A Notion of Equivalence for Multimedia Documents

Paola Bertolotti
Università di Torino
bertolot@di.unito.it

Ombretta Gaggi
Università Ca' Foscari di Venezia
ogaggi@dsi.unive.it

Abstract

In this paper we aim at defining a notion of equivalence *for multimedia documents which can be described according to different models. If we consider a presentation as a collection of media items and constraints among them, the same temporal behavior can be defined in different ways. In particular, using SMIL language, different sequences of tags can describe the same temporal behavior which can be translated in more than a SMIL script.*

In this paper, we propose the use of automata to describe the behavior of a multimedia document independently from the synchronization model it uses, and we define a notion of equivalence to decide when two documents have the same temporal evolution.

1. Introduction

Multimedia presentations are more and more present in web pages; e.g., many web sites contain Flash [8] introductions, and SMIL [10] documents begin to be popular in the Internet.

A multimedia presentation can be defined as a collection of different types of media items and a set of spatial and temporal constraints among them. Authoring such complex documents is more difficult when they are interactive, since uncontrolled user interaction can alter the correct timing relationships between media objects. In literature, many models [1, 5, 7] have been proposed to solve this problem. Among them, SMIL [9, 10], Synchronized Multimedia Integration Language, is a W3C standard markup language to describe multimedia presentations.

In this paper we aim at defining a notion of *equivalence* for multimedia documents which can be described according to different models. If we consider a presentation as a collection of media items and constraints among them, the same temporal behavior can be defined in different ways. In particular, using SMIL language, different sequences of tags can describe the same temporal behavior which can be translated in more than a SMIL script.

Consider for example, the following SMIL document:

⟨par id="par1" dur="60s"⟩
    ⟨video id="intro" end="20s"/⟩   (1)
    ⟨audio id="artwork" begin="intro.end" dur="30s"/⟩
    ⟨img id="picture" begin="artwork.begin+5"/⟩
⟨/par⟩

Intuitively, its temporal behavior is described in Figure 1: the video starts with the script, the audio starts when the video ends (as specified by the value of the `begin` attribute) and the image starts five seconds after the audio. After the end of media *artwork*, *picture* is rendered until the termination of the whole script.

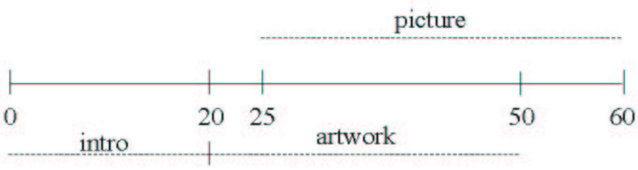

Figure 1. Timeline of the SMIL documents

The same behavior is also obtained in a different way, since the use of attributes `begin`, `end` and `dur` can deeply modify the behavior of the corresponding media items, as described by the following SMIL script:

⟨seq id="seq2"⟩
    ⟨video id="intro" dur="20s"/⟩
    ⟨par id="par2"⟩
        ⟨audio id="artwork" end="30s"/⟩   (2)
        ⟨img id="picture" begin="5s" dur="35s"/⟩
    ⟨/par⟩
⟨/seq⟩

In fact, at the end of the media *intro*, the audio starts, followed by *picture* after five seconds. The image ends thirty-five seconds after its begin, i.e., sixty seconds after the begin of the whole document.

These two documents are, therefore, *behaviorally equivalent*. In this paper we discuss the use of automata to describe the temporal evolution of a multimedia document and we define a notion of *behavioral equivalence* based on the comparison of these automata.

Our notion of equivalence can be applied to multimedia presentations described using different models as long as their behaviors can be described by an automaton; we use the language SMIL only as a case study in the rest of the paper, since this standard allows to describe the same behavior in more than one script.

2. The SMIL language

Among several models available in literature to describe multimedia documents' dynamics, SMIL [9, 10], Synchronized Multimedia Integration Language, is a W3C standard markup language, which is getting more and more used in this field.

A SMIL file is divided in two sections: the `layout` section defines the regions, i.e. rectangular areas on the user screen in which media items are visualized, and the `body` section contains the definitions of media items involved in the presentation and the temporal relationships among them. The language SMIL also allows to define *transitions*, i.e., visual effects between two objects, and *animations*, i.e., modifications to the value of some attributes (e.g., the color, the size, the position, etc.) of a media item.

SMIL does not define a reference model for the data structure, but only tags for describing media objects behavior. Synchronization is achieved essentially through two tags: `seq` to render two or more objects sequentially, one after the other, and `par` to play them in parallel.

As already seen in Section 1, using attributes `begin` and `end` it is possible to change the start and the end time of a media item. Considering a media item inside a `par` block, if its attribute `begin` (`end`) is undefined, it starts at the beginning of the `par` block (ends at the end of it). Otherwise it begins (ends) a certain amount of time after the beginning (end) `par` block equal to the corresponding attribute value. In example 1, the attribute `begin = ''intro.end''` makes audio *artwork* to start when video item *intro* ends. Therefore, the beginning and the end of a media item can be defined with reference to the tag in which it is contained or according to a particular event, even completely changing the sematic of tags `par` and `seq`, as in the case of example 1.

The attribute `dur` defines the duration of an object and attributes `clipBegin` and `clipEnd` allow to play segments inside the time span of an object.

The tag `excl` is used to model some user interactions. It provides a list of children elements and only one of them may play at any given time.

3. A formal description of the temporal evolution of a multimedia document

Multimedia documents dynamics can be described through the use of a great variety of design models, each of which provides some advantages and some drawbacks. Sometimes, the same evolution along the time can be represented in several manners, even using the same model. Therefore we need a notion of *equivalence* to decide when two multimedia documents have the same behavior, i.e., given an event, they have the same reaction in terms of the result perceivable by the user.

To this end, we provide a formal description of the evolution of a presentation which is independent from the reference model of the document. Since the user interaction is not always managed in the same way by different models, in this work we consider only the natural evolution of a multimedia document.

The playout of a multimedia document can be described in terms of media objects presented to the user, the events which cause a media item to start or end, the spatial and synchronization constraints which describe dynamic media behavior and their layout, and a set of timers which represent the temporal constraints such as offset between media objects playback or constraints on their durations. As an example, in the case of SMIL documents, these constraints are defined by the attributes `begin`, `end` and `dur`. Such elements provide an intensional representation of the evolution of the presentation in time.

Since we are interested in the temporal evolution of a multimedia document, spatial disposition of the objects is neglected in the rest of the paper.

A formal definition of a multimedia document is as follows:

Definition 3.1 (Multimedia Document) We can define a multimedia document as a 4–*tuple* $D = \langle \mathcal{MI}, \mathcal{TS}, \mathcal{E}, \mathcal{TC} \rangle$ where

- $\mathcal{MI}$ is a set of media items $\{m_0, m_1, \ldots, m_n\}$;
- $\mathcal{TS}$ is a set of timers;

- $\mathcal{E}$ is a set of events $e \in \mathcal{ET} \times \mathcal{MI} \cup \mathcal{TS} \times \mathbb{N}$, where $\mathcal{ET} = \{start,\ end\ ^1\}$ is the set of event types and $i \in \mathbb{N}$ is the time instant in which an event occurs and

- $\mathcal{TC}$ is a set of temporal constraints representing the synchronization relationships among the objects, described according to the reference model of the document.

For clarity we shall denote event instances by pairs of the form $\langle e(m), i\rangle$ where e is an event type, m is a media item or a timer, and i is the time instant in which the event occurs.

If we observe a multimedia document evolution along time, it can be divided into a number of time intervals in which some conditions hold, e.g., some media are active while other are waiting for the correct time instant. If $\mathcal{MI}$ is the set of media components that play a role in a presentation, we can describe the presentation evolution in terms of active media at any given time instant. We assume that time is discrete, and marked by a variable i ranging over $\mathbb{N}$ which is updated by a clock. The actual time resolution is not important as long as it allows the capture of all the events related to media execution and it is possible to observe the effect of time distinct events as distinct effects. Two or more events are *contemporary* if they occur at times denoted by the same value of the variable i. In this case, the instances with the same i are denoted by a unique complex event in the form $\langle\{e(m)\}, i\rangle$, where $\{e(m)\}$ is the set of events that occur at the same time instant i.

A timer is a dynamic media object whose length is defined by the author at the design time and which ends at fixed time unit, said n. A document contains as many $t(n) \in \mathcal{TS}$ as its offsets and its temporal constraints. We define a mapping function *waiting*: $\mathcal{TS} \to 2^{\mathcal{MI}}$ which returns, for each timer, the list of media items that are waiting for it.

We introduce also the two functions *begin*, *stop*: $\mathcal{MI} \to \mathbb{N}$ which return, for each element contained in the document, the time instant in which it starts or ends, respectively.

Considering the SMIL language as a case study, since we are interested in the behavior of the document, we can investigate only the section `body` which contains the synchronization tags among the object: in this case the set $\mathcal{TC}$ of temporal constraints is equal to the set $\mathcal{T}ag$ of tags contained in the SMIL script. For the same reason, we do not consider transition effects.

[1]Since we describe only the natural evolution of the document, there is only a *start* event which starts of the presentation, and then a sequence of *end* events of timers and media items.

Each tag $tg \in \mathcal{T}ag$ has three attributes which we denote by $tg.begin$, $tg.end$ and $tg.dur$, which contain the values of the same attributes in the SMIL documents (zero if undefined). The function $parent(tg)$ returns the tag in which tg is contained and $children(tg)$ returns the list of tags defined inside the considered tag; $first(tg)$ returns the first child. A *succ()* function is also available that, given a tag in the list of children, gives as output the next element in that list[2].

Tables 1 and 2 show how timers and functions are applied to script 1 introduced in Section 1.

values	Tags			
	par1	intro	artwork	picture
attribute `begin`	0	0	intro. `end`	(artwork. `begin`)+5
attribute `dur`	60	0	30	0
attribute `end`	0	20	0	0
parent()	null	par1	par1	par1
children()	intro artwork picture	null	null	null
first()	intro	null	null	null
succ()	null	artwork	picture	null
begin()	0	0	20	25
stop()	60	20	50	60

Table 1. Values of attributes and functions for the tags in script 1.

timer	type of control	*waiting()*
$t_1(60)$	termination	$\{intro, artwork, picture\}$
$t_2(20)$	termination	$\{intro\}$
$t_3(20)$	begin	$\{artwork\}$
$t_4(25)$	begin	$\{picture\}$
$t_5(50)$	termination	$\{artwork\}$

Table 2. Timers and function *waiting()* applied to script 1.

At any time instant, the document behavior is completely described by the set of media and timers that are active at that time. This information is captured by the notion of *state* of the presentation. Before the presentation starts, no media item or timer is active, thus no media item is waiting for a timer expiration and variable $i = 0$. When an event occurs, the state of

[2]The children list is ordered with the same order in which the media items (or tags) appear in the SMIL file.

the presentation changes: some items that were not active become active, some active items end, some timers expire and force the starting or the interruption of a list of items.

Definition 3.2 (State) The state of a multimedia presentation is a 5–tuple $S = \langle \mathcal{AM}, \mathcal{T}, \mathcal{B}g, \mathcal{E}nd, \mathcal{W} \rangle$, where $\mathcal{AM}$ is the set of active media, $\mathcal{T}$ is the set of timers whose expiration denotes an instant in which some objects start or end, $\mathcal{B}g$ and $\mathcal{E}nd$ are the set of pairs $\langle m, i \rangle$ where m is a media item and i is the time instant in which the item begins or ends, respectively, as defined by the functions *begin* and *stop* and $\mathcal{W}$ is the set of pairs $\langle t, mi \rangle$ where t is a timer and mi is a set of media items which are waiting for the timer t as defined by the mapping *waiting* above.

For clarity, in the following of the paper we shall refer to the association between media items and time instants with $begin(m) = i$ if $\langle m, i \rangle \in \mathcal{B}g$ and $stop(m) = i$ if $\langle m, i \rangle \in \mathcal{E}nd$, and between timers and media items with the functional notation $waiting(t(i)) = mi$ if $\langle t(i), mi \rangle \in \mathcal{W}$.

If we observe the system along time, the only relevant time instants are the *observable time instants*, i.e., the time instants in which an event occurs, or the effects of an event are assessed and perceivable. Indeed, these are the time instants in which something in the state of the presentation might change, as a consequence of the occurred event.

The state of a multimedia presentation is thus a function of *observable time instants*. We assume that at any observable time instant at least one event occurs, i.e., either a media item is activated or naturally ends or a timer expires. At the same time instant other items may be activated, according to the synchronization rules.

All the possible evolutions along time of a multimedia document D can be described by a finite state automaton, defined as follows.

Definition 3.3 (Automaton) Let $D = \langle \mathcal{MI}, \mathcal{TS}, \mathcal{E}, \mathcal{TC} \rangle$ be any multimedia document. Its associated finite state automaton is the 5–*tuple* $AUT(D) = \langle \S, \mathcal{E}, s_0, \textit{next}, \textit{Final} \rangle$, where

- $\S$ is the set of possible states for the document D;
- $\mathcal{E}$ is the set of possible event instances in the form $\langle \{e(m)\}, i \rangle$, $e \in \{start, end\}$, $m \in \mathcal{MI} \cup \mathcal{TS}$, $i \in \mathbb{N}$;
- s_0, the initial state, is $\langle \mathcal{AM}_0, \mathcal{T}_0, begin_0, end_0, waiting_0 \rangle$, where $\mathcal{AM}_0 = \mathcal{T}_0 = \emptyset$, $begin_0(el) = end_0(el) =$ null for all $el \in \mathcal{MI}$ and $waiting_0(t) = \emptyset$, for all $t \in \mathcal{TS}$;
- the transition function $next : \S \times \mathcal{E} \rightarrow \S$ is the mapping that deterministically associates any state s to the state s' in which s is transformed by an event $e \in \mathcal{E}$;
- $Final = \{s_f, s_{err}\}$, is the set of states which correspond to the end of the presentation playback. The state s_f is $\langle \mathcal{AM}_f, \mathcal{T}_f, begin_f, end_f, waiting_f \rangle$, where $\mathcal{AM}_f = \mathcal{T}_f = \emptyset$ and $waiting_f(t) = \emptyset$, for all $t \in \mathcal{TS}$. The state s_{err} is equal to $\langle \mathcal{AM}_{err}, \mathcal{T}_{err}, begin_{err}, end_{err}, waiting_{err} \rangle$, where $\mathcal{AM}_{err} = \emptyset$, $\mathcal{T}_{err} = \mathcal{TS}$, $begin_{err}(el) = end_{err}(el) =$ null for all $el \in \mathcal{TC}$ and $waiting_{err}(t) = \emptyset$, for all $t \in \mathcal{TS}$.

The state s_f corresponds to the natural termination. If the document evolution reaches state s_{err}, then the presentation contains some time conflicts.

We note here that, if the sets of media items $\mathcal{MI}$ and timers $\mathcal{TS}$ of the document D are finite, also $AUT(D)$ is finite, even if it can contain infinite paths.

4. A definition of equivalence

The automaton defined in Section 3 completely describes the behavior of a multimedia document independently from the synchronization relationships it contains. Therefore, it can be used to check if two documents have the same behavior: since, from the user point of view, the only relevant modifications are changes in the set of active media $\mathcal{AM}$, two documents have the same behavior when the states of the respective automata after the events $\langle \{e(m)\}, i \rangle$ contain the same set of active media. More formally:

Definition 4.1 (Behavioral Equivalence) Let D and D' be two multimedia documents, $D = \langle \mathcal{MI}, \mathcal{TS}, \mathcal{E}, \mathcal{TC} \rangle$, $D' = \langle \mathcal{MI}', \mathcal{TS}', \mathcal{E}', \mathcal{TC}' \rangle$ and $\mathcal{MI} = \mathcal{MI}'$. D and D' are *behaviorally equivalent*, denoted by $D \sim D'$, iff:

- for each word $w = \langle e_1(el_1), ist_1 \rangle \ldots \langle e_i(el_i), ist_i \rangle \ldots \langle e_n(el_n), ist_n \rangle$, where $\forall i\ ist_i \leq ist_{i+i}$, accepted by $AUT(D)$ exists a word $w' = \langle e'_1(el'_1), ist'_1 \rangle \ldots \langle e'_i(el'_i), ist'_i \rangle \ldots \langle e'_n(el'_n), ist'_n \rangle$, where $\forall i\ ist'_i \leq ist'_{i+i}$, accepted by $AUT(D')$ such that $\forall i = 1 \ldots n\ ist_i = ist'_i$ and if $el \in \mathcal{MI}$ then $el_i = el'_i$ and viceversa;
- for each pair of words w, w' accepted by the automata if $sq = s_0 \ldots s_j \ldots s_n$ and $sq' = s'_0 \ldots s'_k \ldots s'_n$ are the sequences of states reached by $AUT(D)$ and $AUT(D')$, respectively, and $s_i = \langle \mathcal{AM}_i, \mathcal{T}_i, \mathcal{B}g_i, \mathcal{E}nd_i, \mathcal{W}_i \rangle$ and $s'_i = \langle \mathcal{AM}'_i, \mathcal{T}'_i, \mathcal{B}g'_i, \mathcal{E}nd'_i, \mathcal{W}'_i \rangle$ then $\forall i = 0 \ldots n$ $\mathcal{AM}_i = \mathcal{AM}'_i$.

5. A case study: behavioral equivalence for SMIL documents

Definition 3.3 can describe the behavior of multimedia documents designed with any synchronization model properly defining *next* function. As an example, in [4] the authors discuss the use of an automaton to describe the behavior of a presentation designed according to an event-based model.

In this paper we provide the definition of the *next* function needed to describe the behavior of SMIL scripts. In order to define the state transformation function we need to define a set of auxiliary functions. First of all, we need two functions **ChkBEGIN** and **ChkEND** which calculate the start and the end time of an element following the W3C specification and algorithms for SMIL documents [10]. If any such values can be calculated, i.e., it does not depend on a future element which has not been considered yet, the functions activate a timer which expires at the correct time instant, by putting it in the set of active timers $\mathcal{T}$, and update the mapping functions *begin* and *stop*. Function **ChkBEGIN**, if necessary, puts the element, or its children if the element is a tag `par`, into the list of items waiting for the activated timer (i.e., $waiting(t)$ where t is the activated timer) and it returns the value True; otherwise it returns False. Function **ChkEND** checks if there are some time conflicts, i.e., if different end time values can be calculated according to the attributes defined for that element or for one of its parents. In this case, it changes the state to s_{err} as defined in Definition 3.3 and returns the value True. Otherwise it puts the element, or its children if the element is a tag `par` or `seq`, into the list of items waiting for the activated timer.

Another useful function is **NextEL** which, given a tag as input, returns the next element to playback, according to the synchronization constraints contained in the SMIL script.

All the functions defined have as input the considered element, denoted by x, the set $\mathcal{TS}$ which contains all the timers of the document, the set of active timers $\mathcal{T}$, the current time instant t_{curr}, the set of active media $\mathcal{AM}$ (see Definition 3.2) and the mapping function *begin*, *stop* and *waiting* defined in Section 3.

For the sake of readability we introduce two functions that take care of the two most complex actions, i.e., activating and deactivating a media item. The **ACTIVATE** function checks if a media item can start or if it must wait for a delay. In the first case it starts the element otherwise it simply activates a timer (function **ChkBEGIN**).

ACTIVATE (x: a SMIL Tag; $\mathcal{TS}$, $\mathcal{T}$: set of timers;
    t_{curr}: time instant; $\mathcal{AM}$: set of media
    items; begin, stop: item-time mapping;
    waiting: timer-media mapping)
begin
  if ChkBEGIN(x, $\mathcal{TS}$, $\mathcal{T}, t_{curr}, begin, stop, waiting$)
  then exit ;
  else ActEL (x, $\mathcal{TS}$, $\mathcal{T}, t_{curr}, \mathcal{AM}, begin, stop,$
      $waiting$)
end.

The function **ActEL** checks if the element to be activated is a media item, or a `par` or a `seq` block. In the first case it adds the element to the set of active media $\mathcal{AM}$, otherwise, it starts all its children (if it is a `par` block) or only its first child (if it is a sequence of items).

ActEL (x: a SMIL Tag; $\mathcal{TS}$, $\mathcal{T}$: set of timers;
    t_{curr}: time instant; $\mathcal{AM}$: set of media
    items; begin, stop: item-time mapping;
    waiting: timer-media mapping)
begin
  $y = first(x)$;
  if y = null **then**   // x is a media item
    if ChkEND(x, $\mathcal{TS}$, $\mathcal{T}, t_{curr}, \mathcal{AM}, begin,$
      $stop, waiting$) **then exit** ;
    else $\mathcal{AM} = \mathcal{AM} \cup \{x\}$;
  else
   begin
    if x = "seq" **then**
      if ChkEND(x, $\mathcal{TS}$, $\mathcal{T}, t_{curr}, \mathcal{AM}, begin,$
        $stop, waiting$) **then exit** ;
      else ACTIVATE (y, $\mathcal{TS}$, $\mathcal{T}, t_{curr}, \mathcal{AM},$
        $begin, stop, waiting$)
    if x = "par" **then**
      if ChkEND(x, $\mathcal{TS}$, $\mathcal{T}, t_{curr}, \mathcal{AM}, begin,$
        $stop, waiting$) **then exit** ;
      else for all $child \in children(x)$ **do**
        ACTIVATE($child$,$\mathcal{TS}$, $\mathcal{T}, t_{curr}, \mathcal{AM},$
          $begin, stop, waiting$);
   end
end.

The function **DeACTIVATE** stops a single media object if it is contained in the set of active media $\mathcal{AM}$ using the function **StopMEDIA**. When applied to a `par` or a `seq` block, if this block was waiting for a timer, the next element, according to the synchronization rules, is started, otherwise all active elements in the block are deactivated.

DeACTIVATE (x: a SMIL Tag; $\mathcal{TS}$, $\mathcal{T}$: set of timers;
    t_{curr}: time instant; $\mathcal{AM}$: set of media
    items; begin, stop: item-time mapping;
    waiting: timer-media mapping)
begin
  if $first(x)$ = null **then** // x is a media item
    if $x \in \mathcal{AM}$ **then**

StopMEDIA (x, $\mathcal{TS}$, $\mathcal{T}, t_{curr}\mathcal{AM}, begin,$
$stop, waiting$);

else // x is a block `par` or `seq`

begin

$Set = children(x) \cap\ \mathcal{AM}$; // set of active media

// x was waiting for a timer

if $Set = \emptyset$ **then**

if $stop(x) = t_{curr}$ **then**

begin

y =**NextEL**(x);

if $y \neq$ null **then**

ACTIVATE(y,$\mathcal{TS}$, $\mathcal{T}, t_{curr}, \mathcal{AM},$
$begin, stop, waiting$);

end

else for all $child \in Set$ **do**

DeACTIVATE ($child$, $\mathcal{TS}$, $\mathcal{T}, t_{curr}, \mathcal{AM},$
$begin, stop, waiting$);

$stop(x) = t_{curr}$;

end

end.

The function **StopMEDIA** deactivates a media item removing it by the set $\mathcal{AM}$. If the media item is contained in a `par` block, the function checks if the block ends with this media (e.g., when its attribute `end` is equal to the end event of this media), and in this case it stops the whole block and activates the next element. If the media item is contained in a `seq` block, the function starts the next element according to the SMIL script.

StopMEDIA(x: a SMIL Tag; $\mathcal{TS}$, $\mathcal{T}$: set of timers;
t_{curr}: time instant; $\mathcal{AM}$: set of media items; begin, stop: item-time mapping; waiting: timer-media mapping)

begin

$\mathcal{AM} = \mathcal{AM} \setminus \{x\}$;

$y = parent(x)$;

if $y =$ null **then exit** ;

if $y =$ "par" **then**

if $y.end =$ "first" **or** $y.end = x$ **then**

begin

DeACTIVATE(y,$\mathcal{TS}$, $\mathcal{T}, t_{curr}, \mathcal{AM}, begin,$
$stop, waiting$);

y =**NextEL**(y);

if $y \neq$ null **then**

ACTIVATE(y,$\mathcal{TS}$, $\mathcal{T}, t_{curr}, \mathcal{AM}, begin,$
$stop, waiting$);

end

else

// check if x is the last child to terminate and

// there are no attributes end and dur or its

// timer is expired now

if $children(y) \cap \mathcal{AM} = \emptyset$ **and** ($stop(y) =$ null

or $stop(y) = t_{curr}$) **then**

begin

y =**NextEL**(y);

if $y \neq$ null **then**

ACTIVATE(y,$\mathcal{TS}$, $\mathcal{T}, t_{curr}, \mathcal{AM}, begin,$
$stop, waiting$);

end

else //$y =$ "seq"

begin

if $succ(x) \neq$ null **or** ($stop(y) =$ null

or $stop(y) = t_{curr}$) **then**

begin

y =**NextEL**(x);

if $y \neq$ null **then**

ACTIVATE(y,$\mathcal{TS}$, $\mathcal{T}, t_{curr}, \mathcal{AM}, begin,$
$stop, waiting$);

end

end

end.

Definition 5.1 (State transition function) The state transition function $next : \S \times \mathcal{E} \rightarrow \S$, where $\S$ is the set of all possible states, and $\mathcal{E}$ is the set of events, is the function that, given a state s and a complex event $ev(i) = \langle\{e_k\}, i\rangle$ at the observable time instant i, returns the state $s' = next(s, ev(i))$ at the observable time instant $i+1$ where $s' = \langle\mathcal{AM}', \mathcal{T}', \mathcal{B}g', \mathcal{E}nd', \mathcal{W}'\rangle$, and $\mathcal{AM}'$, $\mathcal{T}'$, $\mathcal{B}g'$, $\mathcal{E}nd'$, $\mathcal{W}'$ are defined according to the following process, in which $ev = \{e_k\}$ is set of occurring events, el the element to which the event applies and i is the current time instant:

for all $e \in ev$ **do**

case $e = start(el)$:

if $el \notin \mathcal{AM}_n$ **then**

ACTIVATE (el, $\mathcal{TS}$, $\mathcal{T}, i, \mathcal{AM}, begin, stop,$
$waiting$);

case $e = end(el)$:

begin

// ignore an event end(el) with $el \in \mathcal{AM}$

// if el has an attribute dur or end

if $el \in \mathcal{AM}$ **and** ($stop(el) =$ null

or $stop(el) = i$) **then**

begin

StopMEDIA (el, $\mathcal{TS}$, $\mathcal{T}, i, \mathcal{AM}, begin,$
$stop, waiting$);

$stop(el) = i$;

end

if $el \in \mathcal{T}$ **then**

begin

if $stop(m) = i$ for some $m \in \mathcal{T}ag^3$ **then**

for all $item \in waiting(el)$ **do**

begin

DeACTIVATE ($item$, $\mathcal{TS}$, $\mathcal{T}, i,$
$\mathcal{AM}, begin, stop, waiting$);

$stop(item) = i$;

end

if $begin(m) = i$ for some $m \in \mathcal{T}ag^3$ **then**

for all $item \in waiting(el)$

such that $begin(item) = i$ **do**

ActEL ($item$, $\mathcal{TS}$, $\mathcal{T}, i, \mathcal{AM},$

$begin, stop, waiting$);
  $waiting(el) = \emptyset$;
  $\mathcal{T} = \mathcal{T} \setminus \{el\}$;
end

Consider as an example, a multimedia presentation about an artwork: an initial video (*intro*) introduces the history period related to it and the artist who made it. At its end, an audio (*artwork*) starts to describe the particular of the artwork, that is rendered (*picture*) until the end of the script. The SMIL scripts 1 and 2, introduced in Section 1, model this situation, in different ways. Therefore we can apply our definition of behavioral equivalence and construct the two automata to verify that the documents have the same behavior.

If D is the document corresponding to script 1 and D' corresponds to script 2, they both contain the following media: $\mathcal{MI} = \mathcal{MI}' = \{intro, artwork, picture\}$.

The evolution of D is represented in Figure 2: from the initial state, in which neither media nor timers are active, the activation of the `par` block causes the activation of the video *intro*, while the other media are not activated because of their `begin` attributes. Four timers are inserted in $\mathcal{T}$ (see Table 2): t_1 controls the termination of `par1`, t_2 the termination of media *intro*, t_3 and t_4 the beginning of *artwork* and *picture*, respectively. At time instant 20, timers t_2 and t_3 expire: as a consequence, media *intro* ends and media *artwork* becomes active (adding in $\mathcal{T}$ timer t_5 that controls its termination). After five seconds, the expiration of t_4 forces the beginning of media *picture*. At time instant 50, when corresponding timer t_5 expires, media *artwork* ends and, finally, the expiration of the last timer t_1 forces the termination of all active media contained in the `par` block and then of the whole script.

The evolution of D' is represented in Figure 3: the activation of the presentation forces the activation of media *intro* and timer t'_1, that controls its termination, is inserted in $\mathcal{T}$. At time instant 20, this timer expires: media *artwork* becomes active, while media *picture* is not activated because of its `begin` attribute (the corresponding timers t'_2 and t'_3, that control the termination and the beginning of *artwork* and *picture*, respectively, are inserted in $\mathcal{T}$). After five seconds, the expiration of timer t'_3 forces the activation of media *picture* and timer t'_4 for its termination is inserted in $\mathcal{T}$. At time instant 50, timer t'_2 expires and *artwork* ends; finally, the last timer t'_4 expires and forces the termination of *picture* and of the whole document.

[3]Since in SMIL language also media items are defined using appropriate tags (`img`, `video`, or `audio` according to the object's type), media items are contained also in $\mathcal{T}ag$ and $\mathcal{MI} \subset \mathcal{T}ag$.

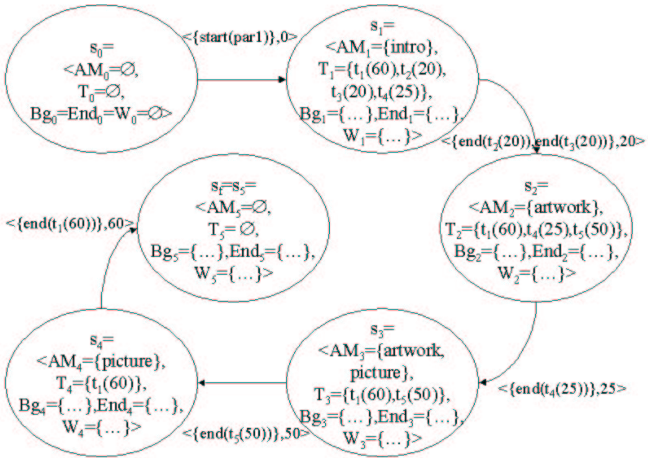

Figure 2. The evolution of script D

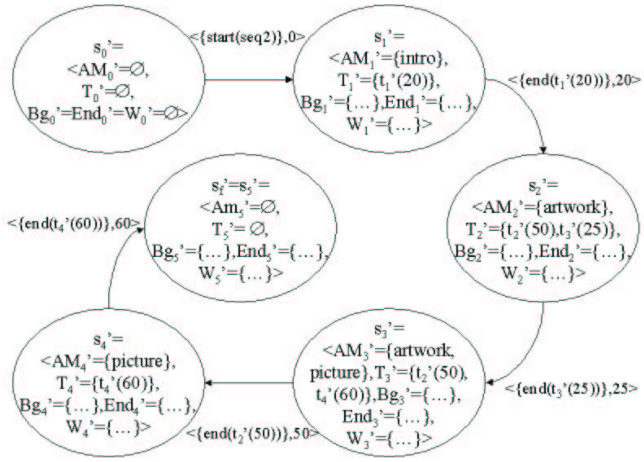

Figure 3. The evolution of script D'

Figures 2 and 3 represent the finite state automata $AUT(D)$ and $AUT(D')$ where the word $w = \langle\ \{start(par1)\}, 0\ \rangle$, $\langle\ \{end(t_2(20)),\ end(t_3(20))\}, 20\ \rangle$, $\langle\{end(t_4(25))\}, 25\rangle$, $\langle\{end(t_5(50))\}, 50\rangle$, $\langle\{end(t_1(60))\}, 60\rangle$ is accepted by $AUT(D)$ with $ist_1 = 0$, $ist_2 = 20$, $ist_3 = 25$, $ist_4 = 50$ and $ist_5 = 60$ and the word $w' = \langle\{start(seq2)\}, 0\rangle$, $\langle\{end(t'_1(20))\}, 20\rangle$,$\langle\{\ end(t'_3(25))\}, 25\rangle$, $\langle\{end(t'_2(50))\}, 50\rangle$, $\langle\{end(t'_4(60))\}, 60\rangle$ is accepted by $AUT(D')$ with $ist'_1 = 0$, $ist'_2 = 20$, $ist'_3 = 25$, $ist'_4 = 50$ and $ist'_5 = 60$.

Therefore we have $\forall i = 1...5$, $ist_i = ist'_i$ and at every instant the events do not involve any element in $\mathcal{MI}$; moreover, does not exist a word accepted by $AUT(D')$ such that there is no corresponding word accepted by $AUT(D)$: then the first step of Definition 4.1 is verified.

Given w and w' accepted by the automata, we obtain the sequences of states $sq = s_o s_1 s_2 s_3 s_4 s_5$ and $sq' = s'_o s'_1 s'_2 s'_3 s'_4 s'_5$ reached by $AUT(D)$ and $AUT(D')$

(detailed in Figure 2 and 3) such that $\forall i = 0...5$, $\mathcal{AM}_i = \mathcal{AM}'_i$: the second step is verified and the two SMIL documents are *behaviorally equivalent* w.r.t. Definition 4.1.

6. Related work

Other works use the concept of equivalence in different contexts and with different purposes.

In [3], Boll et al define a notion of equivalence dealing with the problem of adaptation for multimedia presentation: a multimedia information system must *adapt* the delivery and the rendering of a presentation to the current user context, i.e., his/her preference, the system environment and other resources. In order to adapt and reuse multimedia information to user context, media elements can be adapted/replaced by media elements of different quality and type but that are *semantic equivalent alternatives*. Different from our approach, the authors define a equivalence only with respect to certain aspects of media elements and not with respect to the structure of the document and its components. Moreover, this notion considers also the semantic and the content of items: two media are equivalent if they are equal in specified discriminating aspects, like *duration* and *subject*.

Another work [6] focuses its attention on the temporal evolution of multimedia applications and proposes an extension of Dynamic Meta Modeling (a graph transformation-based approach introduced for the specification of UML's semantics) to specify the semantics of time-dependent dynamic behavior of UML models. Instead of our use of the automata, the authors define a timed graph in which all vertices are attributed with a special attribute that represents the state of the local clock of the object. Then they discuss the equivalence of graph transformation rules with respect to time specification: if it is not possible to distinguish different interleavings of concurrent action, the sequences of transformation steps are equivalent.

Other works in literature study the behavior of a multimedia presentation described using different models, but they do not provide an abstract representation of the dynamics of a multimedia document, independent from the model it adopts.

In particular, Yang [12] focus on the study of the SMIL language to find out possible errors, resulting in temporal conflicts, in the definition of the document. He proposes a conversion of SMIL 1.0 documents [9] in the *Real Time Synchronization Model (RTSM)* This model, based on Petri Nets, provides two kinds of places (*regular places* and *enforced places*) and the firing rule specifies that, when the action related with the enforced place is completed, the following transition will be immediately fired, regardless the state of other places. The synchronization elements in SMIL are converted in transitions and the related attributes are associated with places. Our approach is more general, because it does not apply only to SMIL document and our purpose is not necessary to convert every single SMIL element, but we focus our attention on the overall document behavior, captured by the notion of state in an automaton.

In [11] Sampaio et al. present a formal approach to design an Interactive Multimedia Document (IMD) and to perform the schedule of a presentation, if the temporal constraints are satisfied. Through different steps, a document is translated into a scheduling graph, called *Time Labeled Automaton* (TLA): at every state in the automaton is associated a clock (timer), that measures the time during which the automaton remains in that state. The timers defined in our work are different: they do not represent the time elapsed in a state, but they represent temporal constraints on the media in the presentation; this approach is more flexible, e.g., to describe user interactions which are events without a predefined occurred time.

Automata with temporal characteristics to model the behavior of real-time systems are also used in [2]: the authors define timed automata which are able to express timing delay in real-time systems. A timed automaton has a set of real-valued clocks: they can be reset with a transition and constraints over the clocks control the firing transition, that is, a transition takes place if the associated clocks constraints are satisfied. In this case, the clocks control if an event occurs in the right time instant; once again, our approach is more flexible since the timer itself can generate events for their expiration.

7. Conclusion

The paper proposes a notion of *behavioral equivalence* for multimedia documents based on the comparison of the automata which describe the temporal behavior of each presentation in the absence of user interactions.

Although other works in literature address similar problems, our approach is interesting since the final representation of automaton is independent from the design model, therefore it can be also used to compare documents with different reference models, simply defining the transition function of the automaton for the reference model used by the document.

In this paper we adopt the SMIL language as a case study and we provide the algorithms for the state transition function to build up the automaton of a SMIL document.

The use of automata to represent multimedia document evolution can also be applied to determine if two documents have a common component or if one of them is included in the other one. In this case, it is sufficient to find out equivalent paths, in the automata representing the temporal evolution of the documents. I.e., given the automata of the two documents, two equivalent paths are substrings of the same length of two words, accepted by the two automata respectively, in which events occurs after the same interval of time and the media active, after each step, are the same in both the automata. In other words, a multimedia document is included into another one if its automaton is a subgraph of the automaton corresponding to the bigger document.

In the future, we plan to better investigate this possibility together with the problem of finding time conflicts in a multimedia document's definition.

References

[1] J. F. Allen. Maintaining Knowledge about Temporal Intervals. *Comm. ACM*, 26(11):832–843, November 1983.

[2] R. Alur and D.L. Dill. A Theory of Timed Automata. *Theoretical Computer Science*, 126:183–235, 1994.

[3] S. Boll, W. Klas, and J. Wandel. A Cross-Media Adaptation Strategy for Multimedia Presentations. In *Proc. of Multimedia Conference 1999*, Orlando, Florida, USA, October 30-November 5 1999.

[4] A. Celentano, O. Gaggi, and M.L. Sapino. Retrieving Consistent Multimedia Presentation Fragments. In *Workshop on Multimedia Information Systems (MIS 2002)*, pages 146–154, Tempe, Arizona, USA, November 2002.

[5] L. Hardman, D.C.A. Bulterman, and G. van Rossum. The Amsterdam Hypermedia Model: Adding Time, Structure and Context to Hypertext. *Communications of the ACM*, 37(2):50–62, February 1994.

[6] J.H. Hausmann, R. Heckel, and S. Sauer. Dynamic Meta Modeling with Time: Specifying the Semantics of Multimedia Sequence Diagrams. In *Proc. of Workshop on Graph Transformation and Visual Modelling Techniques (GT-VMT 2002)*, Barcelona, Spain, October 11-12 2002.

[7] M. Jourdan, N. Layaïda, C. Roisin, L. Sabry-Ismail, and L. Tardif. Madeus, an Authoring Environment for Interactive Multimedia Documents. In *ACM Multimedia 1998*, pages 267–272, Bristol, UK, September 1998.

[8] Macromedia Inc. Macromedia Flash. http://www.macromedia.com/software/flash/.

[9] Synchronized Multimedia Working Group of W3C. Synchronized Multimedia Integration Language (SMIL) 1.0 Specification. http://www.w3.org/TR/REC-smil/, June 1998.

[10] Synchronized Multimedia Working Group of W3C. Synchronized Multimedia Integration Language (SMIL) 2.0 Specification. http://www.w3.org/TR/smil20, August 2001.

[11] P.N.M. Sampaio and J.-P. Courtiat. A Formal Approach for the Presentation of Interactive Multimedia Documents. In *Proc. of ACM Multimedia 2000 (short paper)*, Los Angeles, USA, October 30, November 4 2000.

[12] C.-C. Yang. Detection of the time conflicts for SMIL-based Multimedia Presentation. In *Proc. of 2000 Computer Symposium (ICS-2000)-Workshop on Computer Networks, Internet, and Multimedia*, pages 57–63, Chiayi, Taiwan, 2000.