin 이면 논리적 모순이 발생하며, par(a,v)에서 a.begin = v.begin 으로 객체 a가 객체 v의 재생 시작을 참조하는 관계이면 의미적 모순이 발생한다.

SMIL 문서를 검증하는 프로시저는 다음과 같으며, 함수적 표현을 가진 트리 구조를 이용하여 문서의 모순을 검증한다. 검증 프로시저는 항상 부모 노드를 기준으로 좌우 서브 트리에 위치한 각 단말 노드들 간의 상호 참조 및 시간 구간의 겹침을 구분하여 모순을 탐색한다.

```
procedure Verification_of_SMIL(tree exp)
{input: exp //tree on functional expression
output: existence of inconsistency
// a: nonterminal node, a ∈ exp
// L, R: left and right sub-trees of a
// β, δ: terminal node of a, β ∈ L, δ ∈ R
for ( all a in exp)
if (a is a "seq" node) {
if ( ((β.begin,β.end) overlaps (δ.begin,δ.end))
or((β references δ for starting or ending)
or((δ references β for starting or ending)))
logical inconsistency occurrence;
} else if (a is a "par_begin", "par_end", or
"par_max" node) {
if (!((β.begin,β.end) overlaps (δ.begin,δ.end)))
and((β references δ for starting or ending)
or((δ references β for starting or
ending)))
semantic or temporal inconsistency occurrence;
} else if (a is an "excl" node) {
// a's child node: βi,βj, i ≠ j, i,j=1, ..., n
if (βi references βj for starting or ending)
logical inconsistency occurrence;
}
}
```

#### IV. 구현 및 검증 사례

검증 사례를 보이기 위하여 본 논문에서는 SMIL 2.0의 베이직 프로파일에 기반을 두는 문서를 이용한다. 베이직 프로파일은 SMIL 2.0 언어 프로파일[1,2]의 부분집합으로 주요 시간 요소로 <seq>, <par>, <excl>을 가지고 있으며, 여기서 주어진 SMIL 문서는 유효(valid)하다고 가정한다. SMIL 문서의 검증을 위한 절차는 SMIL 문서를 먼저 DOM 트리로 바꾸고, 다음 시간 구간 트리로 변환 후 단말 노드의 시간 구간 비교와 참조를 통하여 모순을 탐지한다. 구현은 TurboC를 이용하여 검증 프로시저를 구현하였고, 입력으로 SMIL 문서를 입력받고, 출력으로 모순 유무를 나타낸다.

먼저 첫 번째 검증 사례로서, 병렬적 관계 속에 배타적 관계를 내포하는 다음 SMIL 문서를 그림 2와 같은 시간 구간 트리로 변환하고, 검증 프로시저를 적용하면 비 단말 노드를 중심으로 좌우 서브트리 내에 상호참조가 발생하지 않으므로 모순은 발생하지 않는다. 그러나 만약 약문서에서 v1 객체의 재생 시작이 a1 객체 재생의 시작 또는 종료 이벤트 객체를 참조한다면 주어진 문서는 재생이 불가능하여 논리적 모순을 발생한다.

```
<par endsync="select">


<excl id="select">
<text id="t1" src="..." begin="btn1.activeEvent"
dur="25s"/>
<video id="v1" src="..."
begin="btn2.activeEvent"/>
</excl>
<audio id="a1" src="..." repeat="indefinite"/>
</par>
```

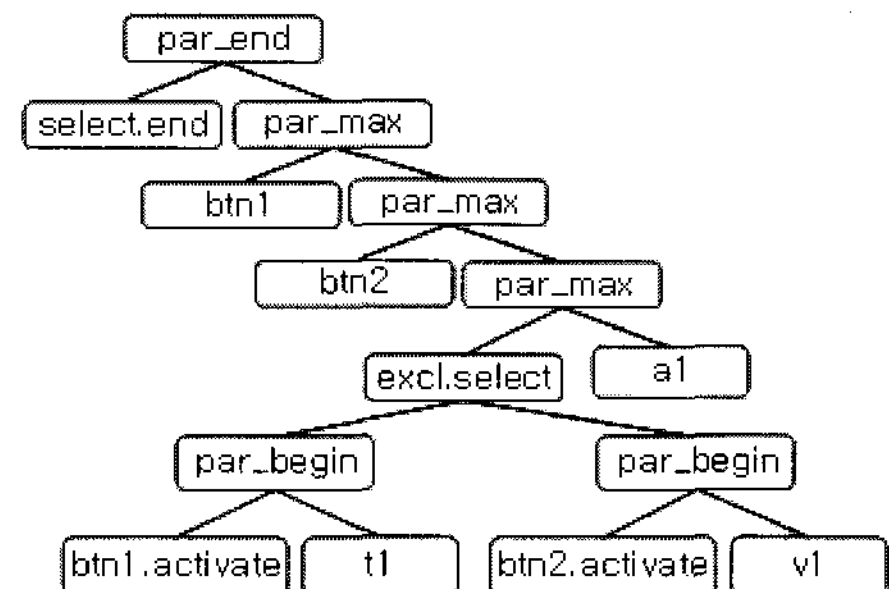


그림 2. 배타적 관계를 가진 경우  
Fig. 2 Case of Exclusive Relation



그러므로 배타적 수행을 가진 경우라도 순차 및 병렬 시간 연산자 노드를 기준으로 서브트리 내의 각 객체간 참조관계를 확인함으로써 검증이 용이하다.

두 번째 검증 사례로서, 참고문헌[2]에서 사용된 SMIL 문서의 예로 순차적, 병렬적, 배타적 관계가 모두 포함된 일반적인 문서이다. 그림 3은 주어진 SMIL 문서의 DOM 트리과 시간 구간 트리 및 검증 결과를 나타낸다. 검증에 사용된 SMIL 문서는 다음과 같다.

```

<smil
xmlns="http://www.w3.org/2001/SMIL20/
Language">
<head>
...
</head>
<body>

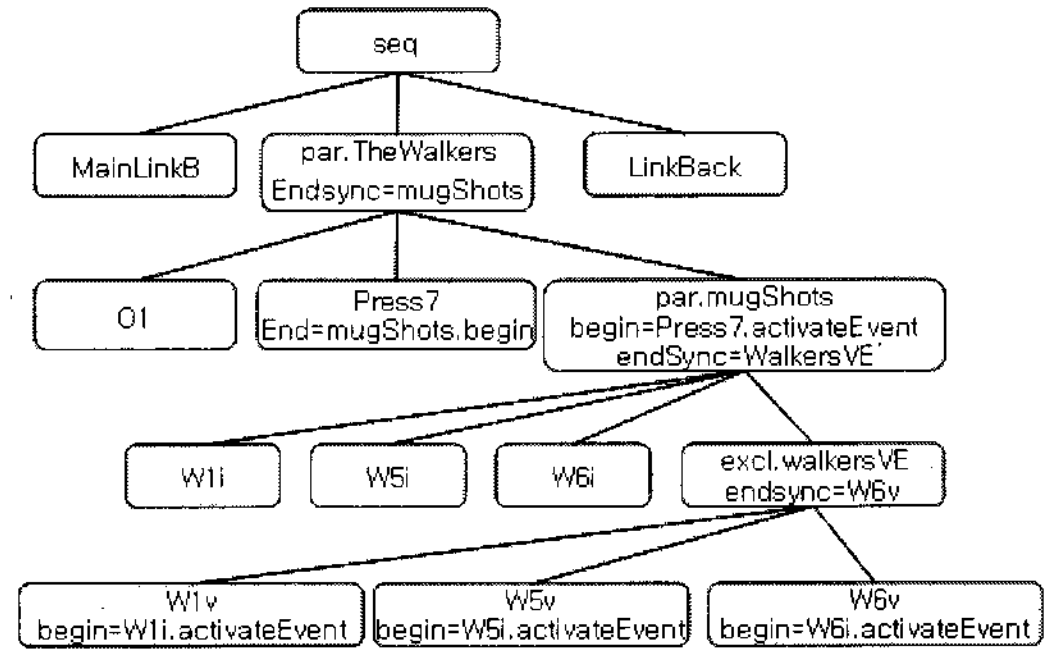
<par id="TheWalkers" endsync="mugShots">


<par id="mugShots"
begin="Press7.adctivateEvent"
endsync="WalersVE">

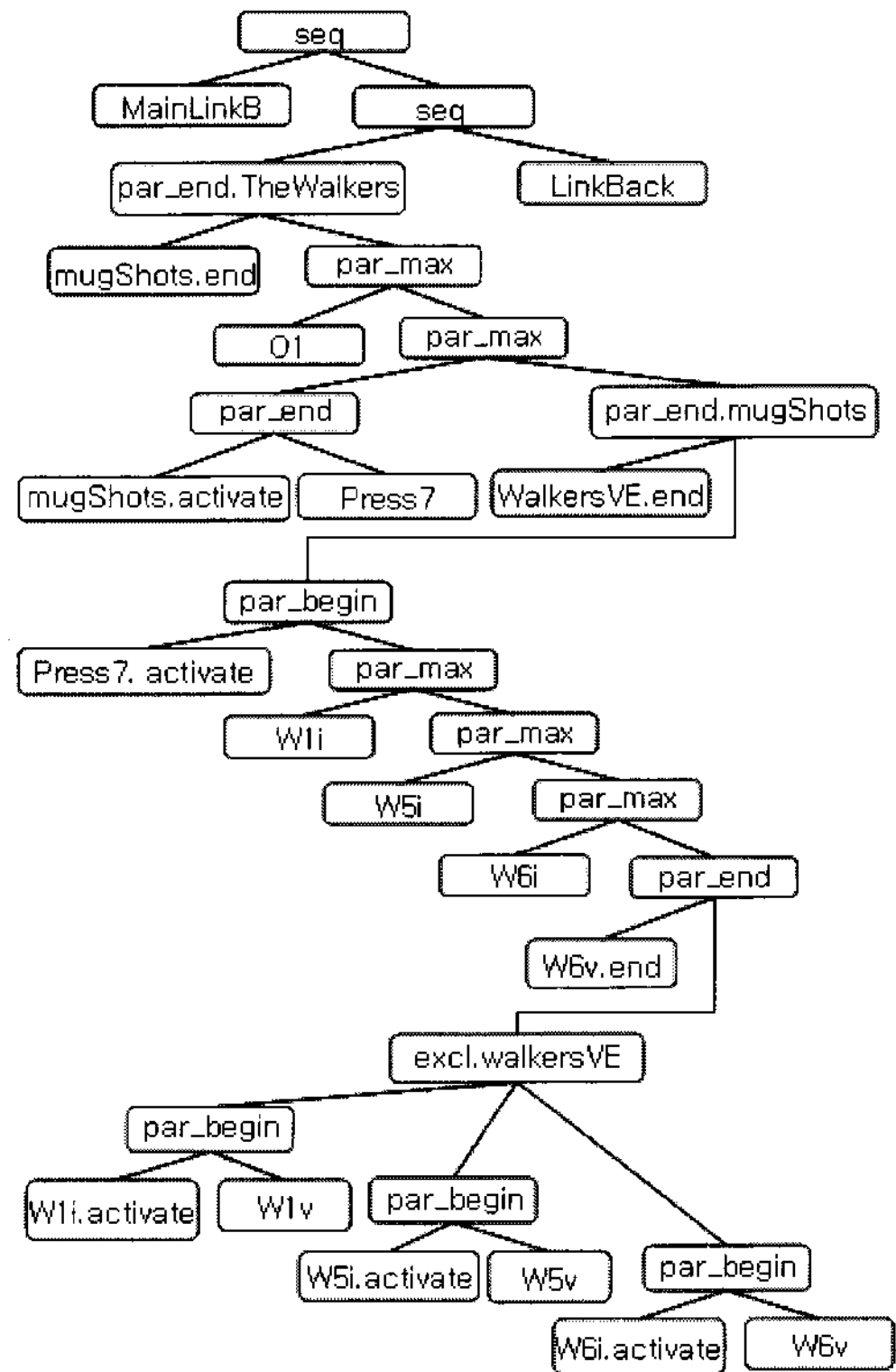


<excl id="WalkersVE" endsync="W6v">
<video id="W1v" region="P3"
begin="W1i.activateEvent" src="W1.avi"/>
<video id="W5v" region="P6"
begin="W5i.activateEvent" src="W5.avi"/>
<video id="W6v" region="P7"
begin="W6i.activateEvent" src="W6.avi"/>
</excl>
</par>
</par>

</body>
</smil>
    
```



(a) DOM 트리



(b) 시간 구간 트리

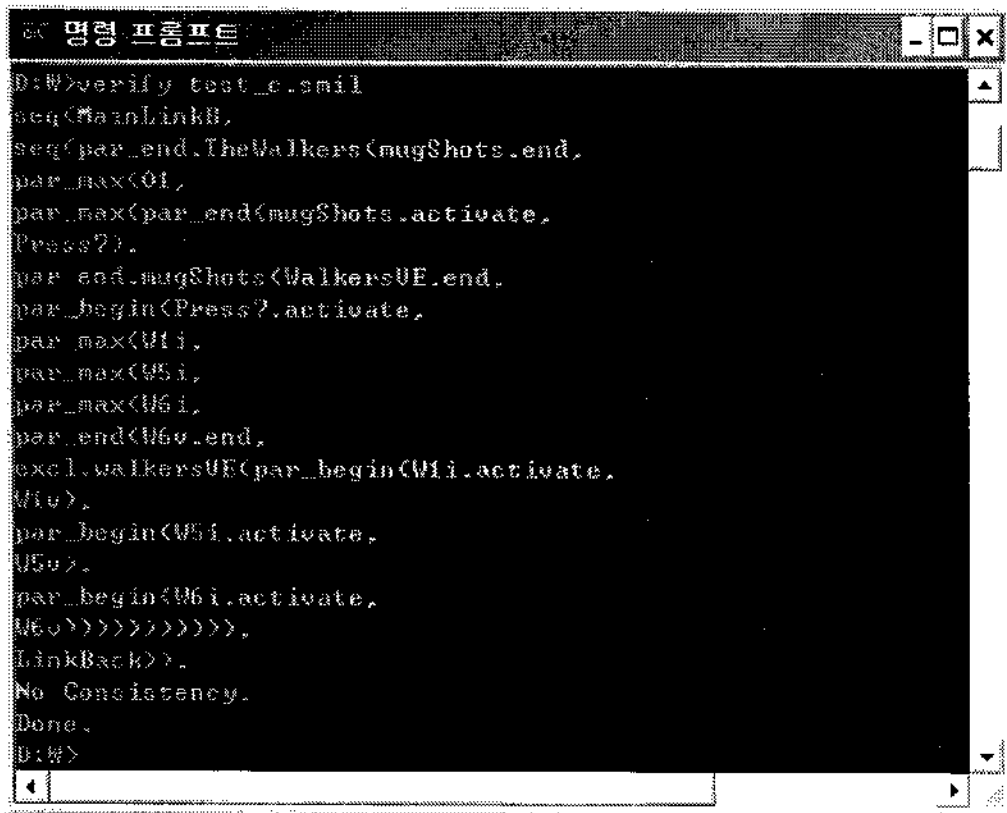
```

seq(MainLinkB,
seq(par_end.TheWalkers(mugShots.end,
par_max(O1,
par_max(par_end(mugShots.activate,
Press7),
par_end.mugShots(WalkersVE.end,
par_begin(Press7.activate,
par_max(W1i, par_max(W5i,
par_max(W6i, par_end(W6v.end,
excl.walkersVE(
    
```

```

par_begin(W1i.activate,W1v),
par_begin(W5i.activate,W5v),
par_begin(W6i.activate,W6v))))))))),
LinkBack)).
    
```

(c) 시간 구간의 함수적 표현



(d) 검증 결과

그림 3. 복합적 관계를 가지는 SMIL 문서의 검증 예

Fig. 3 Case of Verification on a SMIL Document with a Complex Relation

SMIL 문서 검증 프로시저는 그림 3의 (b)와 같이 시간 구간 트리에서 순차, 병렬, 배타 연산자를 의미하는 비단말 노드를 기준으로 좌우측 서브 트리의 단말 노드 간 상호참조 및 시간 구간의 중첩 관계를 확인한다. 즉, 검증 프로시저에서 시간 구간의 순차적 관계를 검증하는 첫 번째 조건문에서 레벨1과 2의 순차연산자 비단말 노드의 좌우측 서브트리의 단말 노드 중에서는 재생 객체가 상호 참조에 의한 시작 및 종료 관계를 가지지 않음을 확인하고, 시간 구간의 병렬 및 배타적 관계를 검증하는 두 번째 및 세 번째 조건문에서 병렬 및 배타연산자 비단말 노드의 좌우측 서브트리의 단말 노드 중에서도 재생 객체의 시간 구간 겹침 없이 시작 및 종료 속성이 상호 참조 관계를 가지지 않음을 확인한다. 이와 같이 검증 프로시저는 주어진 SMIL 문서의 시간 구간에서 모순이 없음을 검증한다.

SMIL2.0 문서는 이벤트에 의하여 객체의 재생 시간과 순서가 변화되고, 순차 및 병렬 수행관계에서 내부 및 외부 이벤트 참조에 의하여 모순이 발생할 수 있어 문서의 검증이 쉽지 않다. 그러나 제안한 검증 방법은 SMIL

2.0 문서를 함수적 관계로 표현하고 객체의 재생 시간 구간과 상호 참조관계를 이용하여 발생 가능한 모순을 탐지하여 수정이 용이하다.

## V. 결 론

본 논문에서는 다양한 상호작용을 지원하는 SMIL 2.0 문서를 시간 구간에 기반을 둔 함수적 표현으로 변환하여 재생 객체간 시간 관계를 검증하는 방법을 제안하였다. 검증을 위하여 기존 시간 구간 모델을 확장하였고, SMIL 문서의 이벤트를 객체화하여 확장된 시간구간 모델에 따라 함수적 표현으로 변환함으로써 문서의 검증을 용이하게 하였다. 검증은 객체간 시간 구간과 상호참조 관계를 이용하여 수행하였다. 기존 검증 방법들도 이벤트 관련한 표현 방법을 제공하고 있으나 사용방법이 복잡하고 표현이 어려워 검증 및 분석이 미흡하다. 그러나 본 논문에서 제안한 검증 방법은 이벤트 및 사용자 상호작용을 함수적 표현에 따라 구조적으로 변환하여 문서의 검증과 분석이 용이하였다.

향후 연구 방향은 확장된 시간구간 표현을 객체 재생을 위한 스케줄링 방법에 적용하는 연구이다.

## 참고문헌

- [1] C. A. Dick, Bulterman, "SMIL 2.0 Part1: Overview, Concepts, and Structure," IEEE Multimedia, pp.82-88, 2001.
- [2] C. A. Dick, Bulterman, "SMIL 2.0 Part2: Examples and Comparisons," IEEE Multimedia, pp.74-84, 2002.
- [3] P. Ackermann, "Direct Manipulation of Temporal Structures in a Multimedia Application Framework," Proc. of ACM Multimedia Conf., pp.51-58, 1994.
- [4] M. Vazirgiannis, "Specifying and Authoring Multimedia Scenarios," IEEE Multimedia, pp.24-37, Vol. 6, Issue 3, 1999.
- [5] S. Chung and A. Pereira, "Timed Petri Net Representation of SMIL," IEEE Multimedia, pp.64-72, Jan.-Mar. 2005.
- [6] R. Tan and S. Guan, " A Dynamic Petri Net Model for



Iterative and Interactive Distributed Multimedia Presentation," IEEE Transactions on Multimedia, Vol. 7, No. 5, pp.869-879, 2005.

- [7] P.N.M. Sampaio, J.P. Courtiat, "Providing Consistent SMIL 2.0 Document," Proc. of the IEEE Conf. on Multimeida and Expo 2002, Vol. 2. pp. 337-340, 2002
- [8] C. Keramance and A. Duda, "Interval expressions-a functional model for interactive dynamic multimedia presentations," Proc. of the 3rd IEEE Int. Conf. on Multimedia Computing and Systems, pp.283-286, 1996.
- [9] 김경덕, "함수적 변환을 이용한 SMIL2.0 문서의 검증," 한국정보과학회 가을학술발표논문집(c), 34권, 2호, pp. 284-287, 2007.

### 저자 소개

김 경 덕(Kyungdeok Kim)



1989년 경북대학교 자연과학대학  
(이학사)

1991년 경북대학교 컴퓨터공학과  
(공학석사)

1999년 경북대학교 컴퓨터공학과 (공학박사)

1991년 ~ 1996년 (주)웨스트시스템 기술연구소

2000년 ~ 현재 위덕대학교 컴퓨터공학부 부교수

※관심분야: 멀티미디어 시스템, 분산 멀티미디어,  
모바일 멀티미디어