Spatio-temporal Validation of Multimedia Documents

Joel A. Ferreira dos Santos, Christiano Braga, Débora Christina Muchaluat-Saade, Cécile Roisin, Nabil Layaïda

Spatio-temporal Validation of Multimedia Documents

Joel A. F. dos Santos
Univ. Grenoble Alpes, Inria, LIG - Grenoble, France
MídiaCom Lab, Univ. Federal Fluminense - Niterói, Brazil
joel.ferreira-dos-santos@inria.fr

Christiano Braga
Computer Science Dep.
Univ. Federal Fluminense
Niterói, Brazil
cbraga@ic.uff.br

Débora C. Muchaluat-Saade
MídiaCom Lab, Computer Science Dep., Univ. Federal Fluminense - Niterói, Brazil
debora@midiacom.uff.br

Cécile Roisin
Univ. Grenoble Alpes
Inria, LIG
F-38000 Grenoble, France
cecile.roisin@inria.fr

Nabil Layaïda
Inria, LIG
Univ. Grenoble Alpes
F-38000 Grenoble, France
nabil.layaida@inria.fr

ABSTRACT

A multimedia document authoring system should provide analysis and validation tools that help authors find and correct mistakes before document deployment. Although very useful, multimedia validation tools are not often provided. Spatial validation of multimedia documents may be performed over the initial position of media items before presentation starts. However, such an approach does not lead to ideal results when media item placement changes over time. Some document authoring languages allow the definition of spatio-temporal relationships among media items and they can be moved or resized during runtime. Current validation approaches do not verify dynamic spatio-temporal relationships. This paper presents a novel approach for spatio-temporal validation of multimedia documents. We model the document state, extending the Simple Hypermedia Model (SHM), comprising media item positioning during the whole document presentation. Mapping between document states represent time lapse or user interaction. We also define a set of atomic formulas upon which the author's expectations related to the spatio-temporal layout can be described and analyzed.

Categories and Subject Descriptors

D.2.4 [**Software Engineering**]: Software/Program Verification—*Validation*; F.3.2 [**LOGICS AND MEANINGS OF PROGRAMS**]: Semantics of Programming Languages—*Program analysis*

General Terms

Verification

DocEng'15, September 8–11, 2015, Lausanne, Switzerland.

ACM 978-1-4503-3307-8/15/09 ...$15.00.
DOI: http://dx.doi.org/10.1145/2682571.2797060.

Keywords

Multimedia document validation; Spatial validation; Temporal validation

1. INTRODUCTION

One approach for authoring multimedia documents is using a declarative authoring language. Following such an approach, an author describes the spatial and temporal layout of a multimedia presentation in terms of media items and relations (in time and/or space) among them. We consider user interaction as a special case of temporal relation.

Although the use of a declarative authoring language is intended to make the authoring effort easier, it is still possible that the resulting spatio-temporal[1] layout does not fit the author's expectations due to the incorrect use of constructions available in the authoring language in use. We call an *undesired behavior* every mismatch between "what the author wants" and "what the author gets". It is important, thus, that authors are able to identify undesired behaviors in the created multimedia document prior to its deployment.

An attempt to identify undesired behaviors is usually simulation of the document presentation. This process, however, is usually not *effective* since several executions would be necessary for the verification of undesired behaviors, and may be *incomplete*, from a correctness perspective, since the computations representing the document presentation may be infinite. Moreover, multimedia documents can be automatically generated inside a production pipeline. In this case, simulating the execution of the document before deploying it would be costly, if possible.

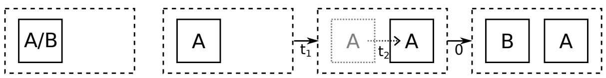
In previous work [8, 9], we presented a state space [1] approach aiming at the temporal validation of multimedia documents. Our approach relies on a formal representation of the behavior of multimedia documents with *SHM* (Simple Hypermedia Model). *SHM* captures the general behavior of multimedia documents as a rewrite theory $\mathcal{R}_{SHM}$. The description of a given document d in *SHM* as a rewrite theory $\mathcal{R}_{SHM}(d)$ induces a transition system $\mathcal{S}_{SHM}(d) = (S, \rightarrow)$,

[1]From now one we use the term *spatio-temporal layout* to represent a combination of both the spatial and temporal layout, the same holds for the term *spatio-temporal relation*.

where each state in S represents the state of d as a whole in a given moment of its execution, and each transition in $\rightarrow$ models a user interaction or a time lapse.

In our previous work, and also in related work, the validation of the spatial layout was performed separated from the temporal layout of a multimedia document. Spatial validation is usually offered by indicating possible overlapping of media items. In our previous work, this was done in a two-step approach, where we first verified if media items would overlap based on their initial position and then verified if they were presented together.

As an example of spatio-temporal validation, consider the document in Figure 1, which has media items A and B[2]. In the beginning of the document execution, media A is presented in the *left position*[3]. t_1 time units after the beginning of the presentation, A begins to slide to the *right position*. t_2 time units after t_1, A arrives at the *right position* and then B is presented in the *left position*.



(a) initial positioning (b) positioning in relation to time

Figure 1: Spatial layout of document example

If we consider just the initial positioning for A and B, and not the relative positions they occupy later on, we may infer that they overlap in space, since both have the same initial positioning (Figure 1a) and both are presented together. However, when the position of media items changes in time, as it can be seen in Figure 1b, reasoning about space separate from time gives incorrect results. Therefore we need to consider the position of media objects during the whole document execution.

We propose a novel spatial validation by extending *SHM* such that the document state now comprises media positioning. Positioning information may change over time (if applicable) according to the document specification. We also create a set of atomic formulas upon which the author's expectations, related to the spatio-temporal layout, can be described.

The remainder of this paper is structured as follows. Section 2 discusses our understanding of spatio-temporal validation. Section 3 presents related work considering the validation of multimedia documents. Section 4 discusses our approach for modeling of the spatio-temporal layout of a multimedia document in order to perform validation. Section 5 discusses limitations and future directions to our work. Section 6 concludes this paper and presents future work.

2. SPATIO-TEMPORAL VALIDATION

The spatio-temporal layout of a multimedia presentation is commonly defined as two disjoint sets of definitions. In the temporal axis, media items are placed in time either with absolute values or in relation to other media items or event occurrences, such as user interaction. In the spatial axis, as presented in [13], media items are placed in relation to the screen, another media or into predefined channels. Usually, media positioning is defined in relation to the screen, in absolute values (pixels) or relative (percentage) values.

A media position may change over time. Some authoring languages allow the author to change media position in response to the occurrence of events in the presentation. Such a change may comprise: moving a media around by changing, for example, its left/top attributes; or scaling a media by changing, for example, its width/height attributes. It is worth mentioning that such changes may occur incrementally over a time interval, as it happens in the example of Figure 1.

In this section we present a series of small examples of a multimedia document d that presents two media items A and B. The author of such examples defines the position of A and B in absolute values. Each figure represents an example, where the dashed rectangle represents the screen (at some moment) and solid rectangles represent the region where a media item is presented. Arrows between two screens represent a time lapse and arrows between regions inside the same screen represent movement, which can be done incrementally over a time period whose duration is presented over the arrow.

Case 1. This example describes a static spatial layout, i.e. A and B do not change their position over time. This case is a pure temporal example, where A is presented (just) before B in time. Figure 2 presents the spatio-temporal layout the author perceives from document d.

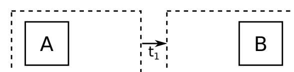


Figure 2: Case 1 spatio-temporal layout

considering a temporal layout example, most of the author's expectations can be described with temporal properties. For example, the author may want to ensure that, for this document, A *before* B, or that *not*(A *together* B). However, it is also possible for the author to express spatial properties. For example, the author may wish to ensure that A *sideof* B, or A *samesize* B. Since the spatial layout is static, spatial validation can be done over the initial position of media items.

Case 2. This example involves the change over time of the spatial layout of a multimedia document. In this example, A has a fixed position and B moves across the screen, changing its position incrementally over t_1 time units. Figure 3 presents the spatio-temporal layout the author perceives from document d.

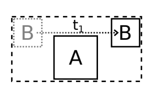


Figure 3: Case 2 spatio-temporal layout

Validating the spatio-temporal layout of such an example is not so simple as statically validating the spatial layout and, in parallel, validating the temporal layout. For example, suppose the author wish to ensure that at some point, while moving across the screen, B will overlap A. This requires verifying if, in at least one moment during the document execution, A and B will overlap. Such kind of property

[2]For simplicity, from now on we may refer to a media item just by media or by its name (A or B in this example).
[3]We describe media positioning by *left position* and *right position*, with respect to the device screen, since a specific value is not important for this example.

has to be encoded by composing temporal and spatial properties, such as *somepoint*(A *overlap* B).

Case 3. This example, as in case 2, involves change of the spatial layout of a multimedia document over time. However, more than one change in the spatial layout occur in sequence. Figure 4 presents the spatio-temporal layout the author perceives from document d.

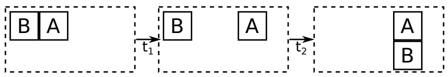


Figure 4: Case 3 spatio-temporal layout

In this example, we have two sequential changes in the position of media items A and B. First, A after t_1 time units changes its position getting detached from B. After A and B are detached for t_2 time units, B changes its position to be below A and, once again, attached to A. The author may wish to enforce, in this case, for example, that *somepoint*(A *detached* B), or yet A *together* B. Interesting properties to be verified, however, involve the sequencing of states, such as: *given that we reach a state where A is at the side of B, can we reach another state where A is above B?* To encode such kind of property, one could use a temporal connector such as (A *side* B) *then* (A *above* B).

3. RELATED WORK

The literature is rich on the discussion of spatio-temporal validation of multimedia documents. Some of the approaches discussed in this section do not primarily address validation but could be considered in such a task. We classify them according to coverage in either spatial or temporal dimensions, as follows: (i) purely temporal, (ii) both temporal and spatial but with the spatial dimension static, and (iii) both spatial and temporal dimensions (*spatio-temporal*). We also classify them according to the reasoning principle applied for document validation. The first group of papers (first line of Table 1) relates to validation of a multimedia document by investigating the document state over its presentation. This may be done by reachability analysis or the application of axioms over the document state. The second group of papers (second line of Table 1) relates to validation of a multimedia document by checking the consistency of a set of constraints.

	Purely Temporal	Temporal + Spatial (static)	Spatio-Temporal
Doc. State	[20, 7, 12, 5]	[15]	[16]
Constraint	[4, 11, 17]		[3]*

Table 1: Related work classification

Santos et al. in [20] discuss an approach for the temporal validation of multimedia documents by translating them into the real-time process algebra framework RT-LOTOS. It combines processes that specify the document presentation with other processes that model the available platform. A minimum reachability graph is built from the RT-LOTOS formal specification such that each node in the graph represents a reachable state and each edge the occurrence of an event or temporal progression. The validation is achieved by verifying, for example, if the state corresponding to the end of the document presentation can be reached from the document's initial state. Similarly, verifying if a media item will be executed is performed by determining if a state where it is being executed is reachable from the document's initial state. The tool presented in [20] can validate NCM (Nested Context Model) [22] and SMIL (Synchronized Multimedia Integration Language) [23] documents.

Oliveira et al. in [7] introduce HMBS (Hypermedia Model Based on Statecharts). An HMBS multimedia application is described by a statechart, where states represent pages (i.e. the information presented to the user) and transactions and events represent a set of possible link activations. The validation of an HMBS application is performed over a reachability tree, which is built from the application statechart. From the reachability tree, it is possible to determine if a given page is reachable or not and also if a group of pages is presented simultaneously or not. The reachability graph also allows the detection of configurations from which no other page may be reached or that present cyclical paths.

Felix in [12] presents an approach for the validation of temporal properties of NCL documents through the application of model checking techniques. He presents a notation for the description of NCL's temporal characteristics. Such a description is transformed into a timed automata net that indicates the document temporal behavior. The transformation creates a state machine for each media item and a synchronizer machine for each link declared in the document. A synchronizer machine is used for tying together the occurrence of events in media state machines. The validation of an NCL document is performed over the timed automata net representing the document using temporal logic formulas created by the author.

Bossi and Gaggi in [5] define a formal semantics for SMIL through a set of inference rules inspired by Hoare logic. The rules describe the document state before and after the execution of a given SMIL construction. Thus, in the authoring phase, the structure of a SMIL document may be enriched with assertions expressing temporal properties. Another application resulting from the defined formal semantics is the concept of equivalence, which guarantees that two sets of SMIL constructions may be replaced, without changing the presentation behavior. The validation of a document is performed by the application of axioms, also defined in the proposed semantics, that verify if a given construction or set of constructions correctly change the document state. Otherwise, it presents to the author the problem found so it can be corrected.

As can be seen in Table 1, the works presented in [20, 7, 12, 5] present a purely temporal approach, where the validation of a multimedia document is performed by investigating the document state over its presentation. In [20, 7, 12], it is done by reachability analysis and in [5] by analyzing if the document state changes according to some axioms. Validation of the spatial layout of a document is not discussed in those papers.

Júnior et al. in [15] use a model-driven approach for the presentation behavior validation of NCL documents. The validation is achieved by transforming an NCL document into a Petri Net. This transformation is done in two steps. In the first step, the document is represented in a language called FIACRE as a set of components and processes (representing the behavior of a component). The second step transforms the FIACRE representation into a Petri Net. The validation uses a model-checking tool and temporal logic formulas representing the properties to be validated. Spatial validation is briefly discussed in [15] and is performed over the document initial positioning. As presented in Table 1, their approach covers both temporal and spatial dimensions, however the spatial dimension is static.

King et al. in [16] define extensions for the SMIL language allowing authors to describe how the spatio-temporal layout should change in reaction to events. Changes in position and size are described by a set of expressions, which may consider the state of the document. The paper presents an approach for calculating at runtime the value of such expressions, and therefore the change in the spatial layout to be performed. Although such an approach does not refer to document validation, it is an interesting example on how to parameterize by time or an event occurrence the rules that specify the spatial layout of a presentation.

Bertino et al. in [4] propose an authoring model based on constraints. A multimedia application in that model consists of several topics, where each topic is composed by semantically-related media items. The system automatically group media items into topics according to the constraints defined by the author. The application generation process is responsible for three main tasks: consistency checking, presentation structure generation and topics generation. The system enlarge the set of constraints with others that, even not defined explicitly, are consequences of the constraints defined by the author. Consistency checking is then performed over the constraint set. If an inconsistency arises, the system applies relaxation techniques, to reduce the constraint set to a consistent one. When such a reduction is not possible, the author review is required. The presentation structure generation process creates a direct graph that represents the application structure. Each vertex of such graph represents a topic and each edge a connection between topics. After this step, the system relates media items to topics and builds, for each topic, the spatial layout and the temporal sequence of media items belonging to it.

Elias et al. in [11] also propose an authoring model based on constraints. It defines two operators $TEMPORAL$ and $SPATIAL$, to model temporal and spatial relations, respectively. Each operator allows the author to define a priority value. In order to maintain the consistency of the constraint set, whenever necessary, constraints are removed according to this priority value. In case two inconsistent constraints present the same priority, relaxation techniques are applied to determine the constraint to be removed. Besides the verification of inconsistencies among constraints, this approach also enables the author to verify if the constraint set is incomplete, that is, if there is one or more media items that are not reached during presentation. The consistency checking is done by finding the minimum spanning tree T in the constraint graph. Constraints that create cycles are removed to maintain the acyclic nature of T. Completeness checking is done by searching all media items reachable from the first media item. If this search returns the vertex set of T, then all items are reached directly or indirectly from the initial one. Otherwise, the author has to define constraints to make the constraint set complete. With the use of the $SPATIAL$ operator, it is possible to determine if A overlaps B and vice versa. The spatial consistency is checked the same way as the temporal one.

Laborie et al. in [17] presents an approach for the automatic adaptation of the layout of a multimedia presentation according to the display used for its presentation. The approach creates an abstract description of the author's document as a set of objects and constraints representing temporal and spatial relations among objects. It also takes into account a profile comprising device constraints together with user preferences. Given the set of potential document executions $\mathcal{M}_s$ given by the abstract description and the set of potential executions $\mathcal{M}_p$ given by the profile, the adaptation process calculates $\mathcal{M}_s \cap \mathcal{M}_p$ to determine if some adaptation is required or not. In case the document has to be adapted, the goal is to change document relations such that it now complies with the profile and the (behavioral) distance from the previous declaration is minimum.

As can be seen in Table 1, the works presented in [4, 11, 17] cover both temporal and spatial dimensions, where the validation of a multimedia document is performed by consistency checking over a set of constraints. In each work, however, the spatial dimension is static, since spatial constraints do not change over time. Moreover, reasoning about time and space in [4, 11, 17] is performed as two separated problems.

Belouaer and Maris in [3] present a SAT Modulo Theory (SMT) [2] approach for solving spatio-temporal planning problems. A set of constraints modeling the spatial disposition of items and their hierarchy is used to describe both the initial state of a given problem and its goal. Other constraints model actions that change the spatial position of items. Such actions may define an inherent duration and also at each moment (in time) they should be applied. By solving the problem, taking into account the constraints representing actions, it is possible to verify if the goal can be achieved or not. Although this work does not refer to multimedia documents, it is an interesting example on how spatial constraints can be parameterized by time in order to cover both spatial and temporal dimensions.

4. MODELING SPATIO-TEMPORAL LAYOUT

In our work, we use a state space approach for the validation of multimedia documents. We rely on the representation of the behavior of multimedia documents as a rewrite theory, which we call *SHM* [9]. In Section 4.1 we briefly describe the *SHM* model. The proposed extensions to allow spatio-temporal validation are presented in Section 4.2. Next, Section 4.3 discusses how the author expectation can be described in *SHM*. Section 4.4 presents the use of *SHM* with a real document. Finally, Section 4.5 presents an evaluation of our approach.

4.1 SHM

This section gives a brief description of *SHM*. Our intention is to equip the reader with sufficient tools to understand our validation approach and its extensions in Section 4.2. A more comprehensive description of *SHM* is available in [9].

Essentially, multimedia documents describe the spatio-temporal layout of a multimedia presentation in terms of media items and relations (in time and/or space) among them. The presentation as a whole can be parameterized by variables inherent to media items or global to the document.

SHM captures the general behavior of multimedia documents as a rewrite theory $\mathcal{R}_{SHM} = (\Sigma, E, R)$, where (Σ, E)

defines the constructs [4] to be used to represent the document state (in a given moment during its execution), and R defines rules that induce transitions among states.

The document state is represented by the composition of the states of all media fragments declared inside a document and the value of variables. A media fragment represents a subpart of a media item, possibly the whole item. State information for a given fragment or variable is represented by means of state machine configurations (*SMConf*). An *SMConf* has the general form $\langle id, ty, st, oc, cl, v\rangle$, where *id* is a fragment or variable identifier, *ty* is the state machine type, *st* its state, *oc* its occurrences counter, *cl* its countdown clock and v its value (in case *SMConf* represents a variable). The type of an *SMConf* can be either *[pre]sentation* when it represents the presentation state of a fragment, *[sel]ection* when it represents user selection over a fragment and *[att]ribution* when it represents changes in the value of variables. The state of an *SMConf* has the following possible values: *sleeping*, *occurring* and *paused*.

Every state machine configuration starts in the *sleeping* state. As the presentation goes on, the *st* component will eventually change to the *occurring* state. If we suppose an *SMConf* associated with a media item representing a video object, as the first frame of the video begins its presentation, the *st* component of the *SMConf* representing its presentation transits to the *occurring* state. The *st* component of an *SMConf* remains in the *occurring* state for a given period of time, represented by the *SMConf*'s countdown clock. As soon as the last frame of the video finishes its presentation (i.e. its countdown clock reaches zero), the *st* component of *SMConf* goes back to the *sleeping* state. This is what we call the *natural end* of a media presentation. It is important to highlight that not all media items may have a natural end. One example is a media item representing an image, which does not have an inherent duration. In this case, the *st* component of the *SMConf* representing that media item will remain in the *occurring* state indefinitely.

Relations in *SHM* may be defined among fragments, variables or a combination of both. Every relation is represented by an equation with the general form $[l]\ C_o \rightarrow C_f\ \textit{if}\ P$, where l is a label to the relation, C_o and C_f represent (part of) the document configuration and P is a predicate over the state of fragments or the value of variables. Given that the document state reaches a configuration that contains C_o and predicate P is evaluated as *true*, we rewrite C_o by C_f, thus changing the document configuration according to a given relation. Relations are triggered by state changes in an *SMConf* and, as an effect, produce changes in the state of other *SMConf*s (possibly the same). Labels in Figure 5 correspond to the state changes considered by *SHM*.

Relations may be applied from time to time, whenever the presentation reaches a configuration where a relation can be applied. Those configurations are reached either by a time lapse (when the natural end of a fragment occurs) or by user interaction with a fragment. Therefore, we model time lapse and user interaction by rules *step* and *interact* in R. Rule *step* fast-forwards the presentation by decrementing countdown clocks of every *SMConf* in the *occurring* state. Given that at least one *SMConf* clock will reach zero, the resulting state change (possibly) triggers the application of relations further changing the document state as a whole. Given that the presentation state of a fragment is in the *occurring* state, rule *interact* changes its selection state at any point during its presentation. The non-determinism induced by the interleaving of both rules produces different traces of the document presentation.

[4]As presented along this section, such constructs declare state machine configurations, their composition into the whole document and operations for changing their state.

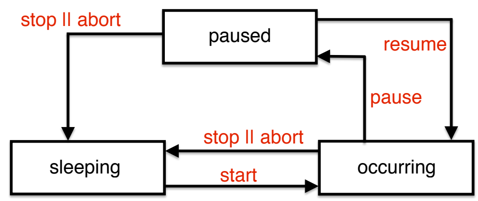


Figure 5: State changes

For a given document d, we extend theory $\mathcal{R}_{SHM}$ into a rewrite theory $\mathcal{R}_{SHM}(d)$ such that $\mathcal{R}_{SHM}(d)$ declares all fragments inside d and variables considered in relations. For each relation in d one or more rewrite rules are created to represent such a relation. Moreover, additional rewrite rules are created to model relations among fragments of a media item and the media item as a whole, and relations among media items and their parent composition, if there is one. Transformation τ_{SHM} automates such process creating rewrite theory $\mathcal{R}_{SHM}(d)$ from a multimedia document d.

In [8, 9] we have used *SHM* for the temporal validation of multimedia documents through model checking. Theory $\mathcal{R}_{SHM}(d)$ induces a transition system $\mathcal{S}_{SHM}(d) = (S, \rightarrow)$, where each state in S represents the state of d as a whole in a given moment of its execution, and each transition in $\rightarrow$ models the user interaction or a time lapse. Moreover, a set of predefined expected behaviors were formalized as LTL [18] (Linear Temporal Logic) formulas, such as: *reachability* (will a media be presented?), *media termination* (given that a media is presented, does it end?) and *document termination* (does the document as a whole end?).

4.2 SHM extension for Spatio-Temporal Validation

In our previous work, we have attempted to offer spatial validation by indicating possible overlapping of media items. This was done in a two-step approach, where we first verified if media items might overlap based on their initial position and then verified if they were presented together.

In this work, to enhance the spatio-temporal validation, we extend *SHM* in two ways: (i) the document state now stores the position of a media item (or a fragment of it) and (ii) given that the position of a media changes over time this behavior has to be captured by *SHM*.

The former is achieved by representing every positioning attribute - *left*, *top*, *width* and *height* - as a variable, so that the document state is able to store its value during the execution. In order to be able to relate a given variable to a media item attribute, we extend *SHM* with the following functions:

$$left, top, widht, height : MedId \rightarrow VarId$$

where *MedId* and *VarId* are sets of identifiers for media items and variables in *SHM*, respectively. With such approach, we can evaluate the value of the *left* attribute of media A, by evaluating the value of variable $\mathit{left}(A)$.

During the document presentation, it is possible that media items change their position. A common way to achieve it is to declare relations that change the value of the positioning attributes in response to an event occurrence. This change can be either *discrete*, or *incremental* over a time interval. In the latter, the relation provides, together with the new values, the duration for the change and the increment by which values have to be changed.

In order to model such behavior, we extend transformation τ_{SHM} so that for every relation in d that changes the value of positioning attributes of media items, one or more equations perform the same change in $\mathcal{R}_{SHM}(d)$. For example, suppose the following causal relation declared in d[5].

$$[\mathit{r1}]\ A.\mathit{begin} \rightarrow B.\mathit{left} := 400\ \mathit{if}\ \top$$

It states that whenever the document reaches a configuration where media A begins its presentation, we change the value of attribute *left* of media B to 400.

Transformation τ_{SHM} will create the following rewrite rule for representing relation *r1*, where *pre* and *att* is the short for presentation and attribution, respectively.

$$[\mathit{r1}]\ A.\mathit{pre}.\mathit{begin} \rightarrow \left\{ \begin{array}{c} \mathit{left}(B).\mathit{att}.\mathit{start}, \\ \mathit{left}(B).\mathit{value} = 400 \end{array} \right\}\ \mathit{if}\ \top$$

The relation presented above exemplifies a discrete change of the positioning attributes of media B. It is possible that this change is incremental over a time interval. For example, suppose the following causal relation *r2*.

$$[\mathit{r2}]\ A.\mathit{begin} \rightarrow \left\{ \begin{array}{c} B.\mathit{left} := 400 \\ \mathit{during} : 4s\ \mathit{by} : 10px \end{array} \right\}\ \mathit{if}\ B.\mathit{left} == 0$$

It extends relation *r1*, such that, given A beginning its presentation and B's left position is equal to 0, the positioning attribute *left* changes its value by an increment of 10 *pixels* for 4 *seconds*. The *left* value changes 400 *pixels*, with an increment of 10 *pixels*, thus 40 incremental changes are produced over 4 seconds. Therefore each change occurs at each 0.1 seconds.

Transformation τ_{SHM} will create the following rewrite rules for representing relation *r2*.

$$[\mathit{r2\_init}]\ A.\mathit{pre}.\mathit{begin} \rightarrow \left\{ \begin{array}{c} \mathit{left}(B).\mathit{att}.\mathit{start}, \\ \mathit{left}(B).\mathit{value} += 10, \\ C.\mathit{pre}.\mathit{start} \end{array} \right\}$$
$$\mathit{if}\ \mathit{left}(B).\mathit{value} == 0$$

$$[\mathit{r2\_inc}]\ C.\mathit{pre}.\mathit{end} \rightarrow \left\{ \begin{array}{c} \mathit{left}(B).\mathit{att}.\mathit{start}, \\ \mathit{left}(B).\mathit{value} += 10, \\ C.\mathit{pre}.\mathit{start} \end{array} \right\}$$
$$\mathit{if}\ C.\mathit{pre}.\mathit{occur} < 40$$

where C has a duration of 0.1 seconds and is used to represent the delay between two incremental changes. Operation $+=$ represents an increment operation over the value of variables.

[5]We use the same notation of *SHM* for simplifying the example.

4.3 Describing the author's expectation

Multimedia documents can be created by an author with a range of available authoring tools. Depending on the multimedia language in use, tools may be able to provide the author with different views of the document, besides a preview of its spatio-temporal layout. Moreover, multimedia documents can be automatically generated, making it difficult for an author to know all of its spatio-temporal layout specifications. Our approach is intended to be used after the authoring phase, where the author wants to verify if the spatio-temporal layout of a given existing document is the expected one.

As presented in Section 4.1, in [8, 9] we formalized a set of predefined expected behaviors, such as: *reachability* (will a media be presented?), *media termination* (given that a media is presented, does it end?) and *document termination* (does the document as a whole end?). For each document to be validated those properties were verified.

With the *SHM* extensions presented in Section 4.2 we are able to represent media item position and their change across the document presentation. In order to enable the description of the author's expectation about the spatio-temporal layout he/she might perceive, we define a set of atomic formulas in either temporal or spatial axis. The description of a spatio-temporal layout, therefore, is achieved by a combination of both.

In the temporal axis, formulas represent Allen's relations [1] between time intervals as presented in Figure 6, where rectangles represent time intervals.

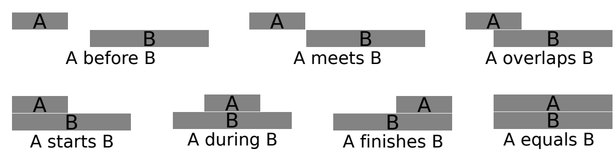


Figure 6: Allen's relations between time intervals

In the spatial axis, formulas represent RCC spatial relations between regions [19] as presented in Figure 7, where rectangles represent regions.

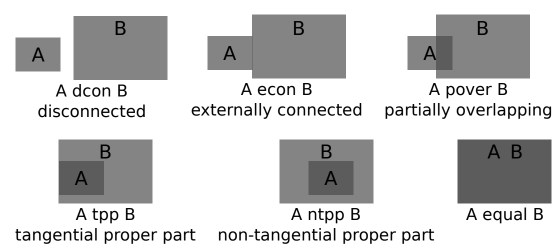


Figure 7: RCC spatial relations between active regions

Although RCC relations enable to relate media items in space, they may not be enough to represent the exact spatial layout expected by an author. Suppose, for example, the three configurations presented in Figure 8. Each example represents a spatial layout relating media items A and B.

It is worth noticing that every example can be described by relation *pover*, since in each example A partially overlaps B. However, they can not be considered the same spatial layout since A and B assume different relative positions in each example.

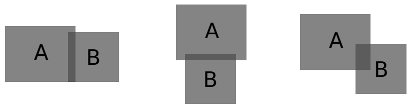


Figure 8: Possible cases of A *pover* B

To be able to describe the relative position between media items, we extend RCC spatial relations (except *equal*), so that relations are parameterized by the angle between media items. The angle between two media items is calculated as presented in Figure 9. It presents two *disconnected* media items A and B with A being at an angle α with respect to B.

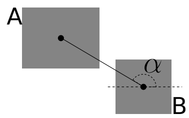


Figure 9: Angle between media items with respect to B

We represent such an example by the formula $A\ dcon(\alpha)\ B$.

Angle α can be described in degrees or, if such precision is not necessary, by a cardinal direction as presented in Figure 10.

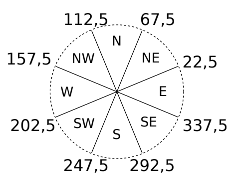


Figure 10: Cardinal direction

Thus, the example of Figure 9 may also be represented by formula $A\ dcon(NW)\ B$.

We formalize the properties we want to validate in LTL [18]. An LTL formula φ is defined as follows, where X, F, G, U, W and R are called temporal operators.

$$\begin{aligned} \varphi \quad ::= \quad & \top \mid \bot \mid p \mid \neg(\varphi) \mid (\varphi \wedge \varphi) \mid (\varphi \vee \varphi) \mid (\varphi \rightarrow \varphi) \mid \\ & (X\varphi) \mid (F\varphi) \mid (G\varphi) \mid (\varphi\ U\ \varphi) \mid (\varphi\ W\ \varphi) \mid (\varphi\ R\ \varphi) \end{aligned}$$

$X\varphi$ (*next*) states that a formula φ must be valid for the following state. $F\varphi$ (*future*) states that a formula φ must be valid for some future state. $G\varphi$ (*global*) states that a formula φ must be valid for all states in a path. $\varphi_1\ U\ \varphi_2$ (*until*) states that a formula φ_1 must be valid until a formula φ_2 becomes valid. $\varphi_1\ W\ \varphi_2$ (weak until) states that a formula φ_1 must be valid until a formula φ_2 becomes valid or φ_1 must be valid for all states in the path. $\varphi_1\ R\ \varphi_2$ (release) states that a formula φ_1 must be valid until a formula φ_2 becomes valid and both φ_1 and φ_2 must be valid at the same time for some state.

It is worth noticing that formulas in the temporal axis describe the evolution of (part of) the document state through several states. For example, formula A *meets* B is described by the following LTL formula.

$$\begin{aligned} i_1\ meets\ i_2 = \quad & F((i_1.pre.occurring \wedge i_2.pre.sleeping) \wedge \\ & X(i_1.pre.sleeping \wedge i_2.pre.occurring)) \end{aligned}$$

On the other hand, formulas in the spatial axis consider the values of positioning attributes inside a given state. They are combined with a temporal operator for representing a spatio-temporal layout. As an usage example of a spatial formula, suppose we want to verify in **case 2** of Section 2 if at some point A and B will overlap in space. Thus we write the following formula

$$F(A\ pover\ B)$$

where we combine spatial relation *pover* with temporal operator F (future).

4.4 Modeling in practice

Both the *SHM* model and transformation τ_{SHM} are implemented using the Maude system [6]. The verification of LTL formulas is performed by the Maude model-checker tool. Transformation τ_{SHM} has been implemented for both NCL [14] and SMIL [23] documents. The extensions of τ_{SHM} presented in this paper have been implemented only for NCL document transformation.

Figure 11, adapted from [21], presents the spatio-temporal layout of an excerpt of document *First João* used as case-study. It presents a main video (media item *video*) about a soccer player. At some point during the video presentation, an advertisement icon is presented (media item *icon*) at the upper right corner (as seen in Figures 11a and 11b). If the user interacts with the icon, the video is downsized and repositioned (upper left corner) and a video of a kid thinking about shoes (media item *kid*) starts playing (as seen in Figure 11c).

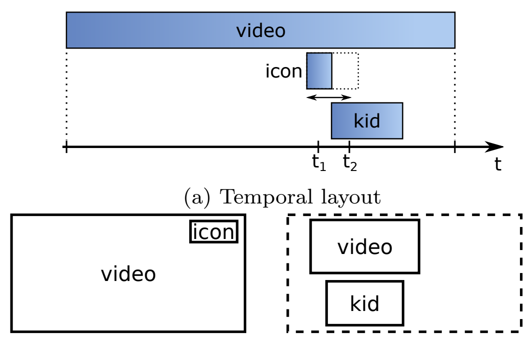


Figure 11: First João spatio-temporal layout

In such an example the author may want to ensure either temporal properties, spatial properties or both. For example, suppose the author wants to ensure that media items *video* and *kid* will never overlap (in space). The author may also want to ensure that media *video* is presented above *kid*.

This expected spatio-temporal layout can be described, by using the formulas presented in Section 4.3 as follows

$$kid\ during\ video \rightarrow G(kid\ dcon(S)\ video) \qquad (1)$$

where, given that *kid* is presented while *video* is presented, then *kid* has to be disconnected from *video* and below it (at its south).

Using *SHM* we can validate Formula 1 by using the Maude model-checker as follows, where [] is used to represent the temporal operator G (global).

```
Maude> red modelCheck(run, ('kid pre during 'video pre) −>
                                                  []('kid dcon[S] 'video)) .
reduce in NCLDOC : modelCheck(run, 'kid pre during 'video pre −>
                                                  []'kid dcon[S]'video) .
rewrites: 2036 in 39ms cpu (48ms real) (51323 rewrites/second)
result Bool: true
```

Formula 1 holds since, as can be seen in Figure 11, *kid* and *video* do not overlap. However, if just their initial position and size were considered, they would overlap and the spatial-temporal analysis would not be correct.

4.5 Evaluation

We performed two tests to evaluate our proposal, in order to indicate a reasonable performance for our approach. Test results are presented in Figure 13. Tests were performed in a Centos 6.7 virtual machine running on four cores of an Intel E5-2650v2, 2.6 GHz with 16 GB of memory.

In the first test we have a document d with two media items A and B. Both A and B start their presentation as d's presentation begins. Media A changes its position and size until it has the same position and size of B and then it remains that way. Document d's presentation ends when both A and B finish their presentation. Figure 12 presents the spatio-temporal layout for this test.

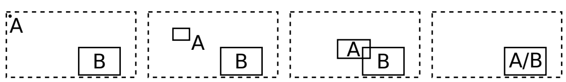


Figure 12: Spatio-temporal layout for the first test

The first test consists in incrementing the number n of steps for changing A's position and size until it reaches the same position and size of B. For each value of n we ran the following Maude command and gathered the statistics provided by Maude.

```
red modelCheck(run, <>('A equal 'B)) .
```

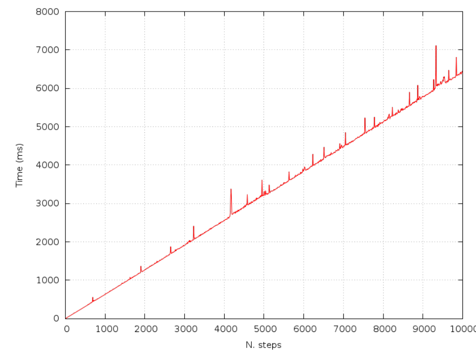
As the number of steps grows, the number of states in $\mathcal{S}_{SHM}(d)$ grows linearly with it. The impact in increasing the number of steps in the time Maude takes to perform the above command is presented in Figure 13a. As it can be seen, time also increases linearly with the number of steps.

In the second test, we fixed the number of steps to 10 and increased the number n of media items inside the document changing their position. For each value of n we ran the following two Maude commands and gathered the statistics provided by Maude.

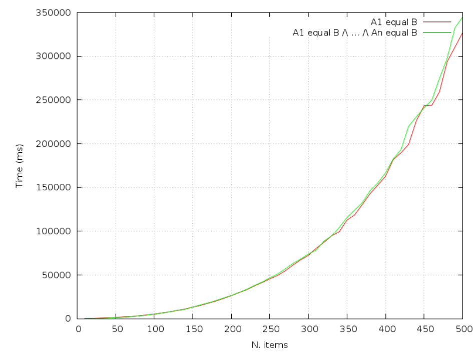
```
red modelCheck(run, <>('A1 equal 'B)) .
red modelCheck(run, <>('A1 equal 'B /\ ... /\ 'An equal 'B)) .
```

The impact in increasing the number of media items in the time Maude takes to perform the two commands above is presented in Figure 13b. As it can be seen, time grows exponentially with the number of items and the size of the formula to be tested has almost no impact in time. Testing the execution of each test document alone indicated that approximately all the time spent by validating the above formula was spent by Maude in building the transition system where the temporal formulas were verified.

Comparing our approach presented in this paper to our previous works indicates that time increases mostly because of the growth in the number of state machines required for



(a) Number of steps X time (ms)



(b) Number of items X time (ms)

Figure 13: Test results

representing the document's spatial layout. In our tests, for each state machine representing a media item, five more (four for the positioning attributes and one for the increment delay) were created. Comparing the time spent to run each document with a similar one, regarding the number of state machines, but without spatial information, indicates a small increase in time.

From the graph in Figure 13a we see a linear growth of time related to the number of steps for changing the position and size of a media item. For a document with 10000 steps the validation is performed in about 6.5 seconds. It is worth mentioning that, in common multimedia documents, movements of media items take a few seconds, thus, even with a precision of milliseconds we are still able to give an answer to the author in a reasonable time. In general, such a huge number of steps is not necessary and can be abstracted to decrease the overall duration of the validation.

Increasing the number of steps for changing the positioning attributes of media items will linearly increase the number of states, as seen above. A real problem of state explosion arrives, on the other hand, when user interaction is possible. Depending on the document, the time interval when a user can interact with a media item may be very long. Moreover, it is possible for media items to be selected more than once. Since in *SHM* when the user interacts with a media item, a new branch is created when that selection occurred, it is easy to see that as selection grows, more states there will be. To handle such a problem, we enable the author to define a maximum number of user interactions for each media item during transformation τ_{SHM}. Other possible solution for avoiding such increase in the number of states is to create abstractions in the state representation.

5. DISCUSSION

In Section 4.5 we presented test results, which indicate a reasonable performance of our validation approach. Together with the results we discussed some limitations of our work. In this section we present a possible direction for our approach regarding such limitations.

Our validation is performed by model-checking over a transition system representing the document state evolution over time. In our approach, the document state represents current values for information about media items, e.g its state, countdown clock, top position or width. Taking into consideration the spatial layout, the document state describes "the whole" layout (i.e. all positioning attributes) in a given instant.

Another interesting approach would be to validate the spatio-temporal layout of a multimedia document even when "a partial" layout description is provided, for example, by a set of spatial constraints. Thus, the author does not need to provide specific positioning for each media item, but instead, spatial relations among them.

A future direction for our work is to provide a constraint approach for the spatio-temporal validation of multimedia documents. In such an approach, both the temporal and spatial layouts are described through a set of constraints. Spatial constraints may be parameterized by time, such that they are enforced just in a given moment (or interval) during the document execution. By the combination of both approaches, we will be able to perform the spatio-temporal validation presented here but even when a partial description of the presentation spatial layout is provided.

We have been working in this new approach by representing constraints as formulas in the SMT solver Yices [10]. Temporal constraints relate begin and end times of intervals. Spatial constraints relate the borders of rectangular regions. Whenever it is possible to build the layout from the set of constraints, the solver presents a valuation for it.

In the temporal dimension, media items are represented as intervals. Each interval is represented as a tuple $\langle i, e, d, b\rangle$, where i and e represent its initial and end times, respectively, d its duration and b is a boolean that indicates if a given interval is considered or not to be part of the temporal layout.

Using that approach we describe the example in Figure 11. The intervals initial time, end time and duration are gathered from the document declaration, whenever possible. We also have an interval called *selec*, to represent the user interaction over media *icon* such that *selec*, if it occurs, has to occur inside the interval for *icon*.

We model the causal relation specifying that, once *icon* is selected, end media *icon* presentation and start media *kid* presentation as follows, where I^i and I^e represent the initial and end times of interval I, I^d its duration and I^b if it is considered or not as part of the temporal layout.

$$(selec^b \wedge kid^b \wedge (icon^e = selec^e) \wedge (kid^i = selec^e)) \quad \vee \\ (\neg selec^b \wedge \neg kid^b \wedge (icon^e = icon^i + icon^d))$$

Therefore, the interval for media *kid* occurs only if *selec* occurs and not otherwise. Whenever *selec* occurs it causes the end of *icon*.

In such an approach we can verify if it is possible for *kid* to end after *video* (as exemplified above) by adding the following constraint,

$$kid^b \wedge (kid^e > video^e)$$

that, as expected, yields an *unsatisfiable* temporal layout. Yices takes around 5ms for presenting this answer.

In the spatial dimension, media items are represented as rectangles by their projection in either x and y dimensions. Each projection is represented as a tuple $\langle i, c, e\rangle$, where i and e represents its initial and end border and c its center in that dimension. We define a set of constraints to organize media items in space. Constraints are able to *align* two items, *distribute* items inside a region, compare the size of two items and also arrange items in a *flow* inside a region.

To present a reasonable performance for that approach we used Yices to verify if it was possible to organize 49 items inside a 450×450 canvas. Sizes of items represent the combinations of width and height ranging from 10 to 70. The result is *satisfiable* and Yices presents a possible valuation for item regions. It is worth noticing that smaller canvas sizes yielded *unsatisfiable* spatial layouts. The evaluation of such an example in Yices takes around 3s.

The direction of this work, aiming at enhancing the multimedia validation we provide is to combine our existing model-checking approach with such SMT approach. For problems involving conditions over events and specific instances of media items (such as looping media items), it seems straightforward to use model-checking. On the other hand, problems involving numerical dependencies, such as placing media items in time and space, it seems straightforward to use SMT.

In our approach, media items are represented in space by their rectangular region. However, it is possible that the visible amount of a media item has a different form. One solution for improving the precision in media items representation (in space) is to represent it by the composition of several small rectangles. The greater the number of rectangles, the greater the representation precision is. On the other hand, more rectangles means more time for validating a document. An approximation, therefore, is necessary. We believe the best solution is to leave such option for the user.

6. CONCLUSION

Although the use of a declarative authoring language is intended to make the authoring effort easier, it is still possible that the resulting spatio-temporal layout does not fit the author's expectations due to the incorrect use of constructions available in the authoring language in use.

This work proposes an approach for the temporal and spatial validation of multimedia documents such that positioning information may change over time (if applicable) according to the document specification.

The extension of *SHM* for enabling spatio-temporal validation, as verified by tests, increases the time necessary to perform the validation. It occurs mainly because of the increase in the number of state machines necessary for storing the positioning attributes of media items. Moreover, although small, increasing the number of steps in an incremental change of position and/or size also contributes to increase time. As discussed, such an increase, however, is not considered a problem for validating common multimedia documents.

By the combination of formulas in the temporal and in the spatial axis the author may describe the desired spatio-temporal scenario. An ongoing work is to provide a tool where the author may create a description of the desired spatio-temporal scenario. Moreover, this tool enables the author to create a set of tests to be validated over a given multimedia document.

This paper also discussed future directions for our multimedia validation approach. Future directions point towards using a constraint-based approach for representing the spatio-temporal layout of a document and also performing the document validation. Initial results indicate the soundness of such an approach.

7. ACKNOWLEDGEMENTS

We thank the support of both the Brazilian agency CNPq and Inria in the context of Science without Borders Program. We also thank the Brazilian agencies FAPERJ and CAPES for their support.

8. REFERENCES

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

[2] C. W. Barrett, R. Sebastiani, S. A. Seshia, and C. Tinelli. Satisfiability modulo theories. *Handbook of satisfiability*, 185:825–885, 2009.

[3] L. Belouaer and F. Maris. SMT Spatio-Temporal Planning. In *ICAPS 2012 Workshop on Constraint Satisfaction Techniques for Planning and Scheduling Problems (COPLAS 2012)*, pages 6–15, 2012.

[4] E. Bertino, E. Ferrari, A. Perego, and D. Santi. A Constraint-Based Approach for the Authoring of Multi-Topic Multimedia Presentations. In *IEEE International Conference on Multimedia and Expo*, pages 578–581, Amsterdam, Netherlands, July 2005. IEEE Computer Society.

[5] A. Bossi and O. Gaggi. Analysis and verification of SMIL documents. *Multimedia Systems*, 17(6):487–506, 2011.

[6] M. Clavel, S. Eker, F. Durán, P. Lincoln, N. Martí-Oliet, and J. Meseguer. *All about Maude - A High-performance Logical Framework: how to Specify, Program, and Verify Systems in Rewriting Logic*, volume 4350. Springer-Verlag New York Inc, 2007.

[7] M. de Oliveira, M. Turine, and P. Masiero. A statechart-based model for hypermedia applications. *ACM Transactions on Information Systems (TOIS)*, 19(1):52, 2001.

[8] J. A. F. dos Santos, C. Braga, and D. C. Muchaluat-Saade. A Model-driven Approach for the Analysis of Multimedia Document. In *SLE (Doctoral Symposium)*, pages 37–44, Dresden, Germany, 2012.

[9] J. A. F. dos Santos, C. Braga, and D. C. Muchaluat-Saade. An Executable Semantics for a Multimedia Authoring Language. In *Formal Methods: Foundations and Applications*, pages 67–82. Springer, Brasília, Brazil, 2013.

[10] B. Dutertre. Yices 2.2. In A. Biere and R. Bloem, editors, *Computer-Aided Verification (CAV'2014)*, volume 8559 of *Lecture Notes in Computer Science*, pages 737–744. Springer, July 2014.

[11] S. Elias, K. Easwarakumar, and R. Chbeir. Dynamic consistency checking for temporal and spatial relations in multimedia presentations. In *Proceedings of the 2006 ACM symposium on Applied computing*, pages 1380–1384, Dijon, France, 2006. ACM.

[12] M. F. Felix. *Formal Analysis of Software Models Oriented by Architectural Abstractions*. PhD thesis, Pontifícia Universidade Católica do Rio de Janeiro, 2004. in Portuguese.

[13] H. L. Hardman. *Modeling and Authoring Hypermedia Documents*. PhD thesis, Universität Amsterdam, 1998.

[14] ITU. Nested Context Language (NCL) and Ginga-NCL for IPTV services. http://www.itu.int/rec/T-REC-H.761-200904-S, 2009. ITU-T Recommendation H.761.

[15] Júnior, D. P. and Farines, J. and Koliver, C. An Approach to Verify Live NCL Applications. In *Proceedings of the 18th Brazilian Symposium on Multimedia and the Web*, pages 223–232, São Paulo, Brazil, 2012. ACM.

[16] P. King, P. Schmitz, and S. Thompson. Behavioral reactivity and real time programming in xml: functional programming meets smil animation. In *Proceedings of the 2004 ACM symposium on Document engineering*, pages 57–66. ACM, 2004.

[17] S. Laborie, J. Euzenat, and N. Layaïda. Semantic adaptation of multimedia documents. *Multimedia tools and applications*, 55(3):379–398, 2011.

[18] A. Pnueli. The temporal logic of programs. In *18th Annual Symposium on Foundations of Computer Science*, pages 46–57, Providence, USA, 1977. IEEE.

[19] D. A. Randell, Z. Cui, and A. G. Cohn. A spatial logic based on regions and connection. *Principles of Knowledge Representation and Reasoning*, pages 165–176, 1992.

[20] C. Santos, L. Soares, G. de Souza, and J. Courtiat. Design methodology and formal validation of hypermedia documents. In *Proceedings of the sixth ACM International Conference on Multimedia*, pages 39–48, Bristol, United Kingdom, 1998. ACM.

[21] L. F. G. Soares and S. D. J. Barbosa. *Programming in NCL 3.0: developing applications for the Ginga middleware, Digital TV and Web*. Elsevier, 2009. in Portuguese.

[22] L. F. G. Soares, R. F. Rodrigues, and D. C. Muchaluat-Saade. Modeling, authoring and formatting hypermedia documents in the HyperProp system. *Multimedia Systems*, 2000.

[23] W3C. Synchronized Multimedia Integration Language - SMIL 3.0 Specification. http://www.w3c.org/TR/SMIL3, 2008. World-Wide Web Consortium Recommendation.