Enriching SMIL with Assertions for Temporal Validation

Annalisa Bossi
Dept. of Computer Science
University Ca' Foscari of Venice
via Torino, 155
30172 Mestre (Venice) Italy
bossi@dsi.unive.it

Ombretta Gaggi
Dept. of Pure and Applied Mathematics
University of Padua
via Trieste, 63
35121 Padua, Italy
gaggi@math.unipd.it

ABSTRACT

In this paper we define a formal semantics for the language SMIL which can be used in a number of applications. First of all, we propose a computer-aided authoring system which include a *Semantic Validator Module* for the evaluation of the temporal consistency of the resulting multimedia presentation. If any temporal conflict is found, the system returns to the user a message pointing out the tag which contains the error and its motivation. This helps the user to correct the error. We also introduce a notion of equivalence for SMIL tags which is useful to find a candidate for substitution in the development of complex multimedia structure, for example in the context adaptation process.

Categories and Subject Descriptors

H.5.4 [**Information Interfaces and Presentation**]: Hypertext/Hypermedia—*Theory*; I.7.2 [**Document and Text Processing**]: Document Preparation—*Standards, Markup Languages*

General Terms

Theory, Verification

Keywords

SMIL, authoring, consistency checking

1. INTRODUCTION

A *multimedia presentation* is the best way to convey information to the user in many advanced applications like distance learning, virtual tourism, news delivery, entertainment and so on. A multimedia presentation is a collection of continuous media, like video or audio files, and static objects, like text pages and images, which can be distributed across the network and rendered to the user according to the author specifications. To be played, media objects must be disposed in the user screen and synchronized according to a time scale; therefore, the author of a presentation must define both the *spatial layout* and the *temporal behavior* of the document.

Multimedia authoring and design is a complex and error-prone activity, especially when the complexity of the temporal structure of multimedia documents increases together with the chance of including a temporal conflict in the synchronization constraints. Many researches address the problem of the specification of a multimedia presentation defining models ([1], [7], [14]), languages and tools ([2], [11], [18]). All the proposed solutions can be divided into two main classes (see [4]): the operational approach, which defines system-dependent structures to model the multimedia presentation, and the constraint-based approach. This second approach is more flexible but it requires the author to have in mind the overall structure of the final presentation which is not represented explicitly. This problem has found a partial solution with the structure of the tags proposed by the Synchronized Multimedia Integration Language, SMIL [9]. In fact, the most part of both the operational and the constraint-based systems use SMIL for building the final presentation.

Since SMIL's first appearance, many authoring tools and players have been implemented, offering to their users different facilities like visual editors or preview windows, and sometimes, tools to check the correctness of the *work-in-progress* multimedia presentation. Unfortunately, most of the tools developed for SMIL check only the *syntactic* correctness of the document before playback, and are not able to find out *semantic* errors.

Motivating Scenario

A semantic error is a *conflict* in the temporal definition of the presentation: there are almost two conflicting values in the definition of the temporal attributes of the document. Usually, syntactic errors can be automatically correct, whereas a semantic conflict points out a contradiction in the definition of the behavior of media items and requires a decision for its solution, which can be built-in in the system [16] or asked to the author. Unfortunately, as described in [10] and [17, 19], in presence of temporal conflicts, even simple multimedia documents may have different behaviors according to the chosen player, therefore the final behavior is almost unpredictable, as reported also by Eidenberger [5].

Let us consider two very common semantic errors: (1) wrong definition of attributes of the same object, e. g., the tag `<text id="txt01" begin="3s" end="5s" dur="5s"/>` defines a text message `txt01` displayed at time instant 3 and removed after 2 time units, but the duration defined

MM'07, September 23–28, 2007, Ausburg, Bavaria, Germany.

by the attribute `dur` is equal to 5 seconds, and (2) wrong definition of the temporal structure of the document, e. g., when the conflicting values involve more than one tag. As an example, the tag

```
<seq dur="5s">
     <img id="img01" dur="5s" />
     <img id="img02" dur="5s" />
 </seq>
```

describes a sequence of two images, each one visualized for 5 seconds. The conflict is between the overall duration of the sequence, i. e., 5 seconds according to its attribute `dur`, and the sum of the durations of the single images it contains.

More common players, e.g. GRiNS [12] or RealPlayer [15], (but not all) do not point out the temporal conflict to the user but the playback goes on and the duration of an object is equal to the minimum duration defined.

This means that, in the first case, text `txt01` lasts for two seconds, and in the second case, image `img02` is not displayed at all and the presentation ends immediately after the first image. Possibly, this is exactly what the author expects but, if not, it is important to individuate the existence of the semantic conflict. Moreover, many players have problems in the resolution of the start and end time of media items, especially when the complexity of the presentation structure increases. Therefore, errors of the second kind are very difficult to find out and fixed, when the complexity of the presentation increases, because it is not always clear if the misbehavior is due to a semantic conflict or to a bug in the player [5].

Some authors, even identifying in the lack of a formal semantics one of the key problems of SMIL language (e. g. Jourdan in [10]), do not consider the example above as *temporal conflicts*, but define a formal semantics which describes the behavior adopted by the majority of the existing players. Others, (e.g. Sampaio et al in [17]) use formal methods to find out temporal conflicts like the ones described in this paper. We agree with this second group of authors thus considering a double inconsistent definition as a semantic error since it points out a contradiction in the description of the result the author of the multimedia presentation tries to obtain. In the examples above, we are not sure that the short interval time of 2 seconds is sufficient to read message `txt01` and the author would not have included `img02` in the presentation if her/his real intention is not to display it.

Summing up, consistency checking is an important issue for multimedia documents and all their applications and any authoring system should consider this aspect to guarantee the generation of a renderable multimedia presentation. We must note here that this paper does not aim at augmenting or correcting the standard SMIL, but at offering a formal semantics which can help guide SMIL developers, thus improving the standard specification: as reported in [10] by Muriel Jourdan, one of the editors of the SMIL 2.0 Timing and Synchronization Module [13], *"... SMIL 2.0 complexity is so great that rejecting the use of formal supports gives rise to a difficult-to-read specification that cannot be free from inconsistency"*.

Our proposal and its practical applications

In this paper we define a formal semantics for the language SMIL 2.1 which is the basis for a *Semantic Validator Module* included into the authoring system depicted in Figure 1. Since an uncorrect multimedia presentation cannot be rendered properly, consistency is checked during all the authoring phase, each time the author asks it or when she/he saves her/his work. We prefer this solution instead of a dynamic checking since we allow temporary inconsistencies due to the *work-in-progress* but we guarantee the correctness of the final result. This choice is also cost-effective. The Semantic Validator Module supports the multimedia authoring process since, if any temporal inconsistency is found, it returns to the user a message containing the tag which contains the error and its motivation; e.g., the tool's message for (1) is *"the text `txt01` contains two conflicting values for the attributes `end` and `dur`"*, and for (2) *"`img02` ends at time instant 10 but its father ends at time instant 5"*. These messages allow the user to easily detect and correct the errors.

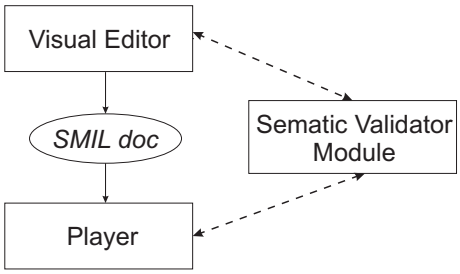

Figure 1: The system architecture

Our semantics is defined by means of a set of inference rules inspired by Hoare logic ([8]). The central feature of Hoare logic is the Hoare triple which describes how the execution of a piece of code changes the state of the computation. This choice brings the advantage that the SMIL structure can be enriched by assertions, expressing the temporal properties, which can be used during the authoring phase, when media items are collected in more complex constructs. As an example, our tool can verify the consistency of a multimedia presentation resulting from a context adaptation process. In this case, the document is dynamically build up by selecting media items compatible with the great number of different situations in which a multimedia presentation can be played, in term of resources availability (e.g., network bandwidth, CPU time), device type (e.g., desktop, laptop, cell-phone) and properties (e.g., screen size, number of colors). This process may often generate conflicts which must be solved in order to guarantee the playback.

Another application of our semantic approach derives from the introduction of a formal notion of equivalence which guarantees that two sets of tags can be substituted each other without changing the behavior of the overall presentation, or generating temporal conflicts. This notion can be used in the context adaptation problem to choose the candidate for a substitution.

As we compose a multimedia presentation by nesting a SMIL tag into another, our rules allow us to compose the semantics by evaluating a single tag inside a more complex nesting. In other words, the proposed semantics is compositional and helps the author to modularize her/his work thus mastering the complexity of the verification of a multimedia presentation consistency.

Despite SMIL 2 specification was first released in 2001, most available players are often unstable or not free of charge as reported in [5]. The major problem is a robust resolution of start and end time of tags. Therefore, another application

SMIL tags	text, img, video, audio, animation, brush par, seq, excl
Attributes	begin, end, dur
Admitted Values	begin: t, accesskey('c')+t, begin(m)+t, end(m)+t end: t, accesskey('c')+t, begin(m)+t, end(m)+t, indefinite dur: t, indefinite

Table 1: List of SMIL tags and attributes used in this work

of our Sematic Validator Module is as the basis for the development of an efficient player for SMIL documents, since it can detect presentations containing conflicts, thus avoiding to start their playback, moreover, if the document is consistent, it generates as output the correct begin and end time of every media item, which can be used for playback.

2. PRELIMINARIES

This paper presents a tentative approach to the formulation of a semantics for the verification of SMIL 2.1 tags using a formal system based on Hoare logic. In this section we start by introducing the basic elements and notations used through the paper.

We note here that our framework currently does not consider the whole SMIL language, even if the values for attributes which are missing is a very limited subset of the one allowed by the standard. Table 1 describes the set of tags, attributes and their possible values addressed in this paper. Note that we do not consider some user interaction like following a link to another document, or stopping the current playback. We plan to fill this gap in our future work.

2.1 The assertion language

In the next section we introduce a set of inference rules which describe how the execution of a piece of SMIL code changes the state of the playback. The rules provide an axiomatic semantics for the temporal aspects of SMIL tags in the spirit of Hoare logic. Therefore they allow us to derive judgements in the form of triplets:

$$\{P\}\ t\ \{Q\}$$

where P and Q are assertions, respectively the *precondition* and the *postcondition*, and t is a SMIL tag. The triple $\{P\}\ t\ \{Q\}$ can be read as: whenever the evaluation of the tag t starts in a state which satisfies the assertion P then it terminates in a state which satisfies the assertion Q.

Since we are interested in describing only those aspects that might influence temporal consistency, a state describes only *significant time instants*: the start and end time instants of all SMIL tags contained in the presentation as well as the duration of each continuous object. Hence the assertion language used to express pre/post conditions includes a set of basic functions representing the significant temporal aspects of the media. Assertions are formed by sets of constraints on values returned by these functions. We say that P holds in a state σ or, equivalently, that σ satisfies P, if all the constraints contained in P are true in σ and we write $P \Rightarrow Q$ if Q holds in any state which satisfies P.

Function	Where	Description
$t_{cr} : Id \rightarrow \mathbf{N}$	Pre	returns the current time instant in which the SMIL tag id is evaluated
$dur : Id \rightarrow \mathbf{N}$	Pre	returns the number of time instants for which a continuous media item plays
$begin : Id \rightarrow \mathbf{N}$	Post	denotes the time instant media item id starts
$end : Id \rightarrow \mathbf{N}$	Post	denotes the time instant media item id ends
Notation	Description	
$begin_B(c)$	returns t if $\{begin(c) = t\} \subseteq B$	
$end_B(c)$	returns t if $\{end(c) = t\} \subseteq B$	

Table 2: List of functions and notations used in the definition of the proof rules

Table 2 lists all the functions used in the assertions. This set includes functions whose values can be obtained by analyzing the SMIL document which implicitly holds all the information about the temporal disposition of the objects they contain, i.e. the begin and end time instants of each item. They are $begin : Id \rightarrow \mathbf{N}$ and $end : Id \rightarrow \mathbf{N}$, which denote respectively, the time instant the tag denoted by the identifier $id \in Id$ starts or ends. For instance the constraint $begin(id) = 3$ states that SMIL tag id starts its rendering at time instant 3. Given an assertion B which contains the equality $begin(c) = t$ (or $end(c) = t$) we use also the notation $begin_B(c)$ (or $end_B(c)$) to denote the time instant t occurring in the corresponding equality.

The only piece of useful information which is not contained in the SMIL document is the natural duration of each continuous media. We define the function $dur : Id \rightarrow \mathbf{N}$ which takes as input an identifier of a continuous media ($id \in Id$) and returns the number of time instants for which the continuous media item plays in absence of user interaction or other temporal specification.

Our assertions also contain the function $t_{cr} : Id \rightarrow \mathbf{N}$ that returns the current time instant in which the SMIL tag id is evaluated. By *current time instant of a tag id* we mean the time instant in which, considering a player executing the presentation, the player evaluates that command. As an example, if more than one media is defined inside a tag `par`, they are evaluated at the same time instant, because they will be played in parallel (unless other attributes specification). Otherwise, in case of sequential composition, the player considers each media item one after the other, therefore the current time instant of each media is the end time of the previous one. This function is used to indicate which is the first tag of the document, i. e. the tag t with $t_{cr}(t) = 0$, or to evaluate a single tag if we want to check the correctness of only a portion of a presentation.

As a general remark, in the triple $\{P\}t\{Q\}$ the precondition P contains, among others, the current time instant of the tag t and the natural duration of media items which it defines (if applicable). The postcondition Q contains, among

Abbreviations
media ::= cont | static;
cont ::= video | audio | animation;
static ::= text | img | brush;
cmd ::= media | par | seq | excl;
m ::= id="m"
acsk('c') = accesskey('c')

Table 3: List of abbreviations used in the definition of the proof rules

Name	Description
$finite(k)$	holds if k is a real value
$indefinite(k)$	holds if k is equal to ``indefinite''
$defined(k)$	holds if k is not equal to ``void'', i. e., $finite(k) \vee indefinite(k)$ holds
$NotDur$	contains all the statements that have their attributes `dur` and `end` equal to `void`
$Closure(c)$	contains all the statements defined inside the tag c, at any level of nesting
$Indef(c)$	holds if in $Closure(c)$ there are tags with attribute `end` (or `dur`) equal to ``indefinite''

Table 4: List of predicates and sets used in the definition of the proof rules

others, the definition of the time instants in which the tags defined in t begin and/or end. Media items definitions are evaluated through axioms, while for `par` and `seq` composition more complex rules are needed.

2.2 Notational conventions

In the following section we use a number of special notational conventions to introduce the set of inference rules describing the semantics of the SMIL tags.

Table 3 lists a set of abbreviations used for the representation of the SMIL tags. For instance `<cmd c>` stands for any tag SMIL with the attribute `id = ''c''`. Moreover, we use the general form `end=''k''` and `dur=''k''` to represent the attributes of a tag where the meta-variable `k` is either any of the admitted values for the particular attribute, or the special value `void`. The value `void` represents the absence of that attribute and allows us to define only one rule for each compound tag. As regards the attribute `begin` we assume it is always defined since its absence can be represented by the value `k=''0''`. For instance, `<video id=''v''begin=''0''dur=''5''end=''void''/>` is considered as a synonymous of `<video id=''v''dur=''5''/>`.

The advantage of this representation is that of avoiding repetition of very similar rules, but we need a set of predicates to check the existence of an attribute's value before using it. We need also to classify the tags which occur in a SMIL document with respect to the values of their attributes `dur` and `end`. Hence we introduce some auxiliary predicates and sets whose description can be found in Table 4.

STATIC+BEGIN

$\{A \cup Pre\}$ `<static m begin=''k1''/>` $\{A \cup Post\}$

where $Pre = \{t_{cr}(m) = start - k1\}$, $Post = \{begin(m) = start\}$

CONT+BEGIN

$\{A \cup Pre\}$ `<cont m begin=''k1''/>` $\{A \cup Post\}$

where $Pre = \{t_{cr}(m) = start - k1, dur(m) = stop - start\}$, $Post = \{begin(m) = start, end(m) = stop\}$

MEDIA+BEGIN+END+DUR

$\{A \cup Pre\}$

`<media m begin='k1' end='k2' dur='k3'/>` $\{A \cup Post\}$

where

$Pre = \{t_{cr}(m) = start - k1\}$

$Post = \{begin(m) = start\} \cup End$

$$End = \begin{cases} \{end(m) = start - k1 + k2\} & \text{if } finite(k2) \\ \{end(m) = start + k3\} & \text{if } finite(k3) \\ \emptyset & \text{otherwise} \end{cases}$$

APPLICABILITY CONDITION:

$(defined(k2) \vee defined(k3))$
$\wedge\ (indefinite(k2) \iff indefinite(k3))$
$\wedge\ ((finite(k2) \wedge finite(k3)) \implies k3 = k2 - k1)$

Table 5: Proof rules for media items definitions

3. A SEMANTICS FOR SMIL TAGS

SMIL language definition provided by [9] does not contain a formal specification of tags and attributes semantics. The recommendation is divided into sections, some of which are defined "normative". Sometimes, an algorithm is provided to better explain how significant time instants are computed, but neither a formal definition nor verification tools have been implemented by Synchronized Multimedia Working group of W3C to check the sematic correctness of SMIL tags.

In this section, we define a formal system which is able to find out temporal conflicts of a multimedia presentation defined using SMIL. The system provides a Hoare logic for SMIL by a set of inference rules describing how the execution of a piece of code changes the state of the playback.

We start by considering self contained tags, i. e., SMIL commands whose synchronization do not refer to other media items or tags. Axioms to verify the correctness of statements which define media items are listed in Table 5. The use of events is discussed in Section 3.2.

Assume we want to verify the triple:

$$\{P\}\texttt{<video id=''v'' begin=''2''/>}\{Q\}$$

where the precondition P is $\{dur(v) = 5, t_{cr}(v) = 0\}$ and the postcondition Q is $\{begin(v) = 2, end(v) = 7\}$. The system verifies its correctness since, by applying the axiom CONT+BEGIN, it obtains the set of constraints $\{t_{cr}(v) = 2-2, dur(v) = 7-2\}$ which is equivalent to the precondition.

The system can also be used as the basis for the implementation of a player. In this case, it applies the axiom

CONT+BEGIN starting from the precondition, obtaining the postcondition that can be used to start and stop the video.

The situation is a little more complicated if the media definition contains also an `end`, or `dur`, attribute. The rule MEDIA+BEGIN+END+DUR defines the end time of a media item m only if both $k2$ and $k3$ are finite (if defined), i. e., they are not equal to "`indefinite`". As an example, we can apply the rule to verify that the triple

$$\{P\}\texttt{<video id=``v'' begin=``2'' end=``3''/>}\{Q\}$$

where $\{Q\} = \{begin(v) = 2, end(v) = 3\}$ is valid. As discussed in Section 2.2, `<video id='v'begin='2'end='3'/>` is a synonymous of `<video id='v'begin='2'dur='void' end='3'/>`. For readability sake, we will use always this form in the following.

Media items definition does not lead to temporal conflicts unless the author defines both the `dur` and the `end` attributes. The applicability condition disallows the application of the rule in presence of uncorrect values of these attributes; e. g. when both the attributes `dur` and `end` are finite, the relation $k3 = k2 - k1$ must hold. The applicability conditions also point out a temporal conflict to the user.

3.1 Rules for more complex constructs

When media definitions are nested into parallel and sequential composition, the evaluation of these structures requires the definition of more complex rules.

Since the flexibility of SMIL tags allows us to describe the same temporal behavior using both a `par` or a `seq` tag, we base the discussion of this section mainly on the description of the rules for the parallel composition. The sequential composition is discussed at the end of this section.

PAR+BEGIN+END

$$\frac{\{A_i \cup \{t_{cr}(c_i) = init_{c_i}\}\}\ c_i\ \{B'_i\} \qquad \forall i\ 1 \le i \le n}{\{A'\}\ \texttt{<par c attribute-list>}\ c_1 \ldots\ c_n\ \texttt{</par>}\ \{B\}}$$

where

`attribute-list` $\equiv$ `begin='k1' end='k2' dur='void'`

and

$$A' = \bigcup_{i=1}^{n} A_i \cup \{t_{cr}(c) = init\}$$

$$init_{c_i} = init + k1$$

$$B = \bigcup_{i=1}^{n} B_i \cup \{begin(c) = init + k1\} \cup End$$

$$stop = \begin{cases} init + k2 & \text{if } finite(k2) \\ max_{c_i}\ \{end_{B_i}(c_i)\} & \text{if } \neg defined(k2) \end{cases}$$

$$End = \begin{cases} \{end(c) = stop\} & \text{if } \neg Indef(c) \\ \emptyset & \text{otherwise} \end{cases}$$

$$B'_i = \begin{cases} B_i \setminus \{end(c_i) = stop\} & \text{if } c_i \in NotDur \\ & \quad \wedge finite(k2) \\ B_i & \text{otherwise} \end{cases}$$

APPLICABILITY CONDITION:

$$finite(k2) \implies \neg Indef(c)$$
$$\wedge\ Indef(c) \implies (\neg defined(k2) \vee indefinite(k2))$$
$$\wedge\ finite(k2) \implies \forall c_i\ end_B(c_i) \le stop \vee c_i \in NotDur$$
$$\wedge\ \forall c_i\ begin_B(c_i) \ge init + k1$$

Table 6: Proof rule for the parallel composition when the attribute `dur` is equal to `void`

We start our analysis by considering the parallel composition expressed by the tag `par` when the attribute `dur` is not present (i. e. `dur=``void''`), the attribute `begin` is always present (possibly equal to zero) and the attribute `end` is `void`, `indefinite` or a real number. The PAR+BEGIN+END rule described in Table 6 defines the semantics of the parallel composition in these cases. In the postcondition we make the components $B_1 \ldots B_n$ evident to make it explicit that the postcondition should contain information about each c_i, be it a media object or a synchronization structure.

To prove the correctness of the tag `<par c>` $c_1 \ldots\ c_n$ `</par>`, each c_i must be proven to be correct by assuming as its current time instant the current time instant of the parallel tag plus the offset given by the attribute `begin`, i. e., if $(t_{cr}(c) = init)$ is contained into the precondition of the tag c, the precondition of each tag c_i must contain $(t_{cr}(c_i) = init + k1)$ where $k1 \ge 0$ is the value of the attribute `begin` and $init$ is the time instant at which the statement `par` is evaluated.

The evaluation of the end time instant of a par tag is a little more complicated, and not always possible. As a general remark, it is not possible to calculate the end time of a media item in two cases: if it is a static object and it does not have an attribute `end` or `dur` defined, or if it has an attribute `end` or `dur` equal to `''indefinite''`. In the same way, the ending time of a `par` statement cannot be calculated if its attribute `end` (or `dur`) is equal to `''indefinite''`, or it is not defined and one of it its children has the attribute `end` (or `dur`) equal to `''indefinite''`.

Once we are able to decide whether a parallel composition terminates, we must calculate the time instant *stop*. The semantics which describes the evaluation of *stop* is complex since different cases have to be considered. We discuss here what happens when an `end` attribute is defined: the case relating the `dur` attribute is very similar and will be discussed in the following.

We have to study four possible situations:

1. the tag c does not contain the definition of attribute `end` (i. e. `end = ''void''`: in this case, the statement c ends when all its children (which are not static objects in $NotDur$) have finished their playbacks, i.e. at time instant $stop = max_{c_i}\ \{end_{B_i}(c_i)\}$;
2. all statements contained in the `par` tag end up before the `par` statement's end, more precisely before time instant $init + k2$;
3. some continuous media items defined inside c have a natural duration wider than the duration of c;
4. some items defined inside c have a duration, defined with an attribute `dur` or `end`, wider than the duration of c.

Cases 1, 2 and 3 are all correct. In the first two cases, each media object or statement within c lasts for a period of time equal or shorter than the duration of c. If a static media item has not a duration defined, its duration is equal to the duration of c. In case 3, if a continuous media c_i has a natural duration longer than the duration of c, its playback will be truncated at c's end.

Case 4 is not correct since the author gives a double, and contradictory, definition of the duration of media items involved, thus generating a temporal conflict. Note that case 4 includes also the case in which the parallel composition has a finite duration, but contains some children with an

indefinite duration, which is, by definition, longer than any other finite value.

We can apply the PAR+BEGIN+END rule in cases 1, 2 and 3, since the applicability conditions are satisfied. In case 1, all media items end before time instant *stop* because it is chosen as the maximum value. In case 2, all media items end before $init + k2 = stop$ from the hypothesis. In case 3 all media items ending after the time instant *stop* belong to $NotDur$, therefore $finite(k2) \implies \forall c_i\ end_B(c_i) \leq stop \vee c_i \in NotDur$, hence the applicability condition is satisfied and the rule can be applied. The same applicability condition prevents us to apply the PAR+BEGIN+END rule in case 4 when a statement c_i has a finite duration longer than c.

The statement $finite(k2) \implies \neg Indef(c)$ states that in presence of a finite value of $k2$, the rule can be applied to the statement c only if it ends, i.e., it does not contain, at any level of nesting, an item with an indefinite duration.

Otherwise, the applicability condition $Indef(c) \implies (\neg defined(k2) \vee indefinite(k2))$ states that if the statement does not end and the attribute `end` is defined, then it must be equal to `''indefinite''`. Finally the condition $\forall c_i\ begin_B(c_i) \geq init + k1$ expresses the fact that all children of c must start together with c or after it.

Let us illustrate how our rules find out temporal conflicts like the one described in case 4, due to an author's error which can happen when the structure becomes more complex, including a lot of tags nested one into the other. Let us consider the following tag:

```
<par id="p" begin="0" end="5s">
    <img id="i" begin="0" end="5s" />
    <text id="tx" begin="0" end="7s" />
</par>
```

Even if the temporal conflict is evident since the tag is simple, (text page `tx` lasts more then the tag in which it is contained), we try to check the semantic correctness of this statement to show how the system works.

We would like to prove that

$$\{t_{cr}(p) = 0\}\texttt{<par p ...>}\{Q\}$$

where $Q \equiv \{begin(i) = 0, end(i) \leq 5, begin(tx) = 0, end(tx) \leq 5, begin(p) = 0, end(p) = 5\}$ but statement `p` is not correct since rule PAR+BEGIN+END (see Table 6) cannot be applied. In fact, since both `tx` and `i` do not belong to the set $NotDur$, in order to apply the rule we would have to prove the premises:

$$\begin{array}{ll} S_i & \equiv \{t_{cr}(i) = 0\}\ \texttt{i}\ \{begin(i) = 0, end(i) = 5\} \\ S_{tx} & \equiv \{t_{cr}(tx) = 0\}\ \texttt{tx}\ \{begin(tx) = 0, end(tx) = 5\} \end{array}$$

The first triple S_i is valid and we can prove it by the axiom MEDIA+BEGIN+END+DUR, but we cannot prove the triple S_{tx} which is not valid. Therefore the PAR+BEGIN+END rule cannot be applied since the premise S_{tx} cannot be verified. In this case, the answer of our tool is that the presentation contains a semantic conflict since media item `tx` ends at time instant 7 while its father ends at time instant 5.

The rule which describes the semantics of the sequential composition is very similar to the PAR+BEGIN+END rule since the two tags can express the same synchronization if the values of the attributes are properly defined. It is presented in Table 7. Also in this case, we consider the attribute `dur` equal to `void`.

SEQ+BEGIN+END

$$\frac{\{A_i \cup \{t_{cr}(c_i) = init_{c_i}\}\}\ c_i\ \{B'_i\} \qquad \forall i\ 1 \leq i \leq n}{\{A'\}\ \texttt{<seq c attribute-list>}\ c_1 \ldots\ c_n\ \texttt{</seq>}\ \{B\}}$$

where

`attribute-list` $\equiv$ `begin='k1' end='k2' dur='void'`

and

$$A' = \bigcup_{i=1}^{n} A_i \cup \{t_{cr}(c) = init\}$$

$$init_{c_i} = \begin{cases} init + k1 & \text{if } i = 1 \\ end_B(c_{i-1}) & \text{if } i > 1 \end{cases}$$

$$B = \bigcup_{i=1}^{n} B_i \cup \{begin(c) = init + k1\} \cup End$$

$$stop = \begin{cases} init + k2 & \text{if } finite(k2) \\ max_{c_i}\{end_{B_i}(c_i)\} & \text{if } \neg defined(k2) \end{cases}$$

$$End = \begin{cases} \{end(c) = stop\} & \text{if } \neg Indef(c) \\ \emptyset & \text{otherwise} \end{cases}$$

$$B'_i = \begin{cases} B_i \setminus \{end(c_i) = h\} & \text{if } (begin_B(c_i) = h) \\ B_i & \text{otherwise} \end{cases}$$

APPLICABILITY CONDITION:

$finite(k2) \implies \neg Indef(c)$
$\wedge\ Indef(c) \implies (\neg defined(k2) \vee indefinite(k2))$
$\wedge\ finite(k2) \implies \forall c_i\ end_B(c_i) \leq stop \vee c_i \in NotDur$
$\wedge\ \forall c_i\ begin_B(c_i) \geq init + k1$

Table 7: Proof rule for the sequential composition when the attribute `dur` is equal to `void`

With respect to the parallel composition there are only two differences: first, the current time instant of each child is equal to the end time instant of the previous child, and not to the current time instant of the `seq` tag. Second, the `seq` statement imposes a duration equal to zero to static media items which have not a defined duration, i. e., $begin_B(c_i) = h$ and $end_B(c_i) = h$ if c_i is a static media contained in $NotDur$. This means that they are never played in the user screen.

So far we consider only the use of the attribute `end`, but, as already discussed for media item definition, statements can also contain an attribute `dur` whose semantics is very similar to the `end` attribute and therefore, an easily translation can be obtained with the rule CMD+BEGIN+END+DUR illustrated in Table 8.

CMD+BEGIN+END+DUR

$$\frac{\{A\}\ \texttt{<cmd c attribute-list>}\ c_1 \ldots\ c_n\ \texttt{</cmd>}\ \{B\}}{\{A\}\ \texttt{<cmd c attribute-list2>}\ c_1 \ldots\ c_n\ \texttt{</cmd>}\ \{B\}}$$

where

`attribute-list` $\equiv$ `begin='k1' end='k2' dur='void'`

and

`attribute-list2` $\equiv$ `begin='k1' end='k4' dur ='k3'`

APPLICABILITY CONDITION:

$defined(k3)$
$\wedge\ (indefinite(k2) \iff indefinite(k3))$
$\wedge\ finite(k3) \implies (finite(k2) \wedge k3 = k2 - k1)$
$\wedge\ defined(k4) \implies k4 = k2$

Table 8: Proof rules for a general composition of tags when the attribute `dur` is defined

3.2 Particular values for `begin` and `end`

The discussion so far consider only a time value (e.g. a number of seconds) as possible value for the attribute `begin` and `end`. SMIL language permits also other ways to define the starting or the ending time of tags using events (see Table 1). Let us consider the case in which the start (or the end) of a media, or a group of media items, occurs when the user keys in a character, say 's', in the keyboard as described by the following tag:

<cmd c begin=''accesskey(s)+k''/>

where `accesskey(s)` means that the user has to key in the character 's' and $k \geq 0$ represents a number of seconds.

The correctness of this statement can be proven only if we already know the instant in which the event $accesskey(s)$ takes place. Therefore, to define a rule for this kind of statements, we need a function $t : \mathcal{A} \rightarrow \mathcal{N}$ which takes as input an event $accesskey(s) \in \mathcal{A}$ and returns the time instant in which this event takes place; this function can be used in the preconditions of the statement to constraint the input event.

BEGIN+ACCESSKEY

$$\frac{\{A\}\texttt{<cmd c begin='keyin+k' end='value'/>}\{B\}}{\{A'\}\texttt{<cmd c begin='acsk(s)+k' end='value'/>}\{B\}}$$

where $A' = A \cup \{t(acsk(s)) = keyin\}$

APPLICABILITY CONDITION:

$A' \implies \{t(acsk(s)) \geq t_{cr}(c)\}$

END+ACCESSKEY

$$\frac{\{A\}\texttt{<cmd c begin='value' end='keyin+k'/>}\{B\}}{\{A'\}\texttt{<cmd c begin='value'end='acsk(s)+k'/>}\{B\}}$$

where $A' = A \cup \{t(acsk(s)) = keyin\}$

APPLICABILITY CONDITION:

$A' \cup B \implies \{t(acsk(s)) \geq begin(c)\}$

Table 9: Proof rules for SMIL statements with an `accesskey` in the definition of the `begin` or the `end` attribute

Table 9 shows the rules to deal with statements with a `begin` or an `end` attribute which is bound to an input from the keyboard. Due to space constraints, $acsk(s)$ is used instead of $accesskey(s)$. Moreover, we write `begin=''value''` or `end=''value''` where `value` can assume any of the admitted values listed in Table 1.

These rules simply state that the input from the keyboard must occur after the evaluation of the statement, represented by $t_{cr}(c)$, or after its beginning if `accesskey` is defined in the `end` attribute of a statement for which also a `begin` attribute is defined. Once this hypothesis is verified, since $(t(accesskey(s)) = keyin) \in A'$, the triple

$$\{A'\}\texttt{<cmd c end=''accesskey(s)+k''/>}\{B\}$$

holds whenever we can prove that

$$\{A\}\texttt{<cmd c end=''keyin+k''/>}\{B\}.$$

Note that almost all other events could be addressed in the same way as soon as we assume the existence of a suitable function recording the time instant in which the event occurs. As an example, the `activateEvent` represents the time instant in which an user clicks on a media items, and therefore, from our point of view, it is not different from the user clicking on the keyboard.

Another possibility offered by the SMIL standard is to bind the begin (or end) event of a (group of) media item m with the begin (or end) event of another (group of) media item n. As an example, consider the tags

```
<par id="p" end="au.end">
    <audio id="au" />
    <text id="tx" />                    (1)
</par>

<cmd id="m" begin="n.begin+5"/>         (2)
```

in case (1), the whole `par` statement ends when media item au ends; in case (2) media item m begins 5 seconds after the beginning of n.

Tag (1) can be considered similarly to the `accesskey` case: if we already know (from the premise) the end point of au, i. e. $end(au) = stop$, we can then analyze the tag `<par id="p" end="stop">...</par>`.

Therefore the following rule can be applied to tag (1).

PAR+END+EVENT

$$\frac{\{A\}\texttt{<par c end='stop+k'>}c_1..c_n\texttt{</par>}\{B\}}{\{A\}\texttt{<par c end='ci.end+k'>}c_1..c_n\texttt{</par>}\{B\}}$$

APPLICABILITY CONDITION:

$end_B(c_i) = \texttt{stop}$

As we have already said, situation is more complex in case (2), which cannot be analyzed singularly since its evaluation needs information about the begin of media item n. For this reason we must consider a set of media items as shown by the following rule:

BEGIN+EVENT

$$\frac{\{A\}\texttt{<cmd c>}\ldots c_i \ldots c'_j \ldots\texttt{</cmd>}\{B\}}{\{A\}\texttt{<cmd c>}\ldots c_i \ldots c_j \ldots\texttt{</cmd>}\{B\}}$$

where $\texttt{n} \in Closure(c_i)$, $d \equiv$ `<cmd id="m" begin="n.begin +k"/>` $\in Closure(c_j)$, and c'_j is obtained from c_j by replacing d with $d' \equiv$ `<cmd m begin="begin`$_B$`(n)+k" />`.

In this paper we consider a limited set of combinations of attributes and events, but the approach is easily generalizable to cover all the possibilities offered by the standard.

3.3 The `excl` tag

SMIL language provides also a tag for the *exclusive* composition of media items, i. e., the tag `excl`, whose semantics states that only one of its children is active at any given time instant. Therefore, this tag is very similar to the sequential composition, even in this case only one child is active at a time, but `excl` does not impose any order in the visualization of the children. This means that each child may contain the attribute `begin` in the definition, or may be activated by the user, e.g. following a link. Let us consider the following example:

```
<par>
    <img id = "a" />  <img id = "b" /> <img id = "c" />
    <excl id="e" dur="10">
        <video id = "video_a" begin="a.activateEvent"/>
        <video id = "video_b" begin="b.activateEvent"/>
        <video id = "video_c" begin="c.activateEvent"/>
    </excl>
</par>
```

in this case, the user chooses a video clip by clicking on an image button chosen between media items `a`, `b` and `c`.

The corresponding video is activated by the proper *activateEvent*. The `excl` tag simply states that only one video clip plays at a time: in fact, the video currently playing is stopped when the user clicks on another image, choosing another video clip.

The example shows how the `excl` command does not deal with the activation of its children but with their deactivation; in fact, the playback order of the video clips completely depends on the user choices and not on the tags' definitions.

The semantics of the `excl` tag is described in Table 10 by rule EXCL+BEGIN+END. Like tags `par` and `seq`, the `excl` tag begins at its current time instant, or after $k1$ time instants if the attribute `begin` is finite, and ends, when there are no children playing. This means that, it can have an instantaneous duration if no child starts together with it. For this reason, the attribute `end` of this statement usually does not contain the special value `''void''`.

The EXCL+BEGIN+END rule is very similar to the rule which describes the semantics of parallel composition, therefore we do not repeat here the problem of the termination of the tag. Even in this case, to prove the correctness of the statement `<excl>` $c_1 \ldots c_n$ `</excl>`, each c_i must be proven to be correct, assuming as its current time instant the current time instant of its father. Since the exclusive tag may impose a premature stop of the playback of its children, in some cases, we do not require to know the time instant in which the child c_i ends in the premises, i. e.:

1. when c_i ends together with `excl` (i.e. `k2` is finite) if it does not contain the attribute `end` or `dur` in its definition (i.e $c_i \in NotDur$);
2. when the playback of c_i is stopped before its natural termination due to the user interaction or some other external event (i. e. $\exists j \ |begin_B(c_j) = t_i$).

The applicability condition prevents the application of the rule in presence of temporal conflicts. Among the conditions already discussed for the parallel composition, the condition $\forall c_i, c_j \ (begin_B(c_i) \leq begin_B(c_j)) \implies (end_B(c_i) \leq begin_B(c_j))$ states that only one child plays at any given time instant, i. e., if child c_i begins before child c_j, it also ends before c_j's beginning.

4. EQUIVALENCE OF SMIL TAGS

In this section we introduce two notions of equivalence between SMIL tags: an *observable equivalence*, $\approx$, and a *strong equivalence*, $\approx_S$. The first one formalizes the informal idea that two tags are equivalent if a user cannot distinguish them just by observing the playback of the various media items in her/his screen. The second one is stronger because it specifies when two tags can be substituted each other. For this reason it also asks the same time reference for the two tags, i. e., the same value for the function t_{cr}.

First we introduce an equivalence on assertions. We say that two assertion P_1 and P_2 are player_equivalent, and write $P_1 \sim_m P_2$, if they contain the same constraints on media items. More formally:

Definition. Let P_1 and P_2 be assertions. $P_1 \sim_m P_2$ if for every function $f \in \{dur, begin, end\}$ and media m:

$$(f(m) = k) \in P_1 \Leftrightarrow (f(m) = k) \in P_2.$$

Then we can introduce the notion of observable equivalence between tags.

EXCL+BEGIN+END

$$\frac{\{A \cup \{t_{cr}(c_i) = init_{c_i}\}\}\ c_i\ \{B'_i\} \qquad \forall i\ 1 \leq i \leq n}{\{A'\}\ \texttt{<excl c attribute-list>}\ c_1 \ldots\ c_n\ \texttt{</excl>}\ \{B\}}$$

where

`attribute-list = begin='k1' end='k2'dur='void'`

$$A' = \bigcup_{i=1}^{n} A_i \cup \{t_{cr}(c) = init\}$$

$$init_{c_i} = init + k1$$

$$B = \bigcup_{i=1}^{n} B_i \cup \{begin(c) = init + k1\} \cup End$$

$$stop = \begin{cases} init + k2 & \text{if } finite(k2) \\ max_{c_i}\{end_{B_i}(c_i)\} & \text{if } \neg defined(k2) \end{cases}$$

$$End = \begin{cases} \{end(c) = stop\} & \text{if } \neg Indef(c) \\ \emptyset & \text{otherwise} \end{cases}$$

$$B'_i = \begin{cases} B_i \setminus \{end(c_i) = t_i\} & \text{if } \exists j\ begin_B(c_j) = t_i \\ & \vee(finite(k2) \\ & \wedge c_i \in NotDur) \\ B_i & \text{otherwise} \end{cases}$$

APPLICABILITY CONDITION:

$$finite(k2) \implies \neg Indef(c)$$
$$\wedge\ Indef(c) \implies (\neg defined(k2) \vee indefinite(k2))$$
$$\wedge\ finite(k2) \implies \forall c_i\ (end_B(c_i) \leq stop \vee c_i \in NotDur)$$
$$\wedge\ \forall c_i\ begin_B(c_i) \geq init + k1$$
$$\wedge\ \forall c_i, c_j\ (begin_B(c_i) \leq begin_B(c_j)) \implies (end_B(c_i) \leq begin_B(c_j))$$

Table 10: Proof rule for the exclusive composition when the attribute `dur` is equal to `void`

Definition (Observable Equivalence). Let c_1 and c_2 be SMIL tags. Then $c_1 \approx c_2$ if

- for any pair P_1, Q_1 such that $\vdash \{P_1\}c_1\{Q_1\}$ there exist P_2, Q_2 such that $P_1 \sim_m P_2$, $Q_1 \sim_m Q_2$ and $\vdash \{P_2\}c_2\{Q_2\}$;
- for any pair P_2, Q_2 such that $\vdash \{P_2\}c_2\{Q_2\}$ there exist P_1, Q_1 such that $P_1 \sim_m P_2$, $Q_1 \sim_m Q_2$ and $\vdash \{P_1\}c_1\{Q_1\}$,

As an example, consider the following commands:

$c_1 \equiv$`<img id="im2" begin="10" dur="5s"/>`

$c_2 \equiv$`<img id="im2" dur="5s"/>`.

We can prove $\{P_1\}c_1\{Q_1\}$ only if $P_1 = A \cup \{t_{cur}(im2) = start - 10\}$ and $Q_1 = A \cup \{begin(im2) = start, end(im2) = start + 5\}$, for some assertion A; similarly, we can prove $\{P_2\}c_2\{Q_2\}$ only if $P_2 \equiv A \cup \{t_{cur}(im2) = start\}$, $Q_2 \equiv A \cup \{begin(im2) = start, end(im2) = start + 5\}$, for some assertion A. Hence c_1 and c_2 are observable equivalent.

The notion of observable equivalence correctly captures the behavior of the overall execution of a SMIL tag but it is not sufficiently strong to induce a substitution property. Consider for instance the following tags:

```
d1 = <seq id="s"  >
       <img id="im1" dur="7s" />
       <img id="im2" dur="5s" />
     </seq>

d2 = <seq id="s"  >
       <img id="im1" dur="7s" />
       <img id="im2" begin="10s" dur="5s" />
     </seq>
```

They are clearly not equivalent even if they differ only for the replacement of the tag c_1 with the observable equivalent tag c_2.

The problem depends on the fact that the notion of observable equivalence does not impose a fixed "starting" time instant for a SMIL tag, that depends on the time in which it is analyzed. The function t_{cr} sets the "starting" time of a SMIL tag when it is nested into another tag. Therefore a tag can be replaced with another one, only if their t_{cr}-values are equal.

Hence we need a stronger notion of equivalence if we want to prove a substitution property.

Definition (Strong Equivalence). Two tags c_1 and c_2 are *strong* equivalent, $c_1 \approx_S c_2$, if they are observable equivalent, $c_1 \approx c_2$, and they are "time coherent":
$(t_{cr}(c_1) = init) \in P_1 \Leftrightarrow (t_{cr}(c_2) = init) \in P_2$ and for all $f \in \{begin, end\}$ $(f(c_1) = k) \in P_1 \Leftrightarrow (f(c_2) = k) \in P_2$.

Note that, in the definition of strong equivalence the condition, for all $f \in \{begin, end\}$, $(f(c_1) = k) \in P_1 \Leftrightarrow (f(c_2) = k) \in P_2$ seems to be pointless since $Q_1 \sim_m Q_2$. This is true when tags c_1 and c_2 start and end with a media item. This is not true when there are time intervals in which no media items are played at the beginning or ending of the tag, or when $\neg Indef(c_1) \wedge \neg Indef(c_2)$. Consider as an example the tag:

```
<seq id="s" begin="10" >
       <img id="im1" dur="10s" />
       <img id="im2" dur="5s" />
</seq>
```

where no media item plays for the first ten seconds. In this case, the condition $\{begin(c_1) = bc, end(c_1) = ec\} \subseteq Q_1 \Leftrightarrow \{begin(c_2) = bc, end(c_2) = ec\} \subseteq Q_2$ must hold to find out a strong equivalent tag.

We can now prove the following theorem.

Theorem. Let d be a SMIL tag in which the sub-tag c occurs, nested at some level inside d, i. e., $c \in Closure(d)$, and d' be the tag obtained from d by replacing the sub-tag c with c'. Then:

$$c \approx_S c' \implies d \approx_S d'.$$

Proof. The proof of the theorem follows by induction on the depth of the nesting of c in d. Let us assume there exists a proof for $\{A\}d\{B\}$.

Base case: $d = c$. It is trivial.

Inductive step: $c \in Closure(d_i)$, and d_i is one of the children of d.

All the rules of our proof system have a similar shape: if the conclusion is $\{A\}d\{B\}$ and d_i is one of the children of d, then in the premises we find $\{A_i \cup \{t_{cr}(d_i) = init_i\}\}\ d_i\ \{B_i\}$. Moreover, $A = \bar{A} \cup A_i \cup \{t_{cr}(d) = init\}$ and $B = \bar{B} \cup B_i \cup \{begin(d) = begin\} \cup End$.

Let d_i' be obtained from d_i by replacing the sub-tag c with c'. By inductive hypothesis $d_i \approx_S d_i'$. Hence there exist A_i' and B_i' such that $\vdash \{A_i' \cup \{t_{cr}(d_i') = init_i\}\}\ d_i'\ \{B_i'\}$ where $A_i' \sim_m A_i$, $B_i' \sim_m B_i$ and $\{begin(d_i') = start_i\} \subset B_1'$ if and only if $\{begin(d_i) = start_i\} \subseteq B_i$ and $\{end(d_i') = stop_i\} \subseteq B_i'$ if and only if $\{end(d_i) = stop_i\} \subseteq B_i$.

We can substitute this premise in the proof of $\{A\}d\{B\}$, thus obtaining a proof for $\vdash \{A'\}d'\{B'\}$ where $A' = \bar{A} \cup A_i' \cup \{t_{cr}(d') = init\}$ and $B' = \bar{B} \cup B_i' \cup \{begin(d') = begin\} \cup End$. Hence $d \approx_S d'$.

5. CONCLUSIONS AND RELATED WORK

This paper presents a formal semantics for the temporal aspects of SMIL documents. We start by defining a set of inference rules inspired by the Hoare triples which describe how the execution of a piece of code changes the state of the computation. A prototype implementation of a computer-aided authoring system, including the *Semantic Validator Module* based on the proposed semantics, is currently under development.

This paper mainly focuses on SMIL 2.1 features but since SMIL 3.0 specification leaves the basic syntax and semantics of the SMIL 2.1 timing model unchanged [3], it also applies to the latest version.

The main advantages of this work are the following:

- it allows the author to check the consistency of a multimedia presentation based on the SMIL standard and not on a proprietary format;
- it assists multimedia authoring by pointing out conflicting values in the document;
- it allows for a modular evaluation of the tags nested in a SMIL document and helps the context adaption process;
- it minimizes the set of preconditions needed to evaluate a SMIL tag, i. e., the natural duration of the continuous objects, if present, and
- the compositionality of the approach allows for an easy extension of SMIL features actually considered.

Note here that all the rules of the proof system can be used both for a top-down construction of a correct playback sequence of the media items involved in the multimedia presentation and for a bottom-up analysis of the SMIL document. This second feature is particularly useful during the context adaption of a document to find out a suitable candidate for substitution or, more in general, during the authoring of the document by composition of tags. Moreover, it is useful to find out the weakest precondition, i.e., the minimal set of requirements needed to evaluate a tag. In our system this set contains the natural duration of the continuous media items and the current time instant of the outer-most tag, equal to zero by standard convention.

We note also that a composition of a SMIL document driven by the rules is correct by construction. The analysis of a tag finds out a temporal conflict, if the construction of the proof fails because one of the needed premises cannot be proved or the applicability conditions are not satisfied. In this case, the prototype system returns a message which shows the inconsistent values, e.g., a child which ends after its father. The compositionality of our approach helps the user to correct the error and to incrementally continue the analysis.

The choice of Hoare logic as basis for the formalism allows us to incrementally extend the subset of SMIL features implemented. New features are added by defining new rules to describe the semantics of a particular tag or attribute, or by defining a translation to a more simple situation, e.g. the CMD+BEGIN+END+DUR translates a tag containing all the attributes `begin`, `end` and `dur` into an equivalent tag without the attribute `dur`. This second approach can be used to support some particular values for an attribute, e.g. multiple values for the attribute `begin` or `end`: it is sufficient

to translate the tag to an equivalent one which chooses the minimum value for the attribute. This also means that our approach is open to future modifications of SMIL.

Other works in literature study a way to find out temporal conflicts into SMIL documents. In [17, 16, 19], Sampaio et al describe RT-LOTOS, a formal description of SMIL tags which enables the generation of a valid scheduling for its rendering, considering QoS problem. The authors do not aim at defining a semantics for SMIL language, but compare different players' behaviors which are still implementation-dependent.

Yang [20] and Yu [21] proposes the use of Petri Nets to describe the temporal evolution of a SMIL document. Yang translates the SMIL synchronization tags into transitions and places of the *Real Time Synchronization Model (RTSM)* and tries to detect possible temporal conflict, but this work is limited to the features of SMIL 1.0. Yu defines a formalism based on Petri Nets named *SAM (Software Architecture Model)* which aims to check if QoS properties, expressed through logical formulas, are satisfied, and not to the verification of the semantic correctness of the SMIL document.

The only real attempt to define a formal semantics for SMIL is presented in [10] by Jourdan. This approach is based on the use of timed automata and has been used during the design of SMIL 2.0 to improve specification, since the author was a co-editor of the document which describes timing and synchronization features of this language. The work presented in this paper mainly focuses on SMIL 1.0 and take into account only two new features of SMIL 2.0.

Other works address the problem of temporal consistency of multimedia documents not described with SMIL language. Among others, Elias [6] presents an algorithm, based on the graph theory, which is able to dynamically maintain a consistent and complete set of constraints during the authoring phase. Other works address the same problem with constraints solver techniques. Differently from our approach, all the works addressed here require to translate the SMIL document into another formalism, e.g., a set of temporal constraints or a Petri Net, in order to check its temporal consistency. This operation is not always cost-effective, especially when the complexity of the input file increases and a non compositional approach is used.

Finally, since most available authoring systems adopt *"... SMIL language for building the final representation scheme"* [4], we argue that a formal semantics for this language is needed.

6. ACKNOWLEDGMENTS

The authors would like to thank Maria Luisa Sapino and the anonymous referees for their helpful suggestions.

7. REFERENCES

[1] J. F. Allen. Maintaining knowledge about temporal intervals. *Comm. ACM*, 26(11):832–843, Nov. 1983.

[2] D. C. A. Bulterman, L. Hardman, J. Jansen, K. S. Mullender, and L. Rutledge. GRiNS: A GRaphical INterface for Creating and Playing SMIL documents. In *WWW Conference*, volume 30(1-7), pages 519–529, Brisbane, AU, Apr. 1998.

[3] Dick Bulterman et al. Synchronized Multimedia Integration Language (SMIL) 3.0 Working Draft, December 2006.

[4] E.Bertino, E. Ferrari, A. Perego, and D. Santi. A Constraint-Based Approach for the Authoring of Multi-Topic Multimedia Presentations. In *ICME*, pages 578–581, July 2005.

[5] H. Eidenberger. SMIL and SVG in teaching. In *Internet Imaging V*, volume 5304, pages 69–80, Dec. 2003.

[6] S. Elias, K. S. Easwarakumar, and R. Chbeir. Dynamic consistency checking for temporal and spatial relations in multimedia presentations. In *SAC*, pages 1380–1384, April 2006.

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

[8] C. A. R. Hoare. An axiomatic basis for computer programming. *Comm. of the ACM*, 12(10):576–585, 1969.

[9] Jeff Ayars et al. Synchronized Multimedia Integration Language (SMIL) 2.0 Specification, January 2005.

[10] M. Jourdan. A formal semantics of SMIL: a web standard to describe multimedia documents. *Computer Standards & Interfaces*, 23(5):439–455, 2001.

[11] 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, Sept. 1998.

[12] Oratrix. GRiNS. http://www.oratrix.com.

[13] Patrick Schmitz and Jeff Ayars and Bridie Saccocio and Muriel Jourdan. The SMIL 2.0 Timing and Synchronization Module, March 2001.

[14] F. Paulo, P. Masiero, and M. F. de Oliveira. Hypercharts: Extended Statecharts to Support Hypermedia Specification. *IEEE Trans. on Software Engineering*, 25(1):33–49, Jan. 1999.

[15] RealNetworks. RealPlayer 10.5. http://www.real.com/.

[16] P. Sampaio and J.-P. Courtiat. An Approach for the Automatic Generation of RT-LOTOS Specifications from SMIL 2.0 Documents. *Journal of the Brazilian Computer Society*, 9(3):39–51, Apr. 2004.

[17] P. Sampaio, C. Santos, and J.-P. Courtiat. About the Semantic Verification of SMIL Documents. In *ICME*, pages 1675–1678, New York, USA, Aug. 2000.

[18] L. F. G. Soares, R. F. Rodrigues, and D. C. M. Saade. Modeling, authoring and formatting hypermedia documents in the HyperProp system. *Multimedia Systems*, 8(2):118–134, 2000.

[19] P. Valente and P. Sampaio. TLSA Player: A tool for presenting consistent SMIL 2.0 documents. In *Proc. of ICEIS2007*, Madeira, Portugal, June 2007.

[20] C. Yang. Detection of the time conflicts for smil-based multimedia presentations. In *Workshop on Computer Networks, Internet, and Multimedia*, pages 57–63, 2000.

[21] H. Yu, X. He, S. Gao, and Y. Deng. Modeling and Analyzing SMIL Documents in SAM. In *MSE*, pages 132–135, Newport Beach, California, Dec. 2002.