HAL
open science

An incremental framework for adapting the hypermedia structure of multimedia documents

Sébastien Laborie, Jérôme Euzenat

To cite this version:

Sébastien Laborie, Jérôme Euzenat. An incremental framework for adapting the hypermedia structure of multimedia documents. Manolis Wallace, Marios Angelides, Phivos Mylonas. Advances in Semantic Media Adaptation and Personalization, 93 (8), Springer, pp.157-176, 2008, Studies in computational intelligence, 978-3-540-76359-8. 10.1007/978-3-540-76361_8 . hal-00825954

HAL Id: hal-00825954

https://inria.hal.science/hal-00825954v1

Submitted on 24 May 2013

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

An Incremental Framework for Adapting the Hypermedia Structure of Multimedia Documents

Sébastien Laborie and Jérôme Euzenat

INRIA Rhône-Alpes and LIG, Montbonnot, France
{Sebastien.Laborie,Jerome.Euzenat}@inrialpes.fr

Summary. The multiplication of presentation contexts (such as mobile phones, PDAs) for multimedia documents requires the adaptation of document specifications. In an earlier work, a semantic approach for multimedia document adaptation was proposed. This framework deals with the semantics of the document composition by transforming the relations between multimedia objects. In this chapter, we apply the defined framework to the hypermedia dimension of documents, i.e., hypermedia links between multimedia objects. By considering hypermedia links as particular objects of the document, we adapt the hypermedia dimension with the temporal dimension. However, due to the non-deterministic character of the hypermedia structure, the document is organized in several loosely dependent sub-specifications. To preserve the adaptation framework, we propose a first straightforward strategy that consists of adapting all sub-specifications generated by the hypermedia structure. Nevertheless, this strategy has several drawbacks, e.g., the profile is not able to change between user interactions. Hence, we propose an incremental approach which adapts document sub-specifications step by step according to these interactions. To validate this framework, we adapt real standard multimedia documents such as SMIL documents.

Key words: Qualitative representation and reasoning, SMIL documents.

1 Introduction

A multimedia document may be played on different devices with different capabilities: phones, PDAs, desktop computers, setup boxes, etc. These devices introduce different constraints on the presentation itself. For instance, keyboard limitations (e.g., portable multimedia player) may prevent selecting hypermedia links concurrently. Other constraints may also be introduced by user preferences, content protection or terminal capabilities [20]. The constraints imposed by a client constitute its profile.

To satisfy profiles, multimedia documents must be adapted, i.e., transformed into documents compatible with target contexts before being played. Several kinds of adaptation are possible, such as local adaptation (adaptation of media objects individually) and global adaptation (adaptation of the document structure). This chapter focuses on the latter.

In [6], a framework for adapting multimedia documents based on the qualitative semantics of the documents and constraints was proposed. This approach transforms the relations between multimedia objects and ensures two main properties: (1) that adaptation constraints are satisfied and (2) that the adapted document is as close as possible to the initial document. As an example, for the temporal dimension of documents, this work has been applied to descriptions based on the Allen algebra [2].

In this chapter, we apply this adaptation framework to the hypermedia dimension of documents, i.e., hypermedia links between multimedia objects. By considering hypermedia links as particular objects of the document which belong to multimedia objects, we adapt the hypermedia dimension with the temporal dimension (§4).

However, due to the non-deterministic character of the hypermedia structure, the document is organized in several loosely dependent sub-specifications. To preserve the adaptation framework, we propose a first straightforward strategy that consists of adapting all sub-specifications generated by the hypermedia structure (§5). Nevertheless, we show that this strategy has several drawbacks, e.g., the profile is not able to change between user interactions. Hence, we propose an incremental approach which adapts document sub-specifications step by step according to these interactions.

For a smooth implementation of this adaptation framework, we provide an algorithm which computes adapted multimedia document specifications (§6). Moreover, we validate our adaptation approach by adapting real standard multimedia documents such as SMIL documents [21] (§7).

Let introduce first a multimedia document example (§2) that will be used in this chapter and a current proposed adaptation framework (§3).

2 Multimedia Document Specification

Multimedia documents are composed of various multimedia objects such as texts, images, sounds and videos. These multimedia objects are organized into several dimensions [15]:

- **temporal:** multimedia objects are synchronized.
- **spatial:** some multimedia objects have a spatial layout.
- **logical:** some multimedia objects can be assemble in groups.
- **hypermedia:** the user is able to interact with some multimedia objects.

Figure 1 shows the temporal and the hypermedia dimensions of a movie trailer presentation composed of different multimedia objects like a movie

Poster, a Trailer, an Abstract, the Characters information and the upcoming Dates.

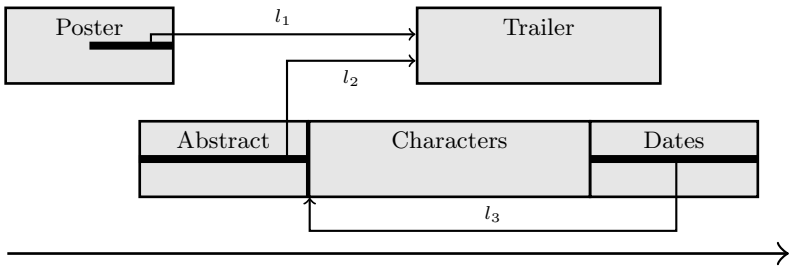


Fig. 1. A multimedia document example.

The hypermedia dimension of multimedia documents is characterized by hypermedia links between multimedia objects. For example in Figure 1, the Poster and the Abstract are linked to the beginning of the Trailer thanks to the hypermedia links l_1 and l_2, respectively. Note that, the hypermedia dimension is not independent from the other dimensions like the temporal one. In Figure 1, suppose the user never selects a hypermedia link, each multimedia object of the presentation is executed. However, if the user selects l_1, the presentation "jump forward" in time to the beginning of the Trailer and the beginning of the Characters information is not presented.

There are several languages or formats for specifying multimedia documents such as SMIL [21], SVG [19], Madeus [9], etc. Making the adaptation format-dependent requires an adaptation strategy for every single format. Instead, we propose an abstraction layer which allows to hide the format specific syntax and details, and capture the essence of a given document with respect to its main dimensions. We call this document abstraction a multimedia document specification.

Definition 1 (Multimedia document specification). *A multimedia document specification $s = \langle O, C \rangle$ is made of a set of multimedia objects O and a set of constraints C between these objects. In the remainder, the constraints will be considered as binary.*

The advantages of such an abstraction and adaptation scheme are twofold. First, it allows to reuse the same strategy for the different languages or formats. Second, the abstract representation provides more flexibility for the adaptation since the relations between multimedia objects can be described qualitatively.

3 Semantic Adaptation of Multimedia Documents

In [6], a semantic approach for multimedia document adaptation is defined. This approach interprets each document as the set of its potential executions (i.e., related to the initial document) and a profile as the set of its possible executions. In this context, "adapting" amounts to find the set of potential executions that are possible. When none is possible, the goal of adaptation is to find possible executions close to potential executions that satisfy the profile.

We consider both the multimedia document specifications and the profiles as a set of relations holding between multimedia objects. The potential and possible executions are ideally represented by relation graphs.

Definition 2 (Relation graph). *Let O be a set of multimedia objects and C a set of constraints between the elements of O. A multimedia document specification $s = \langle O, C\rangle$ relative to a set of executions can be represented as a complete directed labeled graph $g_s = \langle O, E, \lambda\rangle$ called a relation graph. The elements of O are nodes, E is the set of edges and $\lambda : E \rightarrow 2^{\mathcal{R}}$ is a total labeling function from the edges to a set of relations of $\mathcal{R}$ such that for each $x\ r\ y \in C$, $r \in \lambda(\langle x, y\rangle)$.*

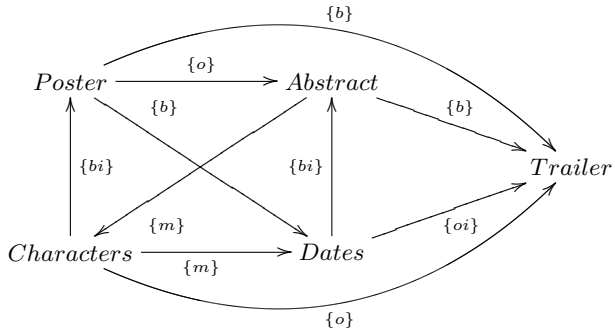
Figure 2 presents two relation graphs. Each node corresponds to a multimedia object and each edge is labeled by a set of relations. In this example, the Allen algebra of temporal interval relations $\mathcal{R}$ [2] is used to describe the temporal dimension. These relations are presented in Table 1.

relation (r): x r y	x / y	inverse: y r^{-1} x
before (b)		(bi) after
meets (m)		(mi) met-by
during (d)		(di) contains
overlaps (o)		(oi) overlapped-by
starts (s)		(si) started-by
finishes (f)		(fi) finished-by
equals (e)		(e)

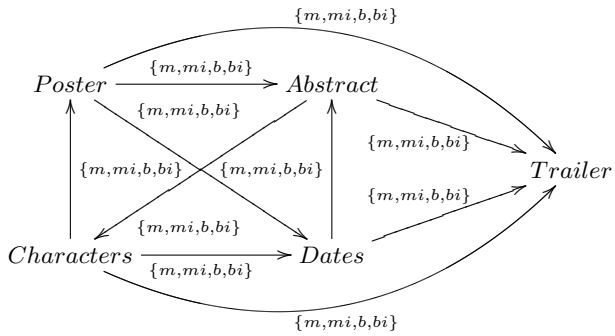
Table 1. The thirteen Allen relations.

The potential relation graph (Fig. 2(a)) includes, in particular, the execution of Figure 1. It corresponds to the initial document. The possible relation graph (Fig. 2(b)) corresponds to the following profile: Impossible for multimedia objects to be played concurrently. It is important to note that it may occur that some potential relations are not possible (e.g., between *Poster* and *Abstract*, and between *Characters* and *Trailer*).

In this context, adapting consists of finding a set of relation graphs corresponding to the possible relation graph (i.e., satisfying the profile) at a minimal distance from the potential relation graph (i.e., the initial document



(a) Potential relation graph.



(b) Possible relation graph.

Fig. 2. Potential and possible executions.

specification). This set of relation graphs is thus called adapted relation graph solutions.

We consider that the proximity between two relation graphs depends on the proximity between relations borne by the same edge in both graphs. This proximity relies on the conceptual neighborhood between these relations and is measured by the shortest path distance in the corresponding conceptual neighborhood graph (Figure 3, [8]).

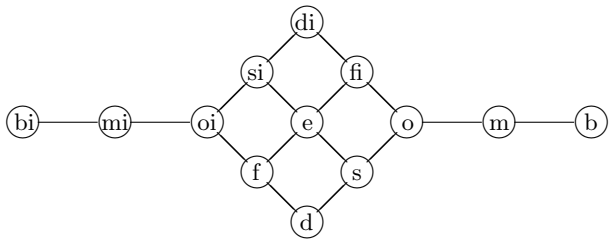


Fig. 3. Allen conceptual neighborhood graph.

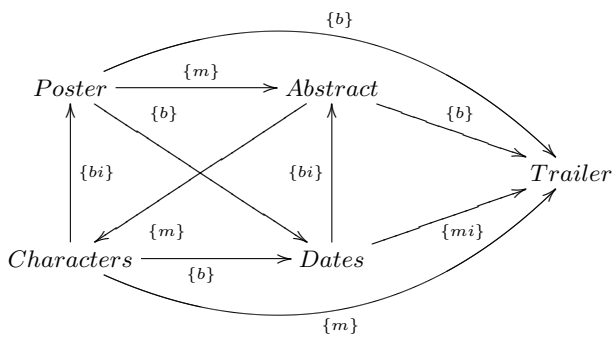
Thus, a distance d between relation graphs is obtained by summing up all the conceptual distances δ between relationships used in both graphs (Def. 3).

Definition 3 (Conceptual distance between two relation graphs).

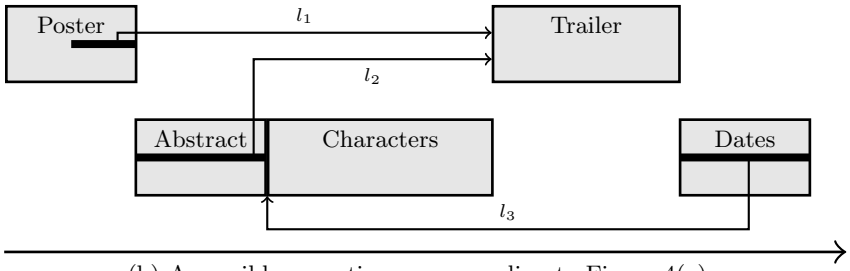
$$d(\lambda, \lambda') = \Sigma_{o_1,o_2 \in O} \mathrm{Min}_{r \in \lambda(\langle o_1,o_2 \rangle),\ r' \in \lambda'(\langle o_1,o_2 \rangle)} \delta(r, r')$$

Figure 4(a) presents an adapted relation graph solution computed from Figure 2(b). Its distance from the initial relation graph of Figure 2(a) is $d = 4$ because the relations between *Poster* and *Abstract*, and between *Characters* and *Trailer* change from o to m ($\delta(o, m) = 1$), the relation between *Characters* and *Dates* changes from m to b ($\delta(m, b) = 1$) and the relation between *Dates* and *Trailer* changes from oi to mi ($\delta(oi, mi) = 1$).

A possible execution corresponding to Figure 4(a) is presented in Figure 4(b). As we may notice, the profile is satisfied and this adapted document is close to the initial one.



(a) An adapted relation graph solution.



(b) A possible execution corresponding to Figure 4(a).

Fig. 4. An adapted relation graph solution and a corresponding execution.

This approach has been fully defined for the temporal [6], spatial [11] and spatio-temporal [10] cases. Our goal is to extend this framework to the hypermedia dimension.

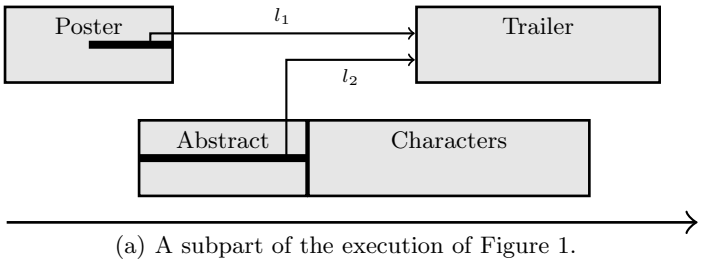
4 Adaptation of the Hypermedia Dimension

As shown in Figure 1, each hypermedia link of a multimedia document is related to a multimedia object. Moreover, hypermedia links possess their own execution time inside their related multimedia object. For example in Figure 1, l_1 is a hypermedia link belonging to *Poster* and is active during a part of its execution, while l_2 belongs to *Abstract* and is active during all its execution.

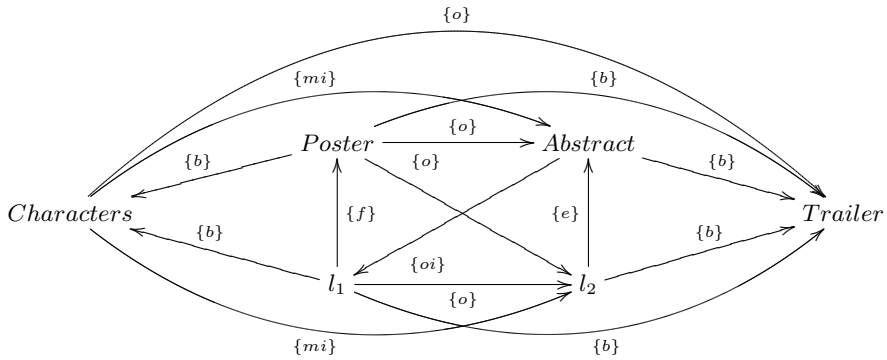
Hence, hypermedia links can be considered as particular objects of the document specification. Thus, we enhance in the following definition the multimedia document specification of Definition 1 with hypermedia links.

Definition 4 (Multimedia document specification with hypermedia). *Let O_m be a set of multimedia objects and O_l a set of hypermedia links. $s = \langle O, C \rangle$ is a multimedia document specification enhanced with hypermedia links with $O = O_m \cup O_l$ the set of objects of the document and C the set of constraints between the elements of O. To express that a link $l \in O_l$ belongs to a multimedia object $m \in O_m$, the constraint between l and m should express such property, e.g., for the temporal case l is active only when m is executed.*

Figure 5 presents a subpart of the document presented in Figure 1.



(a) A subpart of the execution of Figure 1.



(b) The potential relation graph of Figure 5(a).

Fig. 5. A subpart of the document of Figure 1.

The upcoming Dates object is voluntary omitted for visibility reason.

Figure 5(b) is the potential relation graph corresponding to the execution presented in Figure 5(a). Note that the relation graph is enhanced with hypermedia links, i.e., l_1 and l_2. Moreover, the relation between l_1 and *Poster* and the one between l_2 and *Abstract* state that a hypermedia link is active only when its corresponding multimedia object is executed.

Thanks to the multimedia specification extension defined in Definition 4, the adaptation framework described in Section 3 can be used to adapt the hypermedia dimension of documents. Let adapt Figure 5 with the following profile: impossible to select hypermedia links concurrently, e.g., because the device has only one button.

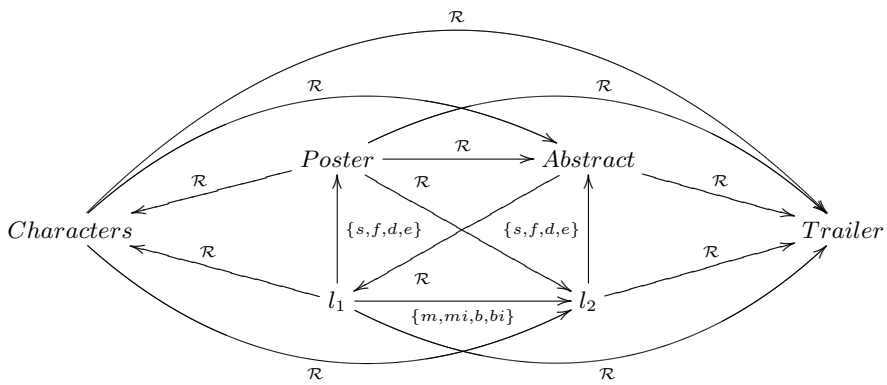
Figure 6(a) presents the possible relation graph corresponding to this profile. The set of relations $\{m, mi, b, bi\}$ between l_1 and l_2 is due to the profile. When no adaptation constraint is expressed in the profile between two objects, we consider that all relations $\mathcal{R}$ are possible. For example, no adaptation constraint is expressed in the profile between *Poster* and *Abstract*, thus all relations $\mathcal{R}$ are possible between these two objects.

Moreover, in the relation graph of Figure 6(a) the constraint between a hypermedia link and its related multimedia object is satisfied. For example, the hypermedia link l_2 belongs to *Abstract*, and the only allowed relations in time between l_2 and *Abstract* are starts (s), finishes (f), during (d) and equals (e), i.e., inclusive temporal relations.

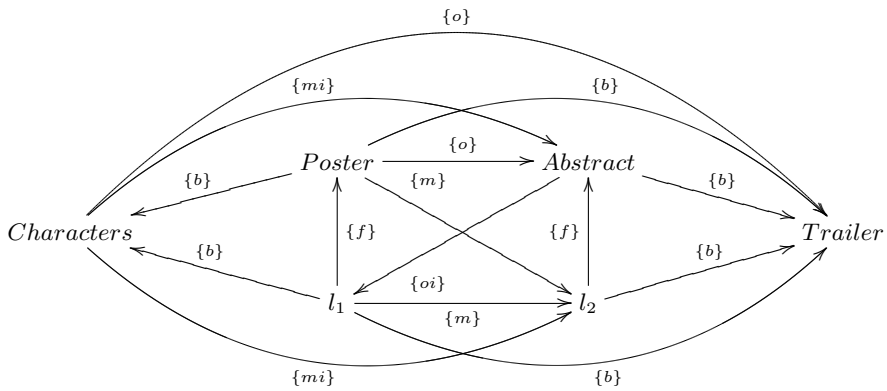
Figure 6(b) presents an adapted relation graph solution computed from Figure 6(a) (cf., Section 3). Its distance from the initial relation graph of Figure 5(b) is $d = 3$ because the relation between l_1 and l_2 changes from o to m ($\delta(o, m) = 1$), the relation between l_2 and *Abstract* changes from e to f ($\delta(e, f) = 1$) and the relation between *Poster* and l_2 changes from o to m ($\delta(o, m) = 1$).

A possible execution corresponding to Figure 6(b) is presented in Figure 6(c). As we may notice, the profile is satisfied and this adapted document is close to the initial one.

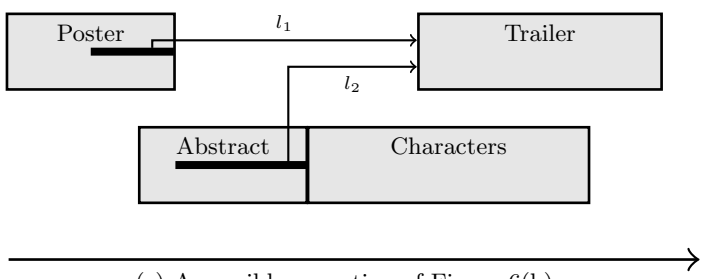
In this section, we adapt with the proposed adaptation framework of Section 3 the hypermedia dimension of multimedia documents by considering hypermedia links as particular objects which belong to multimedia objects. We mix this dimension with the temporal one providing a temporal-hypermedia adaptation. However, due to the non-deterministic character of the hypermedia structure involved by hypermedia links, several loosely dependent sub-specifications have to be considered. In the next section, we propose an incremental approach for adapting these sub-specifications according to the user interactions.



(a) Possible relation graph.



(b) Adapted relation graph solution.



(c) A possible execution of Figure 6(b).

Fig. 6. Adaptation of Figure 5 with the following profile: impossible to select hypermedia links concurrently.

5 Adaptation based on the Hypermedia Structure

A multimedia document may be composed of several hypermedia links. Each hypermedia link points to a particular time in the presentation. For example, in Figure 1 the hypermedia links l_1 and l_2 point to the beginning of the *Trailer*, while l_3 points to the beginning of *Characters*.

Hence, several multimedia document sub-specifications can be identified. They correspond to the hypermedia structure of the document. For example, Figure 7 presents the hypermedia structure of Figure 1. Each node represents a multimedia document sub-specification. In this figure, they are three sub-specifications, namely s_1, s_2 and s_3. For a better understanding, for each sub-specification we show a possible execution. Each edge is labeled by a set of hypermedia links which leads from one sub-specification to another.

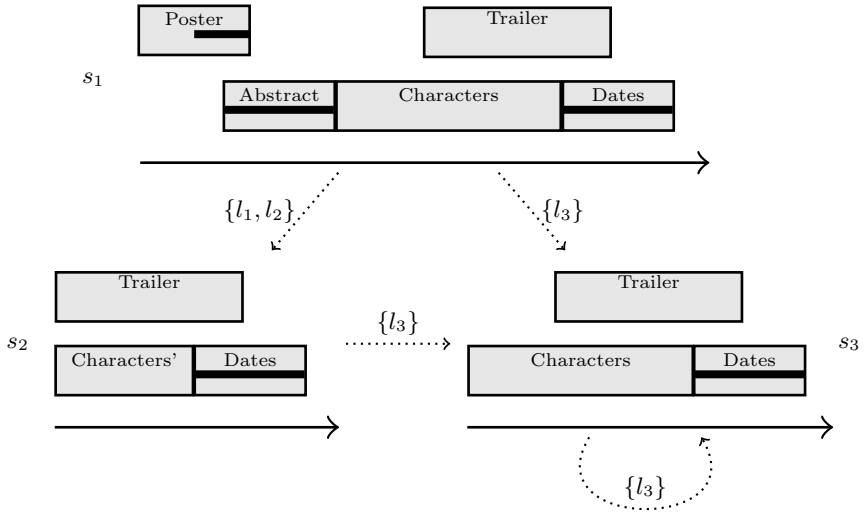


Fig. 7. The hypermedia structure of Figure 1.

We propose, in Section 5.1, a method which build from an initial document specification several sub-specifications corresponding to its hypermedia structure. Thereafter, in Section 5.2, we propose a first straightforward strategy that consists of adapting at once all sub-specifications generated by the hypermedia structure. However, several drawbacks are identified. Hence, in Section 5.3, we propose an incremental adaptation which adapts sub-specifications step by step according to the user interactions.

5.1 Generating the multimedia document sub-specifications according to the hypermedia structure

From an initial document specification, several sub-specifications can be computed according to its hypermedia links. For example, Figure 1 is an initial document specification. If the hypermedia link l_1 is selected, the presentation "jump" in time to the beginning of the *Trailer*. It is thus useless to preserve in the future sub-specification all objects that are before this moment.

Figure 8 presents how s_2 is constructed from the initial document specification of Figure 1. As we may notice, when l_1 is selected, the objects *Poster*, *Abstract*, l_1 and l_2 are useless in the future sub-specification. Moreover, the multimedia object *Characters* is splitting into two parts, i.e., one useless and one presented. In this case, this object is transformed into a partition of *Characters*, called *Characters'*, and starts the present time (cf., s_2 in Figure 7).

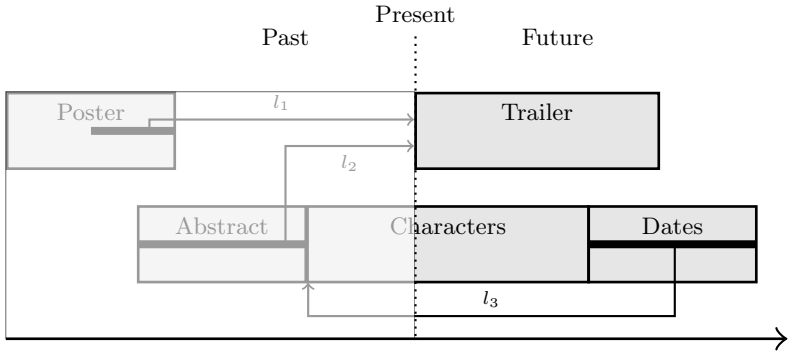


Fig. 8. Building the sub-specification s_2.

This process is done for each hypermedia link of the initial document specification. In the next section, we propose to adapt these sub-specifications.

5.2 Adapting all sub-specifications at once

Each document sub-specification of the hypermedia structure does not contain the same number of objects and the same constraints between them. Moreover, the adaptation framework presented in Section 4 never suppresses objects. Thus, it is difficult to group in one global document specification all sub-specifications generated by the hypermedia structure without any modification of our adaptation approach.

Hence, to preserve our framework, defined in Section 4, we propose the following straightforward strategy:

Suppose p a given profile,
for each sub-specification s_i of a hypermedia structure, $Adapt(s_i, p)$.

For example, suppose the following profile: impossible to play hypermedia links concurrently with other objects (we do not consider the multimedia object related to the hypermedia link). Figure 9 presents the adapted hypermedia structure of Figure 1. In the adapted sub-specification s'_1, the hypermedia link duration of l_1, l_2 and l_3 is reduced, while in s'_2 and s'_3 only the hypermedia link duration of l_3 is reduced.

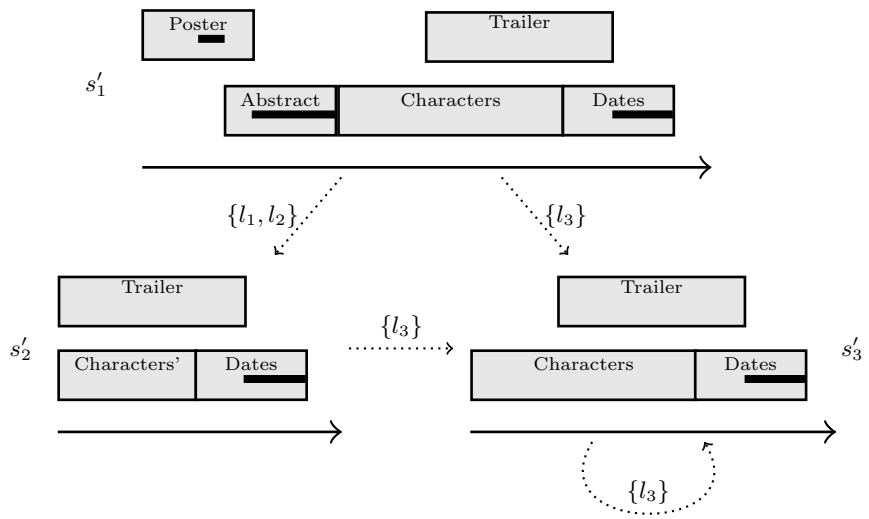


Fig. 9. An adapted hypermedia structure of Figure 1.

This strategy effectively produces an adapted document where all its sub-specifications satisfy the profile and are close to the initial ones. However, this strategy has several drawbacks:

- Some models are adapted even if they are not executed. For example in Figure 7, if the user never selects a hypermedia link, it would be more efficient to adapt only s_1.
- If the profile p changes between each user interaction, all sub-specifications s_i have to be adapted again.
- Parts of several sub-specifications which are identical are adapted several times. For example in Figure 7, s_1 and s_3 have identical parts and are adapted twice (see s'_1 and s'_3 in Figure 9).

Hence, we propose in the following section to adapt the sub-specifications of the hypermedia structure step by step according to the user interactions.

5.3 Adapting sub-specifications step by step

[6] proposed to compute adapted documents close to the initial one which satisfy the profile. Based on the hypermedia structure, we propose in this section to compute adapted sub-specifications close to the previous adapted one if the profile do not changed, and close to the initial one if the profile changed. Our aim is to adapt only subparts of the document that is going to be presented. Let call this adaptation incremental adaptation.

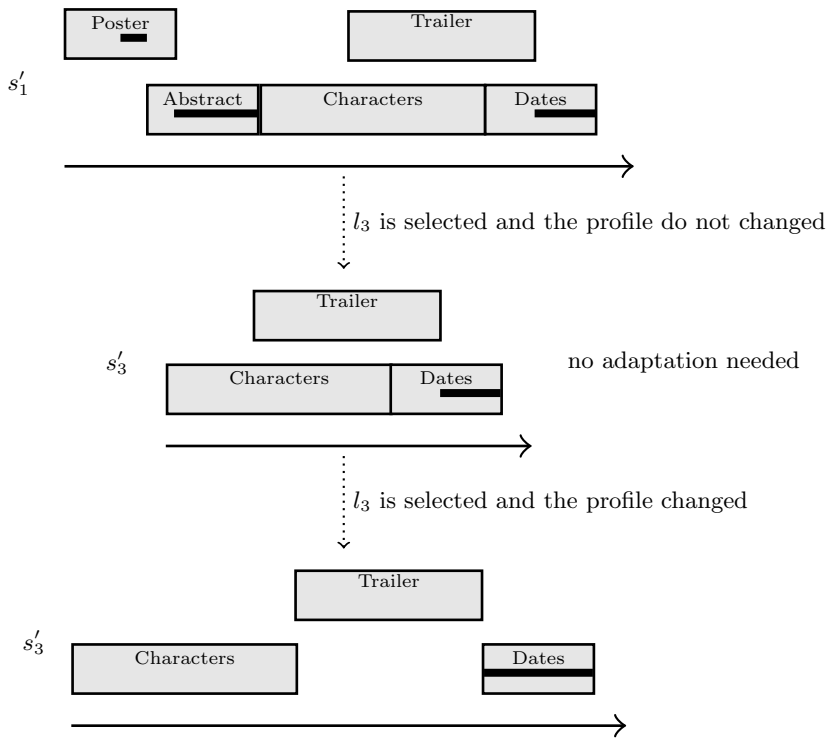


Fig. 10. An incremental approach for adapting the hypermedia structure of Figure 1.

Initially, we compute and adapt a first initial sub-specification. We choose the one executed when the user never selects a hypermedia link. This leads to a first adapted sub-specification. For example, s_1 (Figure 7) is the first initial sub-specification and p a given profile (the same as in Section 5.2). We adapt s_1 according to the profile p, i.e., $Adapt(s_1, p) = s'_1$ (Figure 10 top).

If a hypermedia link is selected by the user and the profile p do not changed, we compute the next corresponding sub-specification according to the previous adapted sub-specification. For example in Figure 10, suppose during the execution of s'_1 the user selects l_3, we compute s_3 according to s'_1. This compu-

tation propagates the adapted information of s'_1 into s_3. Thereafter, we adapt s_3 according to the profile p, i.e., $Adapt(s_3, p) = s'_3$ (Figure 10 middle).

If a hypermedia link is selected by the user and the profile p changed into p', e.g., impossible to play multimedia objects concurrently, we initialize the corresponding sub-specification according to the initial document specification and adapt it. For example in Figure 10, suppose during the execution of s'_3 (computed previously) the user selects l_3, we initialize s_3 according to the initial document specification. Thereafter, we adapt s_3 according to the profile p', i.e., $Adapt(s_3, p') = s'_3$ (Figure 10 bottom). Note this time that l_3 is not affected by the adaptation, thus to be close to the initial document.

This strategy is repeated each time a hypermedia link is selected.

With this incremental adaptation, we only adapt sub-specifications which are executed, the profile is able to change between each user interaction, and we propagate the adapted information between each sub-specification.

6 Implementation

For a smooth implementation of our adaptation framework, we present in this section the *Adapt* algorithm (Alg. 1) which computes adapted relation graph solutions. This algorithm is Nebel's backtracking algorithm [14], which enumerates consistent relation graphs, enhanced with a Branch and Bound optimization [12] for adaptation.

Algorithm 1: Adapt

```
Input: An initial matrix I_{i,j} corresponding to potential executions and
       a matrix P_{i,j} corresponding to possible executions.
Data: S is a set of adapted relation graphs; Min is a current minimum
      computed distance (initialized with a high value).
pathConsistency(P);
if P does not contain an empty relation then
    Choose an unprocessed label P_{i,j} and split P_{i,j} into r_l = r_1, ..., r_k;
    if no label can be split then
        tmp ← d(I, P);
        if tmp < Min then
            Min ← tmp; S ← {P};
        if tmp = Min then
            S ← S ∪ {P};
    else
        for all labels r_l (1 ≤ l ≤ k) do
            P_{i,j} ← r_l;
            if d(I, P) ≤ Min then
                Adapt(I,P);
```

The relation graph corresponding to potential executions (i.e., the initial document specification) and the one corresponding to possible executions (i.e., satisfying the profile) are encoded into two matrix $I_{i,j}$ and $P_{i,j}$, respectively.

Before the $Adapt(I_{i,j}, P_{i,j})$ call, we first sort each label of the matrix $P_{i,j}$ according to the distance δ (cf., Section 3) from each label of the matrix $I_{i,j}$.

Thereafter, Alg. 1 computes consistent possible relation graphs with help from the *pathConsistency* function [2]. Each time consistent possible relation graphs are computed, we select the minimal ones in $\mathcal{S}$ thanks to the distance d defined in Definition 3. When Alg. 1 stops, $\mathcal{S}$ is the set corresponding to adapted relation graph solutions and Min the minimal distance from the initial document specification.

7 Application to Concrete Multimedia Documents

We want to validate our adaptation framework on real standard multimedia documents such as SMIL documents [21]. We present, in Section 7.1, a strategy which generalizes a multimedia document, adapts it with the adaptation framework defined previously, and translates it back to an adapted document. This strategy can be applied to any multimedia document language. In Section 7.2, we present an interactive prototype which allows users to create SMIL documents, specify adaptation constraints and adapt these documents.

7.1 Concrete multimedia documents adaptation

The natural way of using our adaptation framework, presented in the previous sections, for adapting actual documents, consists in taking the equivalent qualitative representation of the document (α), adapting it and translating back to an adapted document (β). Figure 11 presents this straightforward strategy.

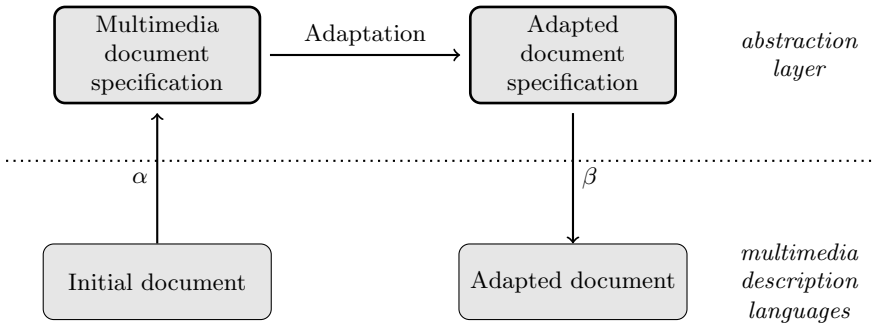


Fig. 11. Concrete multimedia documents adaptation.

The strategy presented above, if it is effective, does not apply to a particular multimedia specification language. It must be precised for each particular language. We apply it to SMIL documents.

SMIL is an XML-based language for specifying multimedia presentations [21]. Figure 12 presents a SMIL specification of Figure 1. As we may notice, the temporal behavior of a SMIL document is not expressed with Allen relations, as used in the previous sections, but time containers and attributes for objects synchronization.

```
<smil xmlns="http://www.w3.org/2001/SMIL20/Language">
  <head>
    <layout>
      <root-layout background-color="#000000" height="400" title="Movie" width="700"/>
      <region height="400" id="Poster" left="0" top="0" width="700" z-index="1"/>
      <region height="400" id="Abstract" left="0" top="0" width="700" z-index="2"/>
      <region height="200" id="Trailer" left="175" top="0" width="350" z-index="3"/>
      <region height="200" id="Characters" left="0" top="200" width="700" z-index="3"/>
      <region height="400" id="Dates" left="0" top="0" width="700" z-index="4"/>
    </layout>
  </head>
  <body>
    <par>
      <a href="#trailer" begin="3s" dur="3s">
        <img id="poster" dur="6s" region="Poster" src="poster.png"/>
      </a>
      <video id="trailer" begin="15s" end="43s" region="Trailer" src="trailer.avi"/>
      <seq begin="4s">
        <a href="#trailer">
          <img id="abstract" dur="6s" region="Abstract" src="abstract.png"/>
        </a>
        <video id="characters" region="Characters" src="character.avi"/>
        <a href="#characters">
          <img id="dates" dur="6s" region="Dates" src="dates.png"/>
        </a>
      </seq>
    </par>
  </body>
</smil>
```

Fig. 12. A SMIL specification of Figure 1.

Hence, the goal of the α function is the extraction of the relational structure involving objects of the document and its encoding as a relation graph on which to apply the previous defined adaptation operations. This extraction is obtained by the following procedure:

1. extract all multimedia objects and hypermedia links, and make them a node in the potential relation graph;
2. add an edge between each pair of nodes (they are labeled by a set of relations $\mathcal{R}$);
3. extract the relations implied by the multimedia objects and hypermedia links synchronization, e.g., the time containers, the objects attributes, etc.

Thanks to the qualitative representation built from α, it is thus possible to adapt it with the adaptation framework described in the previous sections,

e.g., for adapting hypermedia links. Thereafter, we need to inject the adapted information inside the document.

For that purpose, the β function can be defined in a straightforward manner:

1. propagate all adapted constraints from the qualitative adapted representation to the document, i.e., update the multimedia objects and hypermedia links synchronization (e.g., the time containers, the objects attributes).

7.2 Prototype

The presented adaptation framework has been implemented in an interactive adaptation prototype. Its architecture is based on Figure 11 and used Alg. 1 to compute adapted relation graph solutions. Figure 13 presents a screenshot of the prototype.

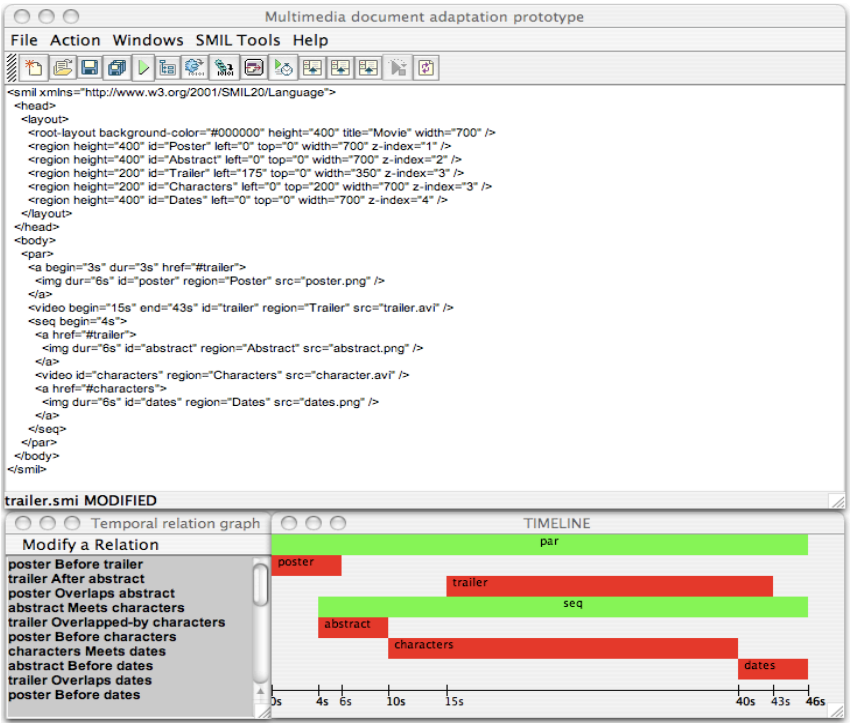


Fig. 13. A screenshot of our prototype

The author can edit his SMIL document via the editor. Then, the relation graph corresponding to his document can be computed by the system.

Moreover, it is possible to indicate to the system adaptation constraints (i.e., specify the profile).

On one hand, if the multimedia document does not need to be transformed, the adapted SMIL document is identical to the initial one. On the other hand, if the document needs to be adapted, the system presents all adapted relation graph solutions, i.e., those which are at the minimal distance from the initial relation graph and satisfy the profile.

Once an adaptation solution is chosen, the system provides an adapted SMIL document and the user can execute it.

8 Related Work

The SMIL language [21] defines a `switch` element that allows authors to specify a set of presentation alternatives from which only the first acceptable one is chosen. Nevertheless, it is necessary to know in advance the different target profiles. Furthermore, authors should order the alternatives from the most desirable to the least desirable.

[4, 5] consider the adaptation of the hypermedia dimension of multimedia documents. They describe methods and techniques for adapting the hypermedia structure. Note that the latter considers SMIL hypermedia adaptation. However, these frameworks only perform link annotation and/or link hiding.

[7] define a spatial hypermedia adaptation framework. Like our adaptation approach they use an abstract representation of documents. Nevertheless, they do not include in their adaptation approach other dimensions like the temporal one.

[1] present a Petri net model for adapting hypermedia presentations. Like our hypermedia adaptation framework presented in this chapter, they mix the temporal and the hypermedia dimensions. However, they only drop multimedia objects to satisfy profiles.

The works described in [3, 13] adapt SMIL documents. [3] adapts the temporal and spatial structure using semantic dependencies between multimedia objects. [13] seeks to compute efficiently an adapted document satisfying a limited form of adaptation constraints, namely the translation of SMIL documents between the different language profiles (i.e., a form of syntactic restrictions of the different SMIL dialects [21]). Our adaptation approach differs in that it operates at a semantic level and is therefore capable of adapting documents in the same language or between different languages or formats. Moreover, our adaptation process produces documents which are as close as possible from the initial ones.

Other approaches for multimedia document adaptation are based on specialized specifications for generating documents such as [17, 18]. However, this requires the content to be initially described with these specifications instead of adapting existing documents.

9 Conclusion

In this chapter, we applied a semantic adaptation framework to the hypermedia dimension of documents by considering hypermedia links as particular objects of the presentation. Moreover, to adapt the several document subspecifications generated by the hypermedia structure and preserve our approach, we proposed an incremental adaptation which takes into account the user interactions. This adaptation framework does not apply to a particular multimedia description language but is generic. In order to validate it we implement a prototype that relies on adapting SMIL documents.

In the future, we want to refine the adaptation by selecting suitable adapted solutions. Indeed, several adapted relation graph solutions can be computed. Currently, it is difficult to select a particular solution from this set because all solutions are close to the initial document. Using the rhetorical structure of multimedia documents, as [16] did for generating multimedia content, could be a promising direction for refining our adaptation framework.

Moreover, if the profile is highly-constrained, the adaptation may produce no solution. In such a case, we have to delete or transform (in terms of transcoding) objects of the document, thus requiring new proximity measures. Hence, it would be possible to modify the hypermedia structure of documents.

To conclude, we also plan to extend our adaptation framework to all multimedia document dimensions, thus providing a spatio-temporal-hypermedia adaptation.

As discussed above, there remains more work to be carried out for covering all the aspects of multimedia documents and for deepening the specification of documents and adaptation constraints so that the adaptation produces quality results.

References

1. Nabil R. Adam, Vijayalakshmi Atluri, Igg Adiwijaya, Sujata Banerjee, and Richard Holowczak. A dynamic manifestation approach for providing universal access to digital library objects. *IEEE Transactions on Knowledge and Data Engineering*, 13(4):705–716, 2001.
2. James Allen. Maintaining knowledge about temporal intervals. *Communications of the ACM*, 26(11):832–843, 1983.
3. Mariam Kimiaei Asadi and Jean-Claude Dufourd. Knowledge-based and semantic adaptation of multimedia content. In Paola Hobson, Ebroul Izquierdo, Yiannis Kompatsiaris, and Noel E. O'Connor, editors, *Knowledge-Based Media Analysis for Self-Adaptive and Agile Multimedia Technology*, pages 285–293, 2004.
4. Paul De Bra, Peter Brusilovsky, and Geert-Jan Houben. Adaptive hypermedia: from systems to framework. *ACM Computing Surveys*, 31(4), 1999.
5. Paul De Bra and Natalia Stash. Multimedia adaptation using AHA! In *Proceedings of the World Conference on Educational Multimedia, Hypermedia and Telecommunications*, pages 563–570, 2004.

6. Jérôme Euzenat, Nabil Layaïda, and Victor Dias. A semantic framework for multimedia document adaptation. In *Proceedings of the 18th International Joint Conferences on Artificial Intelligence*, pages 31–36. Morgan Kauffman, 2003.
7. Luis Francisco-Revilla and Frank Shipman. MASH: A framework for adaptive spatial hypermedia. In *Proceedings of the Third Workshop on Spatial Hypertext, ACM Hypertext Conference*, 2003.
8. Christian Freksa. Temporal reasoning based on semi-intervals. *Artificial Intelligence*, 54(1–2):199–227, 1992.
9. Muriel Jourdan, Nabil Layaïda, Cécile Roisin, Loay Sabry-Ismaïl, and Laurent Tardif. Madeus, an authoring environment for interactive multimedia documents. In *Proceedings of the 6th ACM Multimedia conference*, pages 267–272, 1998.
10. Sébastien Laborie. Spatio-temporal proximities for multimedia document adaptation. In *Proceedings of the 12th International Conference on Artificial Intelligence: Methodology, Systems, Applications*, pages 128–137, 2006.
11. Sébastien Laborie, Jérôme Euzenat, and Nabil Layaïda. A spatial algebra for multimedia document adaptation. In *Poster Proceedings of the First International Conference on Semantic and Digital Media Technologies*, pages 7–8, 2006.
12. A. H. Land and A. G. Doig. An automatic method for solving discrete programming problems. *Econometrica*, 28(3):497–520, 1960.
13. Tayeb Lemlouma and Nabil Layaïda. The negotiation of multimedia content services in heterogeneous environments. In *Proceedings of the 8th International Conference on Multimedia Modeling*, pages 187–206, 2001.
14. Bernhard Nebel. Solving hard qualitative temporal reasoning problems: Evaluating the efficiency of using the ORD-horn class. In *Proceedings of the 12th European Conference on Artificial Intelligence*, pages 38–42, 1996.
15. Cécile Roisin. Authoring structured multimedia documents. In *Proceedings of the Conference on Current Trends in Theory and Practice of Informatics*, pages 222–239, 1998.
16. Lloyd Rutledge, Brian Bailey, Jacco van Ossenbruggen, Lynda Hardman, and Joost Geurts. Generating presentation constraints from rhetorical structure. In *Proceedings of the 11th ACM on Hypertext and hypermedia*, pages 19–28. ACM Press, 2000.
17. Ansgar Scherp and Susanne Boll. mobileMM4U – framework support for dynamic personalized multimedia content on mobile systems. In *Proceedings of Techniques and Applications for Mobile Commerce (TaMoCO)*, pages 204–215, 2004.
18. Jacoo van Ossenbruggen, Frank J. Cornelissen, Joost Geurts, Lloyd Rutledge, and Lynda Hardman. Cuypers : a semi-automatic hypermedia generation system. Technical Report INS-R0025, CWI, Amsterdam (NL), 2000.
19. W3C. *Scalable Vector Graphics (SVG)*, 1999. http://www.w3.org/Graphics/SVG/.
20. W3C. *Composite Capability/Preference Profiles (CC/PP): Structure and Vocabularies*, 2001. http://www.w3.org/TR/CCPP-struct-vocab/.
21. W3C. *Synchronized Multimedia Integration Language (SMIL 2.0) Specification*, 2001. http://www.w3.org/TR/smil20/.