Paving the Last Mile for Multi-Channel Multimedia Presentation Generation

Angsar Scherp
OFFIS Research Institute, Germany
ansgar.scherp@offis.de

Susanne Boll
University of Oldenburg, Germany
susanne.boll@informatik.uni-oldenburg.de

Abstract

Users of multimedia applications today are equipped with a variety of different (mobile) devices that each come with different operating systems, memory and CPU capabilities, network connections, and also different software such as multimedia players. To be able to efficiently deliver appealing multimedia presentations to all the users, we need to overcome the last mile in multimedia presentation delivery and meet the different requirements at the end user's site. Consequently, one needs to provide multi-channel multimedia presentation generation such that all different users can get and use it in their individual device configuration. With our approach, we aim at developing an abstract multimedia content model that embeds the central characteristics of today's multimedia presentation formats: the definition of the temporal and spatial layout as well as the interaction possibilities. The abstract model allows to be easily transformed to different multimedia presentation formats on different devices and by this serve different output channels. We present our abstract multimedia model and transformation process that allows to compose and to generate suitable multimedia presentations for each different channel such as a SMIL 2.0 presentation created for rendering on a PC or a SVG Tiny presentation designed for a mobile device. With our implementation, application developers can efficiently realize on-the-fly multi-channel generation of multimedia content.

Keywords: multimedia document models, multimedia presentation formats, multimedia content transformation, multimedia presentation generation

1. Introduction

Multimedia has long achieved acceptance for the conveyance of information. Examples of the wide spectrum of applications are advertisement, product presentations, tourist information, e-learning units, and virtual museums. The problem more and more applications share is that they face a big variety of heterogeneous devices, platforms, multimedia presentation formats, and players for multimedia. Though we see a predominance of some platforms and multimedia formats it is a clear observation that not one single platform or multimedia format will prevail but rather a set of device settings and configurations will peacefully coexist. The question is, how can multimedia applications solve the tough and expectedly expensive task of creating and delivering multimedia content that meets all the different individual user's settings?

In this paper, we present our approach to pave this last mile for multi-channel multimedia presentation generation, i.e., the delivery of multimedia content to a wide range of end devices, including desktop PCs, laptops, and other mobile devices like PDAs, and cell phones, as well as a wide range of multimedia presentation formats, that are supported by the respective end device. Our general approach here is to dynamically generate multimedia presentations for the different platforms in different formats using an abstract multimedia content model as basis. This abstract model integrates the different characteristics of the existing standard models in regard of the central aspects of multimedia modeling. These central aspects are the definition of the temporal model and visual layout of the multimedia presentation as well as the interaction possibilities provided by it [1]. The automatic transformation of multimedia content represented in this abstract model into the different presentation formats allow the application developers to rely on existing players on the market, focus on the multimedia content creation task, and leave the creation of the final presentations to our multi-channel presentation generation. The abstract content model and the transformation process presented in this paper are embedded in our software framework for the dynamic generation of highly personalized multimedia content for the individual user, which is described in detail in [2]. By this, we provide support for multimedia applications to overcome the last mile in creating and delivering appealing multimedia content to the customer's device and context.

The following Figure 1 illustrates the general multimedia content creation chain. Typically, media elements such as images, text, audio, and video are selected and then assembled in an internal content model. This model captures the different aspects of the multimedia presentation such as temporal course, spatial layout, and interaction possibilities without instantiating these in a concrete syntax and format. Only then in a transformation phase – the last mile – the final presentation format is created and delivered to the client for the actual rendering.

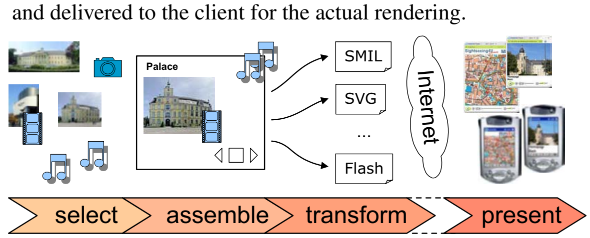

Fig. 1. Multimedia content generation chain.

The multimedia presentation formats we find today at the user's site are W3C standards (available from [3]) like for example HTML+TIME, SVG, SMIL and their different language profiles but also commercial standards like Macromedia's Flash [4]. For the design of our abstract model, we analyze these formats w.r.t. the central aspects of multimedia presentations to embrace their individual characteristics in our abstract model. Our earlier comparison and analysis of multimedia document models [1] had its focus on the specific compositional features of the model rather than finding an abstract content model. Research approaches for the sophisticated modeling of multimedia presentations such as Madeus [5], CMIF [6], and AHM [7] have been contributing to the different facets of composing complex multimedia content but are less significant for our work, since there is no or only limited multimedia player support for the different end user's devices today. Research projects that work on abstract internal models are for example Cuypers and WAM: Cuypers [8] focuses on the high level modeling and sophisticated creation of personalized multimedia content, however, remains with the delivery of SMIL content for desktop PCs for the final presentation. The WAM [9] project works on transforming XML based multimedia content into SMIL with the specific focus of negotiating and meeting the technical constraints of the client device. Supporting a variety of different platforms and formats is not in their focus. The work in [10] proposes XMT [11] for cross-standard interoperability and remains in the field of SMIL-near multimedia document models. Cross-standard transformation among different standards can not be done lossless and favors often one of the standards, here, SMIL.

We present our analysis of the existing standard formats we find on the heterogeneous devices and derive our format-independent multimedia content model in Sections 2 and 3. The multi-channel transformation of this abstract model into the final multimedia presentation formats is described in Section 4. Our implementation and application is presented in Section 5 before we conclude the paper in Section 6.

2. Analysis of standard presentation formats

To achieve the goal to generate content in different existing multimedia document formats we analyzed standard multimedia presentation formats. Our earlier comparison of existing formats [1] was directed towards identifying and comparing multimedia modeling features of existing standard and research multimedia document models. However, in this paper we aim at finding an abstract multimedia content model that joins the format-spanning central aspects of the different formats. The resulting abstract model allows for its transformation into all the different final presentation formats.

At this point, we could have considered to use one of the existing sophisticated multimedia document models as basis to generate all the other presentation formats. But this would not assure that the syntax of the selected model can be optimally transformed to the syntax of the different presentation formats. As the subsequent analysis shows, modern presentation formats such as SMIL and SVG unfortunately intermix the different aspects of multimedia composition, e.g., the definition of media elements and their temporal synchronization or spatial layout. Consequently, we rather aim to design an abstract multimedia content model that can be transformed into the different formats. To reach this goal to provide an optimal support for all the different formats it is crucial to analyze them first and to understand their structure.

Multimedia applications today can use many different presentation formats to display the multimedia content on the user's end device. For our analysis, we selected today's state-of-the-art presentation formats like SMIL and its derivates, SVG and Mobile SVG, Flash, HTML+TIME, and MPEG-4's XMT-Ω. We analyzed these formats in regard of the very central aspects of multimedia modeling: the definition of the temporal model and visual layout as well as the interaction possibilities provided by the formats. Within our analysis we consider in detail how the syntactic elements of the presentation formats support each of these aspects. So, in contrast to the research activities so far, where usually the ideas and requirements in regard of the desired multimedia composition functionality result in a proper document model, we perform a kind of backwards analysis starting at the state-of-the-art of today's presentation formats to obtain the required characteristics and multimedia composition features of a presentation format independent content model.

In the following, the results of our analysis in regard of the temporal model, the spatial model, and the interaction possibilities of the formats are presented. These results may be already known to the reader but are very important w.r.t. the definition of our abstract content model in Section 3 and its subsequent transformation into the different presentation formats described in Section 4.

Temporal models: A temporal model describes the temporal dependencies between the media elements within a multimedia document. As depicted in Figure 2 by a small slideshow example, the temporal model can be realized by defining absolute positioning in time on a global timeline. Here, all media elements used for the slideshow are placed on a global timeline by distinct points in time, which always refer to the zero point of the global time axis. This concept is realized for example by the SVG family.

The other possibility is positioning of the media elements on local timelines. They no longer refer to the zero point of the global time axis but to the local timeline. Local timelines can be nested to build arbitrary relative temporal dependencies between the presentation's media elements. In our example, each slide has its own local timeline where the corresponding image and text elements are placed. These local timelines are then arranged on the global one. The concept of local timelines is supported, e.g., by SMIL and XMT-Ω.

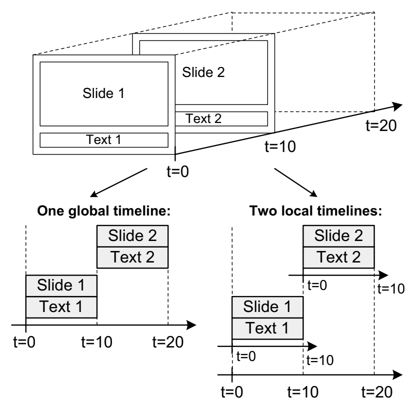

Fig. 2. Example of a slideshow to illustrate global and local points in time.

Spatial models: For visual media elements multimedia presentation formats need the concept of spatial positioning. They allow absolute or relative positioning of visual media elements. The difference between the two is depicted in Figure 3. The top of the figure shows a schematic diagram of a sightseeing application and the visual media elements that are used for it. In the case of absolute positioning the visual media elements always refer to the origin of the coordinate system of the entire multimedia presentation. With relative positioning visual media elements are placed within local areas. In the example given, the medium "Sight's name" and "Image of sight" belong to the coordinate system of the "Info panel". These logical areas can be nested to build arbitrary spatial dependencies between the visual media elements.

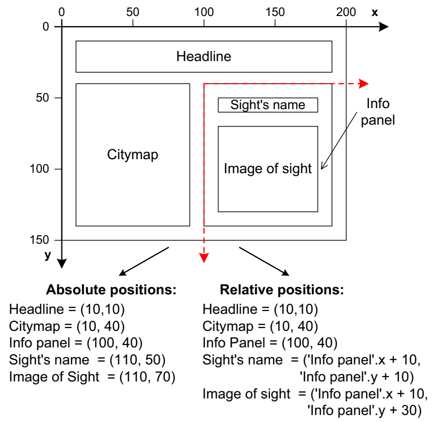

Fig. 3. Example of a sightseeing application to illustrate absolute and relative positioning in space.

Interaction possibilities: The third central aspect in multimedia presentations is navigational interaction. Navigational interaction allows to select one out of many presentation paths that can be followed. This interaction type is realized by, e.g., interactive hyperlinks, menus, and hotspots. Common to these different elements is that they represent one or more interactive external or internal links: External links refer to external multimedia presentations or to a particular part within. Internal links refer to a distinct element within the same presentation. Following an internal link means to jump to a specific point in time on the multimedia presentation's timeline. All considered formats support unidirectional internal and external links. In literature, we find further interaction types such as design interaction and movie interaction [1, 12]. These are typically not defined within today's presentation formats and realized by the multimedia players, e.g., the movie interaction of a VCR-like control. The interaction model covered in this paper considers the navigational interaction only.

Summary of presentation format characteristics: Table 1 gives a summary of the analyzed presentation formats and their individual characteristics in regard of the discussed temporal, spatial, and interaction model. Most of the formats use local timelines as their temporal model. Only SVG and its profiles provide a temporal model based on a global timeline. Also most of the presentation formats support relative positioning of media elements in space. Only the profiles of SMIL aimed for mobile devices and Macromedia's Flash format use absolute positioning of media elements. In regard of the interaction model all formats support external and internal unidirectional links. This distinction becomes important for the transformation of the multimedia content into the final presentation format.

Table 1. Characteristics of different multimedia presentation formats

Presentation format	Temporal model	Spatial model	Navigational interaction
SMIL 2.0	local	relative	int., ext.
SMIL 2.0 BLP	local	absolute	int., ext.
3GPP SMIL 1.0.0	local	absolute	int., ext.
SVG 1.2	global	relative	int., ext.
SVG Basic	global	relative	int., ext.
SVG Tiny	global	relative	int., ext.
XMT-Ω	local	relative	int., ext.
Macromedia Flash	local	absolute	int., ext.
HTML+TIME	local	relative	int., ext.

This analysis of the existing presentation formats is targeted to develop a single, presentation format independent multimedia content model that is optimally suited to be automatically transformed to all the different presentation formats and to exploit the power of these formats at best. The design of this model is introduced in the next section.

3. An abstract multimedia content model

For the development of our abstract multimedia content model first the concrete characteristics of the abstract model must be determined to meet the modeling features of the different final presentation formats. We conclude from the analysis of the presentation formats that our abstract content model must provide a temporal model that allows to define local timelines and a spatial model, which supports relative positioning.

For the representation of multimedia content one can find the most different notations, e.g., petri nets in [13], flow chart like diagrams with Macromedia's Authorware [4], or tree like notations in [6]. As we will show in the following, the requirements for local timelines and relative positioning call for a tree model representation. In addition, a tree structure forms a good basis to be transformed into the syntax of the often XML-based presentation formats.

Our abstract multimedia model is introduced in the following in more detail along the central aspects of multimedia document models. We motivate our design decisions by concrete examples.

Temporal model: The temporal model of our abstract multimedia model has to be chosen such that we can generate local and global timelines to support the temporal model of the different presentation formats at the best. A temporal model that bases on local timelines can be directly mapped to the temporal model of a presentation format that also supports local timelines. If the final presentation format supports a global timeline, these local timelines can be automatically transformed into global positions in time. If we used a global timeline for our abstract model, we would not be able to sufficiently support the creation of semantically reasonable local timelines. Consequently, we decided for our abstract content model in favor of a temporal model with local timelines. For the representation of local timelines a tree structure, as can be found with SMIL and XMT-Ω, forms a reasonable representation. The multimedia document tree represents the time from its left to its right nodes, which can be converted to a global timeline.

To determine the temporal synchronization of the media elements, the abstract content model provides a set of temporal composition elements: The *Parallel* element is used to present its children, which are either media elements or other multimedia document trees, at the same time. It represents the functionality of the <par>-element in presentation formats like SMIL. Specific temporal multimedia composition aspects like `clipBegin` and `clipEnd`, which can be found in different presentation formats, are modeled by a *TemporalSelector*. This composition element selects a distinct time interval of an arbitrary media element or multimedia document tree. The composition element *Sequential* represents the functionality as can be found with the <seq>-element of SMIL. The children of this element are the same as for the Parallel element, but its semantics is to show them one after the other. To shift media elements on the time axis the composition element *Delay* can be used. The Delay element can be placed between any composition elements and defines a gap of arbitrary length within the presentation. For reasons of clarity the TemporalSelector and Delay element abandon the intermix of media element definitions and temporal synchronization as has been introduced with attributes such as `begin` into models like SMIL and SVG for reasons of convenience. An example of a temporal synchronization in the abstract model is depicted in Figure 4. Each temporal composition element determines its own local timeline. Since the temporal composition elements can be nested, multimedia presentations of arbitrary temporal layout can be created and by this, all temporal relations as defined by Allen [14] can be modeled.

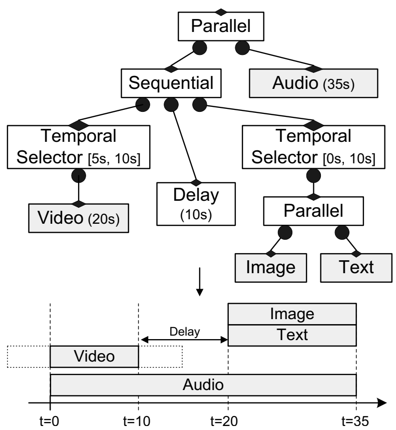

Fig. 4. Temporal synchronization.

Spatial model: For our abstract multimedia model a spatial model has to be chosen such that it can arrange visual media elements by relative and absolute positioning to provide support for the different presentation format's spatial models at the best. A spatial model that supports relative positioning can be directly mapped to a presentation format whose spatial model also supports relative positioning. Such a relative positioning model can also be transformed into absolute positioning of a global coordinate system, as supported by SVG for example. If we used a single global coordinate system for our abstract model, we can not sufficiently support the semantically reasonable creation of local coordinate systems with relative positioning of the media elements. As consequence, we decided for our abstract spatial model in favor of relative positioning.

This relative positioning is realized by so-called *SpatialProjectors*. Projectors realize the visual layout of the multimedia presentation following the region-concept of SMIL and embracing the hierarchically assignment of layout to media elements in SVG. They can be attached to media elements as well as composition elements. For each projector the figure shows its spatial extension in the presentation. As depicted in Figure 5 each projector defines its own (local) coordinate system within the visual media elements can be placed. For example projectors 3 and 4 place the media elements into the local coordinate system that is spawn by projector 2. Since composition elements can be nested, also the projectors are nested and influence each other. Consequently, with projectors multimedia presentations of arbitrary hierarchically organized spatial layout can be realized.

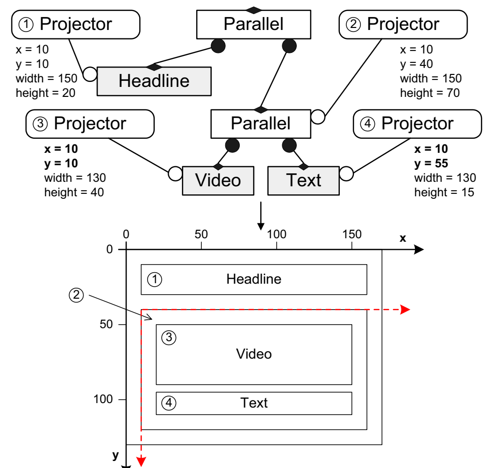

Fig. 5. Visual layout.

Interaction model: In regard of navigational interaction, the abstract content model supports the definition of both internal and external links: With internal links navigational interaction within a single multimedia presentation can be created like it is supported by SMIL and Flash for example. With external links, navigational interaction between different multimedia presentations is modeled as it is supported by the considered formats. The corresponding composition elements are *InternalLink* and *ExternalLink*. The distinction between the two is important for the later transformation of the multimedia content in the concrete presentation format: Internal links might be considered elegant and desirable in some cases, because the entire multimedia presentation is created within a single document. However, the target of internal links is embedded in the document at transformation time and can not be changed subsequently. With external links one can refer to external presentations and distinct elements within them. A presentation referred to by an external link could also be generated on-the-fly with reflecting the latest user profile information, just when the user clicks on the respective link.

Figure 6 shows an example where internal and external navigational interaction is used within our abstract content model. When the user clicks on the internal link associated with "Video 1", the presentation immediately jumps to the external link, which is associated with "Video 2". When the user clicks on "Video 2", the presentation continues with the external multimedia presentation. Providing both internal and external links gives the abstract content model the flexibility that is needed to satisfy different application requirements.

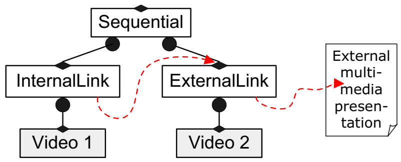

Fig. 6. Navigational interaction.

With the composition elements arbitrary multimedia content for any application area can be created and represented in the abstract tree model. However, the abstract content model is not limited to the presented elements. It also comprises other composition elements for defining, e.g., menus and hotspots, and other types of projectors, such as an *AcousticProjector* and *TypographicProjector*. In addition, the abstract model can be extended by further multimedia composition functionality, such as support for input elements, transitions, or other visual effects like blurring and sharpening of media elements.

Since our tree model has been designed to support local timelines, relative positioning, and interactive linking it serves the individual modeling characteristics of today's presentation formats. However, it also provides support for other aspects of multimedia modeling, such as encapsulation and reuse of composition elements, which are not relevant in the context of this paper.

4. Transformation to final presentation

The multimedia content represented in our abstract model is transformed in the next step by our application-independent transformation process to the features and syntax of the different presentation formats. At this point, it is important to note that from one abstract multimedia document not necessarily all of the different presentation formats are reasonable for the transformation. For example, if the multimedia content is designed for a desktop PC with a minimum screen resolution of 1024 $\times$ 768, reasonable target formats are SMIL 2.0, SVG 1.2, and Flash. To present the same multimedia content on a mobile device by using Mobile SVG or SMIL 2.0 BLP, typically media assets of smaller resolution have to be selected and the spatial layout of the presentation is to be changed. This is supported by our multimedia personalization framework with its ability for a personalized and with it end device dependent selection of media elements and its extensibility in regard of integrating arbitrary (possibly application-specific) multimedia layout functionality.

The transformation process from the abstract content model to the concrete presentation formats depends on the specific characteristics of the target presentation format: its temporal, spatial, and interaction model as well as its specific syntactic elements. For each of the target options we discuss in the following how they are treated in our transformation process.

4.1. From local timelines to global timeline

If the target presentation format's temporal model supports local timelines, then the abstract document tree already holds the time model of the targeted format. If the presentation format supports a global timeline, the local timelines are transformed by the algorithm presented in Listing 1. The algorithm starts at the root of the document tree and the zero point in time with the initial call `totalDuration<-CalculateGlobalTimeline(rootnode,0)`. It traverses the nodes of the document tree from left to right in depth first order. On traversing the algorithm accumulates the global time of the presentation from the local timelines. The duration value which is added to the global presentation time depends on the node type: For the node types Medium, TemporalSelector, and Delay the type's intrinsic duration is added to the global time. Media elements have an intrinsic duration that can be shorten by the usage of a TemporalSelector. By the composition element Delay, all the subsequent nodes of the document tree are shifted on the timeline by a defined duration (lines 8 to 10). The composition element Sequential itself does not add to the global time. The time progresses by the traversal of the Sequential element's children (lines 11 to 18). The duration of the Parallel element is determined by duration of one of its children. This can be the child element with the maximum presentation time (lines 19 to 27), the element with the minimum presentation time, or a particular element i (not shown in Listing 1). On traversal of the leaf nodes for each media element its position on the global timeline is captured (not shown in the listing).

```
1  function CalculateGlobalTimeline( node, time ) returns time
   inputs:
     node, the current node in the multimedia document tree
     time, the current value of the global timeline
5
   type <- GetNodeType( node )
   case type of
     Medium, TemporalSelector, Delay:
       time = time + GetDuration( node )
10     return time
     Sequential:
       subnodes <- Subnodes( node )
       loop do
         subnode <- Remove-Front( subnodes )
15       if subnodes is empty then break
         time <- CalculateGlobalTimeline( subnode, time )
       end loop
       return time
     Parallel:
20     startTime = time
       subnodes <- Subnodes( node )
       loop do
         subnode <- Remove-Front( subnodes )
         if subnodes is empty then break
25       endTime <- CalculateGlobalTimeline( subnode, startTime )
         time <- Maximum( time, endTime )
       end loop
       return time
   end case
```

Listing 1. Algorithm to calculate the global timeline from the local timelines.

4.2. From relative to absolute positioning

For presentation formats that allow the relative positioning of media elements, as it is supported by SVG and XMT-Ω for example, the abstract content model represents already the spatial model of the target format. Here, the projectors of the abstract content model are directly converted to the syntax of the presentation format. If a target format provides a spatial model with absolute positioning, the projectors are recalculated to absolute positions in space by applying the algorithm in Listing 2. The algorithm starts with the root element of the document tree. The initial reference point to calculate the absolute positions of the visual media elements constitutes the origin of the global coordinate system, i.e., the coordinate system of the entire multimedia presentation. The corresponding initial call of the algorithm is `CalculateAbsolutePositions(rootnode,0,0)`. As the transformation from local timelines to a global timeline, the algorithm traverses the document tree from left to right and depth first. For each visual element the projectors determine the position in space. Each time when a SpatialProjector is attached to a node (lines 7 and 8), the projector's x and y value are added to the accumulated current absolute position (lines 9 and 10). The new absolute position (lines 11 and 12) are the reference values for the further traversal of the nodes children (lines 15 to 19). For each visual media element its absolute position in space is stored during traversal. So, for each visual media element the scope is mapped from its prior local coordinate system to the global coordinate system of the entire multimedia presentation.

```
1  function CalculateAbsolutePositions( node, x, y )
   inputs:
     node, the current node in the multimedia document tree
     x, position of the current node on the x-axis
5    y, position of the current node on the y-axis

   if node has SpatialProjector then
     spatial-p <- GetSpatialProjector( node )
     x <- x + GetXPosition( spatial-p )
10   y <- y + GetYPosition( spatial-p )
     SetXPosition( spatial-p, x )
     SetYPosition( spatial-p, y )
   end if
   subnodes <- Subnodes( node )
15 loop do
     if subnodes is empty then break
     subnode <- Remove-Front( subnodes )
     CalculateAbsolutePositions( subnode, x, y )
   end loop
```

Listing 2. Algorithm to calculate absolute positions from local coordinate systems.

4.3. Transforming the interaction model

As all presentation formats support external and internal links, ideally the transformation process can translate the link nodes of the document tree into the syntax of the presentation format. Whereas for the presentation formats listed in Table 1 the transformation of the temporal and spatial model results in a target presentation, that can be rendered by the respective multimedia players. However, this does not hold for internal navigational links: Even though all presentation formats define them, some of the multimedia players do not support these internal links, e.g., the Internet Explorer in conjunction with Adobe's SVG plug-in [15]. In addition, referring to a distinct part of an external presentation is not supported by all players, e.g., the RealPlayer presenting SMIL [16]. Therefore, for practical reasons we currently do not have any other choice than to develop transformers that are not only format specific, but also player specific: If internal links are not supported, the document tree is split up into separate document trees for the target of each InternalLink element. The internal links are thereby converted to respective external links referring to newly created external multimedia documents. If the player does not support linking to a particular fragment of an external presentation, there are two options: Either the external link is changed to refer just to the entire external presentation or the fragment of the external presentation needs to be extracted and presented as a single external document. Pleased, we observe the ongoing development of multimedia players that really support internal and external navigational interaction, e.g., the Ambulant/G player for rendering SMIL 2.0 documents [17].

4.4. Mapping of abstract model to the syntax of the presentation format

Besides the transformation of the temporal, spatial, and interaction model, the abstract content model is mapped to the concrete syntax of the targeted format. Our abstract content model consists of a small set of domain independent and therefore universally applicable composition elements. For each of these there exist adequate syntactic elements in the targeted presentation formats such as SMIL and SVG. Figure 7 gives you an example, how the syntactic transformation is actually realized. It depicts a simple slideshow presentation represented in our abstract content model. This slideshow example is transformed to the presentation formats SMIL 2.0 and SVG 1.2, targeted at desktop PCs, and SMIL 2.0 Basic Language Profile (SMIL 2.0 BLP), aimed for mobile devices. For reason of clearness, we suppose that the transformation of the abstract slideshow to these presentation formats results in meaningful presentations for the targeted end devices (cp. the discussion at the beginning of this section). This can be realized, e.g., by selecting appropriate media elements for the different display resolutions of the end devices. The selected target formats cover the different presentation characteristics as identified by our analysis and listed in Table 1: With SMIL 2.0, the abstract document tree reflects already the temporal and spatial model of this presentation format. In contrary SMIL 2.0 BLP does not support relative positioning. Finally SVG 1.2 does not support local timelines. In Figure 7, the encircled numbers at the edges of the graph indicate the order in which the nodes are transformed to the target format. The transformation results for SMIL 2.0, SMIL 2.0 BLP, and SVG 1.2 are depicted in Listing 3 to 5. The numbers at the beginning of each line indicate which lines are created by which step of the transformation process and allow the reader to keep track of the transformation process.

The transformation process starts at the root node of the multimedia document tree. It traverses the document tree from left to right in depth first order. Each time the algorithm reaches a media element, this media element and its associated projectors are transformed into the corresponding syntactic elements of the target format. The first time this is the case is for the "Title" Image. It is syntactically represented in SMIL by using the `<img>`-tag and in SVG by `<image>` (1). At the same time, the syntactic elements that determine the spatial layout of the presentation are created.[1] Therefor, the SpatialProjectors that are attached to the media elements are transformed to respective `<region>`-elements in SMIL and `<g>`-tags in SVG. Then the algorithm continues with traversing the abstract document tree until it reaches the media elements "Slide 1" and "Text 1". The Image medium "Slide 1" is syntactically represented as the "Title" Image above (2). For the Text medium "Text 1" a particular instance of the `<text>`-element is used (3). Then, the composition element Parallel itself is transformed (4). In the case of SMIL the element `<par>` is used. Since SVG does not provide an appropriate syntactic element for Parallel, the media elements "Slide 1" and "Text 1" are just placed at the same point in time on the global timeline to simulate its functionality. These global points in time are calculated by the algorithm presented in Section 4.2. The transformation process continues with the second child of the Sequential element and transforms it analogously to the first one (5 to 7). Then the composition element Sequential is mapped to the target format: Here, the `<seq>`-tag is used in the case of SMIL. For SVG the media elements of the two slides are placed on the global timeline in a row by setting the values of the `begin` attributes to the corresponding global points in time. As the media elements, also the Sequential element has a SpatialProjector attached. It affects all the projectors of the media elements in the sub-tree. To realize such

[1]In fact, these two steps are performed sequentially. However, we consider them here as being processed concurrently for reasons of simplicity.

a nested spatial layout in the target formats, SMIL 2.0 and SVG 1.2 allow to nest the respective `<region>` and `<g>` elements. Therefore, with the SpatialProjector attached to the Sequential element a sub-region is created that shifts the scope of the coordinate system to a particular point in space to which the visual media elements of the sub-tree then refer to (8). This is syntactically realized in the target formats by embracing the `<region>` respectively `<g>` tags of the media elements with a `<region>` respectively `<g>` tag of the projector attached to the Sequential element. In the case of SMIL 2.0 BLP where only absolute positioning of media elements is supported, the `<region>`-tags can not be nested. Therefore, the absolute positionings calculated by the algorithm presented in Section 4.2 are used for the region's attributes `left` and `top`. After the Sequential element is successfully transformed, the transformation process continues with mapping the root element Parallel to the target format (9). Finally, the transformation process finishes with creating the syntactic frame of the entire presentation (10).

This small example illustrates already how the transformation of the abstract document tree to the concrete syntax of the presentation formats is realized. Since the transformation process described above is domain independent, only one transformer needs to be developed for each presentation format to support any personalized multimedia application that is developed by using our software framework. Please note, that the composition of the multimedia content is separated from the transformation process: The adaption of the multimedia content to the capabilities of the end devices is carried out in the composition. With the transformation process we are mapping the multimedia content to a presentation format that can be rendered by a multimedia player on the end user's device.

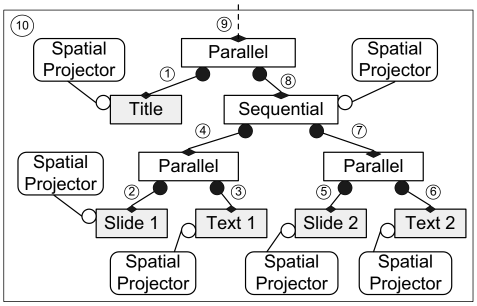

Fig. 7. Slideshow in abstract model that is transformed to different presentation formats.

```
(10)<smil xmlns="http://www.w3.org/2001/SMIL20/Language">
(10)<head>
(10) <layout type="text/smil-basic-layout">
 (9) <root-layout width="720" height="597"/>
 (1) <region id="ti" left="0" top="0" width="125" height="37"/>
 (8) <region id="sr" left="0" top="37" width="720" height="560">
 (2)  <region id="s" left="0" top="0" width="720" height="540"/>
 (3)  <region id="t" left="0" top="540" width="720" height="20"/>
 (8) </region>
(10) </layout>
(10)</head>
(10)<body>
 (9) <par>
 (1)  <img region="ti" src="title.gif" begin="0s" dur="20s"/>
 (8)  <seq>
 (4)   <par>
 (2)    <img region="s" src="slide1.jpg" begin="0s" dur="10s"/>
 (3)    <text region="t" src="text1.txt" begin="0s" dur="10s"/>
 (4)   </par>
 (7)   <par>
 (5)    <img region="s" src="slide2.jpg" begin="0s" dur="10s"/>
 (6)    <text region="t" src="text2.txt" begin="0s" dur="10s"/>
 (7)   </par>
 (8)  </seq>
 (9) </par>
(10)</body>
(10)</smil>
```

Listing 3. The slideshow transformed to SMIL 2.0.

```
(10)<smil xmlns="http://www.w3.org/2001/SMIL20/Language">
(10)<head>
(10) <layout type="text/smil-basic-layout">
 (9) <root-layout width="720" height="597"/>
 (1) <region id="ti" left="0" top="0" width="125" height="37"/>
 (2) <region id="s" left="0" top="37" width="720" height="540"/>
 (3) <region id="t" left="0" top="577" width="720" height="20"/>
(10) </layout>
(10)</head>
(10)<body> [...] </body>
(10)</smil>
```

Listing 4. The slideshow in SMIL 2.0 BLP.

```
(10)<svg xmlns="http://www.w3.org/2000/svg" width="720"
      height="597" version="1.2">
 (1)<image x="0" y="0" width="125" height="37"
      xlink:href="title.gif" style="visibility:hidden;">
 (1) <set attributeName="visibility" attributeType="CSS"
       to="visible" begin="0s" dur="20s" fill="remove"/>
 (1)</image>
 (8)<g id="sr" transform="translate(0 37)">
 (2) <g transform="translate(0 0)">
 (2)  <image x="0" y="0" width="720" height="540"
         xlink:href="slide1.jpg" style="visibility:hidden;">
 (2)   <set attributeName="visibility" attributeType="CSS"
          to="visible" begin="0s" dur="10s" fill="remove"/>
 (2)  </image></g>
 (3) <g transform="translate(0 540)">
 (3)  <text x="0" y="10" style="visibility:hidden;fontsize:24;">
 (3)   <set attributeName="visibility" attributeType="CSS"
        to="visible" begin="0s" dur="10s" fill="remove"/>
 (3)   This is the description of slide1.</text></g>
 (5) <g transform="translate(0 0)">
 (5)  <image x="0" y="0" width="720" height="540"
         xlink:href="slide2.jpg" style="visibility:hidden;">
 (5)   <set attributeName="visibility" attributeType="CSS"
          to="visible" begin="10s" dur="10s" fill="remove"/>
 (5)  </image></g>
 (6) <g transform="translate(0 540)">
 (6)  <text x="0" y="10" style="visibility:hidden;fontsize:24;">
 (6)   <set attributeName="visibility" attributeType="CSS"
          to="visible" begin="10s" dur="10s" fill="remove"/>
 (6)   This is slide two.</text></g>
 (8)</g>
(10)</svg>
```

Listing 5. The slideshow transformed to SVG 1.2.

5. Implementation issues and applications

The abstract content model and the transformation process constitute the core of our multimedia personalization framework [2]. Like the abstract content model and the transformation process, the entire framework is independent of any concrete application area and supports the dynamic creation of arbitrary personalized multimedia content. For the implementation of the transformation process the design pattern abstract factory is used [18]. With this pattern an abstract transformer class is created, which implements the generic algorithms for converting the temporal course, visual layout, and interaction possibilities. From this abstract transformer concrete instances are derived, one for each presentation format. We implemented already concrete transformers for SMIL 2.0, SMIL 2.0 BLP, SVG 1.2, SVG Tiny, SVG Basic, MPEG-4's XMT-Ω, Flash, but also for HTML and XHTML. Currently we are working on transformers for 3GPP SMIL and HTML+TIME.

To show the applicability of our framework approach, we developed demonstrator applications from different domains: These applications are a personalized tourist guide application, a multimedia sports news ticker, a service for interactive instruction manuals on mobile devices, and some other examples of

1550-5502/05 $20.00 © 2005 IEEE

multimedia presentations. For the tourist guide application, we already set up a test site [19] for experimenting with all the different presentation options. The presentations in Figure 8 show the result of the adaptive multimedia composition and its transformation to different target formats for different end devices. It presents a map of a city with some important sights on it. Which sights are actually presented depends on the user's interests and preferences. When the user clicks on these sights he or she receives a multimedia presentation with further information about it. All multimedia presentations of this and the other demonstrator applications are created on-demand by our multimedia presentation framework employing the abstract multimedia content model and transformation process presented in this paper.

Fig. 8. Screenshots of our tourist application.

6. Conclusion

In our work, we have been approaching the problem of delivering multimedia content to a wide range of different devices. As this can be achieved economically only by a multi-channel generation of multimedia presentations, we analyzed the different presentation formats that are supported by today's end user's agents. The result is the design of our abstract multimedia content model that imbeds the central characteristics of today's presentation formats. It is designed to be efficiently transformed by the application independent transformers to the concrete syntax of the different presentation formats. Different prototypes show the applicability of the multi-channel transformation process in the context of our multimedia personalization framework. In a next step we work on extending the abstract model for carrying the meta data of the media elements and the composition operators such that the meta data information will be mapped by the transformation process to the final presentation format's (meta data) syntax, e.g., Dublin Core [20] embedded in (X)HTML, RDF [3] with SMIL's Metainformation module, or the annotation format Annodex [21] for time-continuous media streams and its markup-language CMML [22], respectively.

As we can expect that future multimedia presentation standards will also have to provide means for the central multimedia modeling aspects captured by our model, we are also prepared to support new versions of the presentation formats or even brand-new formats. The presented approach for multi-channel multimedia generation is not only embedded into our software framework for developing personalized multimedia applications but can also be seen as a service provider for systems and applications that create multimedia content and need to deliver it over the last mile to the individual user's end device setting.

7. References

[1] S. Boll, W. Klas, and U. Westermann, "Multimedia document formats - sealed fate or setting out for new shores?," *Proc. of the IEEE Int. Conf. on Multimedia Computing and Systems*, June 1999.

[2] A. Scherp and S. Boll, *Managing Multimedia Semantics*, chapter MM4U - A framework for creating personalized multimedia content, Idea Group Publishing, 2005.

[3] W3C, "World Wide Web Consortium," 1994–2004, All specificiations available at http://www.w3.org.

[4] Macromedia, Inc., USA, "Macromedia," 2004, URL: http://www.macromedia.com.

[5] M. Jourdan, N. Layada, C. Roisin, L. Sabry-Ismal, and L. Tardif, "Madeus, and authoring environment for interactive multimedia documents," *ACM Multimedia*, 1998.

[6] Dick C. A. Bulterman, "Embedded video in hypermedia documents: supporting integration and adaptive control," *ACM Trans. on Info. Systems*, vol. 13, no. 4, 1995.

[7] L. Hardman, D.C.A. Bulterman, and G. van Rossum, "The Amsterdam hypermedia model: adding time and context to the Dexter model," *Commun. ACM*, vol. 37, no. 2, 1994.

[8] J.R. van Ossenbruggen, F.J. Cornelissen, J.P.T.M. Geurts, L.W. Rutledge, and H.L. Hardman, "Cuypers: a semi-automatic hypermedia presentation system," Tech. Rep. INS-R0025, CWI, Netherlands, 2000.

[9] INRIA, France, "Web, Adaptation and Multimedia," 2003, URL: http://www.inria.fr/recherche/equipes/wam.en.html.

[10] M. Kim and S. Wood, "XMT: MPEG-4 Textual Format for Cross-Standard Interoperability," 2004, URL: http://www.research.ibm.com/mpeg4/Projects/XMTInterop.htm.

[11] International Organization for Standardization, "ISO/IEC 14496-1:2001 Information technology – Coding of audio-visual objects – Part 1: Systems; Amendment 3 'XMT'," 2001.

[12] S. Boll and W. Klas, "ZYX - A Multimedia Document Model for Reuse and Adaptation," *IEEE Trans. on Knowledge and Data Engineering*, vol. 13, no. 3, 2001.

[13] T. D. C. Little and A. Ghafoor, "Synchronization and storage models for multimedia objects," *IEEE J. Sel. Areas Commun.*, vol. 8, no. 3, Apr. 1990.

[14] J.F. Allen, "Maintaining knowledge about temporal intervals," *Commun. ACM*, vol. 26, no. 11, 1983.

[15] Adobe Systems, Inc., USA, "Adobe SVG Viewer," 2004, URL: http://www.adobe.com/svg.

[16] RealNetworks, USA, "RealPlayer," 2004, URL: http://www.real.com.

[17] CWI, Netherlands, "The AMBULANT Project," 2004, URL: http://www.cwi.nl/projects/Ambulant.

[18] E. Gamma, R. Helm, R. Johnson, and J. Vlissides, *Design patterns: elements of reusable object-oriented software*, Addison-Wesley, 24th edition, 2002.

[19] "Sightseeing4U Demonstrator," 2004, URL: http://www.offis.de/mm4u/ (Note: The city map image is used with permission from CCV GmbH, Germany).

[20] Dublin Core Metadata Initiative, "Expressing Dublin Core in HTML/XHTML meta and link elements," 1995-2004, URL: http://dublincore.org/documents/dcq-html/.

[21] CSIRO, Australia, "Annodex.net," 2004, URL: http://www.annodex.net.

[22] S. Pfeiffer, C. Parker, and A. Pang, "The Continuous Media Markup Language, Version 2.0," 2003, URL: http://www.annodex.net/TR/draft-pfeiffer-cmml-01.html.