Proceedings of the IASTED International Conference
Internet & Multimedia Systems & Applications and Visual Communications
March 17-19, 2008 Innsbruck, Austria
ISBN Hardcopy: 978-0-88986-727-7 / CD: 978-0-88986-728-4

SEMANTICS SUPPORT FOR PERSONALIZED MULTIMEDIA CONTENT

Ansgar Scherp
OFFIS Institute for Information Technology, Oldenburg, Germany
ansgar.scherp@offis.de

ABSTRACT
Support for personalized multimedia content has become a crucial aspect of today's multimedia applications. We find many different systems and approaches that provide multimedia content tailored to the specific needs and requirements of the users. These systems and approaches exploit semantically-rich information for the multimedia content creation task. However, once the content is created, this very valuable source of information is thrown away. Thus, it is lost for any further processing of the created multimedia presentations. Systems that analyze such presentations can only revive a very limited amount of the semantic information that was initially available and used. Consequently, we present with the SemanticMM4U framework an approach that enhances the state-of-the-art by providing support for deriving semantics at any point in time during the assemble process and integrating it into the syntax of the final presentation formats. By this, the semantics is made machine-readable and machine-processable. It allows for a better indexing, retrieval, and processing of the multimedia presentations.

KEY WORDS
Multimedia Tools and Architectures, Personalized Multimedia Content Authoring, Multimedia Semantics Derivation, Multimedia Albums, Component Framework

1 Introduction

Today's multimedia applications not only need to provide for the assembly of media assets in time, space, and interaction into a coherent presentation. They also need to create and provide this content in a personalized manner, tailored to the specific needs and requirements of their users. This includes the users' knowledge, background information, interests, and preferences as well as current location, used end device, and social context in which the users are. Consequently, we find today many different systems and approaches that provide their users with specifically to their needs and requirements tailored multimedia content. These systems and approaches exploit semantically-rich information for their multimedia content creation and personalization tasks. However, semantically-rich information is also derived during the multimedia content creation process itself. For example, the didactic structure of an e-learning unit or temporal course of a multimedia presentation about the highlights of the last summer vacation. This information is derived based on the most different sources including the existing meta data of the media assets, the users' preferences, needs, and requirements, contextual information, as well as any other source of semantically-rich information.

However, the problem of today's systems and approaches is that none of them disseminates the existing or derived semantics with the actually created multimedia content in the final presentation formats such as SMIL [33], SVG [31], and Flash [1]. Once the personalized content is created, the existing as well as derived semantics is thrown away and not further used. This loss of information is very unfortunate, especially when it comes to a persistent delivery of the created content such as a presentation of the last vacation stored with your personal homepage on the Internet or a system that creates e-learning units and stores them back in a database for later presentation and reuse. We refer to this loss of information as the *semantics barrier* in multimedia content creation [24].

To overcome the semantics barrier, we developed the SemanticMM4U framework. The SemanticMM4U framework bases on our previous work [20, 23, 22], where we focused on the multimedia composition and personalization functionalities. In this paper, we describe the framework's support for bringing the semantic information from the media storages through the assemble process and semantics derivation to the presentation of the content. We present how semantics can be derived at any point in time during the assemble process. We show, how the semantics is internally represented in an abstract multimedia content model and how this representation of semantics is transformed and integrated into the syntax of the final presentation formats such as SMIL, SVG, and Flash. We also consider the mobile profiles of these formats like SMIL Extended Mobile Profile [33] as well as SVG Basic and SVG Tiny [30].

2 Related Work

Research in regard of authoring personalized multimedia content has been conducted for more than a decade. The multimedia community has reached very diverse and sophisticated knowledge in creating such content. Prove of these achievements are systems and approaches like the online bookstore Amazon [2] for text-centric content, the industry system HotStreams [14], and mobile tourist guides like LoL@ [27] and GUIDE [6]. In regard of multimedia content, we find the well-known Cuypers Multimedia Transformation Engine [12, 28], the projects Opé ra [3] and its successor WAM [15, 16], the Semi-automatic Multimedia Presentation Generation Environment [10, 11], and the

Standard Reference Model for Intelligent Multimedia Presentation Systems [26, 9]. In the following, we analyze two of the most relevant systems in detail to get a better understanding of the creation process of semantically-rich personalized multimedia presentations. These systems are the Semi-automatic Multimedia Presentation Generation Environment and the Cuypers Multimedia Transformation Engine.

With the Semi-Automatic Multimedia Presentation Generation Environment (Samp*L*e), first a narrative structure of the presentation is build in an iterative process with the user. This includes identifying the theme of the presentation in terms of content and presentation settings. Content settings comprise specifying the topic and possible related topics, whereas presentation settings deal with the media types used for the presentation such as audio, video, and text. Presentation settings also consider the temporal flow of the presentation and the interaction possibilities of user with the presentation. Subsequently, the structure of the presentation is determined by selecting and altering genre templates. A genre describes a presentation structure for a specific domain and consists of different conceptual parts such as introduction, description, and conclusion. These structures and conceptual parts can be edited by the user. They can be, e. g., extended, reduced, and rearranged. Subsequently, media assets are retrieved that are appropriate for the current conceptual part the user is working on. The media assets are arranged based on rules concerning the content, discourse structure, and presentation structure. Finally, when the theme, presentation structure, and general arrangement of the material is specified, the authored content is made available for consumption to the users in HTML format. However, when generating the HTML pages, the rhetorical structure of the presentation build in the iterative creation process and any other semantically-rich information is thrown away and lost.

The Cuypers Multimedia Transformation Engine is a testbed for developing and experimenting with the dynamic generation of personalized multimedia presentations in the SMIL format. These SMIL presentations adhere to the limitations of the target platform as well as the user preferences. Little et al. [17] present an implementation of a demonstrator application generating such personalized multimedia presentations through semantic inferencing. It bases on the Open Archives Initiative (OAI) [18]. The OAI defines an interoperability framework that allows for sharing and exchanging meta data. The meta data is provided in Dublin Core [8]. However, also any other schema are allowed and can be used, distinguished through unique namespaces [17]. Thus, the OAI can be considered as providing a large ontology on which one can iteratively search and conduct semantic inferences. The demonstrator application allows its users to enter keywords for searching the archives. Once the initial result set is determined, the users can select media assets and by this manipulate the set. When the users tell the demonstrator to generate a multimedia presentation, semantic inferencing rules are applied on the result set, i. e., the selected media assets. Finally, a corresponding multimedia presentation is generated and rendered on the user's end device. Within this multimedia presentation, the user can again select a media asset of interest. The selected media asset is then used as input to search again for other media assets based on keywords. By this, the user redirects the focus of the presented multimedia content interactively. This information is used to further refine the search result and derive new semantic relationships within the archive. Thus, new semantics about the media assets used in the presentations is derived. When the user is satisfied with the search results, he or she starts generating the multimedia presentation. Here, the selected media assets are arranged automatically into time and space of the presentation. For it, *mapping rules* are applied that allow for translating the semantic relationships into spatio-temporal relationships of SMIL. For example, the picture of San Francisco would be arranged side by side with the diary text describing this picture. However, the semantics derived in form of a rhetorical structure and the created SMIL presentation are separated in two different models. Once, the rhetorical structure is exploited to create the SMIL content, it is not further used and thrown away. Consequently, the semantically-rich information derived during the authoring process is lost.

3 Creation Process of Semantically-rich Personalized Multimedia Presentations

Based on the systems analyzed in Section 2 and an extensive analysis of further systems and approaches [24, 20] conducted earlier, we determined a general process chain for creating semantically-rich personalized multimedia content. This process chain is depicted in Figure 1. It starts with the selection of media assets according to the users' needs, preferences, and contextual situation. In the subsequent assemble process, these media assets are arranged in time and space using a multimedia content representation model—again taking the users' needs into account. The representation model also allows for defining navigational interaction in form of hyperlinks. It abstracts from the syntax and features of today's multimedia presentation formats such as SMIL, SVG, and Flash. In the assemble process, the multimedia content in the representation model can be enhanced by further meta data derived in the concurrent annotate process. This meta data is derived based on the created content, the used media assets and their meta data, the users' needs, preferences, and contextual situation, and any other semantically rich information. Only then, in the transform process, the multimedia content in the representation model is mapped to the syntax and features of the concrete presentation formats. The transformation step may be omitted, if the multimedia content is internally already composed in a final presentation format. Finally, the created multimedia presentations are delivered and rendered on the users' end devices in the present process.

Considering today's approaches and systems for generating (semantically-rich) personalized multimedia presentations, they support all or some of these processes in

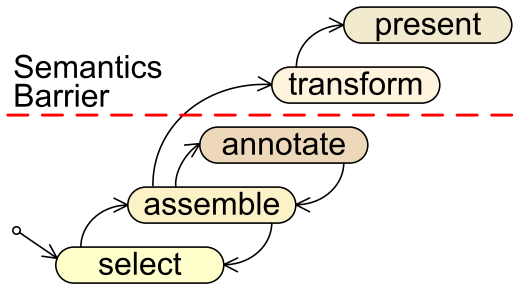

Figure 1. Creation process of semantically-rich personalized multimedia presentations

order to create and deliver personalized content to their users. They use meta data provided with the media assets to select the most appropriate assets in regard of the users' needs, preferences and contextual situation. They exploit further semantic information to assemble the media assets into a coherent multimedia presentation. For creating the presentation, they not only exploit this semantic information but also derive new semantic information. As the detailed analysis of the Cuypers Multimedia Presentation Engine and the Samp*L*e framework in Section 2 shows, this semantics can be derived in two directions: Firstly, it can be derived for the created multimedia presentation. Secondly, semantics can be derived about the media assets used for the presentation. The latter can be back-stored into the media storages and re-used for subsequent multimedia composition tasks. For example, when a set of images is arranged on the same page of a page-based multimedia presentation, it allows for deriving that the images belong together and form a semantic concept. In the other direction, placing a text close to an image can be interpreted as a textual description of the image in form of a caption, which is then back-stored and used for later composition tasks.

Thus, the lack of today's systems and approaches is that the very valuable source of semantically-rich information that existed or is derived during the assemble process of the multimedia content is thrown away when it comes to the content transformation and delivery, respectively. This loss of semantically-rich information is referred to as the *semantics barrier* and is indicated in Figure 1 as a horizontal bar above the annotate process. The created multimedia presentations carry none or only a very small piece of the semantically-rich information that was originally used and derived during the creation process. This is very unfortunate in regard of possible further processing of the multimedia presentations, e. g., when they are made available on the Internet for the general public or stored in a database for later (re-)use.

To relieve this problem, one approach could be to analyze the content post to the creation process. For example, Ding et al. [7] presented an approach to analyze binary Flash presentations. They analyzed specific characteristics of the presentations in order to "rescue" some of the semantic information. However, for this or any other similar approach, none or only very few of the original semantic information is available. In addition, the analysis process is very tedious, time-consuming, and error prone. Consequently, the semantic information that can be revived from the content is very little compared to the rich semantic information that potentially has been available during the assembly process.

To overcome the semantics barrier, an approach is needed that not only allows for authoring personalized multimedia presentations but also provides for deriving semantics during any step of the assemble process, representing it in a flexible fashion, and integrating the semantics with the final presentation formats' syntax and features. With the SemanticMM4U framework, we present such an approach that allows for representing, deriving, and delivering semantics for personalized multimedia presentations.

4 Authoring Support for Semantically-rich Personalized Multimedia Presentations

In the previous sections, we introduced the general creation process for semantically-rich personalized multimedia content and described the semantics barrier today's systems and approaches have. We motivated the need for a new approach to overcome this semantics barrier. Such an approach is presented with the SemanticMM4U framework. Requirements to the framework are authoring semantically-rich personalized multimedia content for different (mobile) end devices and in different presentation formats including SMIL, SVG, and Flash, providing support for application-specific personalization and composition functionalities, and allowing for (application-specific) semantics derivation [20]. The multimedia composition and personalization features of the framework are described in detail in [20, 23, 22]. In this paper, we focus on the framework's support for deriving, representing, and delivering the semantics of multimedia presentations. For it, we briefly introduce in Section 4.1 the component-based architecture of the SemanticMM4U framework. The framework's support for representing, deriving, and delivering multimedia semantics is presented in detail in Sections 4.2 to 4.4, before we conclude the section.

4.1 SemanticMM4U Framework Architecture

The component architecture of the SemanticMM4U framework is depicted as UML component diagram in Figure 1. It is defined based on the general creation process for semantically-rich personalized multimedia presentations described in Section 3. The UML diagram shows six components, their central interfaces, and the relationships between the components. The components are described from bottom to top: The selection process of the general creation chain is realized by the User Profile Accessor and Media Pool Accessor components at the bottom. The User Profile Accessor component provides access to user profile information and context information using the interface IUserProfile. The Media Pool Accessor component allows for retrieving media assets from different media storages with the interfaces IMedium and IMediaList. It also supports back-storing the media assets' meta data derived during the assemble process into the media storages. The media assets are selected by the Media Pool Accessor component based on the information provided from the User Profile Accessor component. It is triggered by the Multimedia Composition component, which implements the assemble process of the general creation chain. The Multimedia Composition component can add arbitrary application-specific parameters to the query. The selected media assets are organized by the Multimedia Composition component in time, space, and navigational interaction into the personalized multimedia content. For the personalized multimedia content, an abstract tree-based representation model is used [22]. This representation model allows for an as easy as possible transformation into the syntax of the different presentation formats like SMIL, SVG, and Flash. It also allows for storing sophisticated meta data about the created multimedia content. The root node as well as any other node of the tree-based representation model is of type IVariable. Thus, the entire presentation or any part of it can be passed to the Metadata Derivation component using the interface IVariable. The Metadata Derivation component provides the annotate process. Instances of the Metadata Derivation component allow for deriving semantically-rich information about the created multimedia content and store this information with the nodes of the abstract representation tree. Once the semantically-rich personalized multimedia content is created in the representation model, it is passed on to the Presentation Format Generators component. This component implements the transform process. Besides the mapping of the actual multimedia content to the syntax and features of the presentation formats [23], also the presentation's meta data is transformed into the target formats. By this, the existing and derived meta data during the assemble process is made explicitly available with the created presentations. Finally, the semantically-rich multimedia presentations in their final formats are passed on to the Multimedia Presentation component, using the interface IMultimediaPresentation. This component actually renders the presentations on the end devices of the users, thus providing the present process.

Having introduced the component architecture of the SemanticMM4U framework, we present in the following the framework's support for multimedia semantics. In Section 4.2, we look in detail into the multimedia semantics representation provided by the Multimedia Composition component. Section 4.3 presents the derivation of new semantics by the framework's Semantics Derivation component. Finally, the mapping of the multimedia semantics in the internal representation model to the syntax and features of the output formats—provided by the Presentation Format Generators component—is presented in Section 4.4, before we summarize this section.

4.2 Representing Multimedia Semantics

The multimedia content representation model of the Multimedia Composition component is capable of organizing and representing media assets in time, space, and navigational interaction in a tree-like data structure. Nodes of this tree are basic multimedia composition units such as media assets (audio, video, images, and text) and composition operators like Parallel and Sequential defining the presentation's temporal course. In addition, projectors can be attached to the tree's nodes arranging the media assets in space and defining the acoustic layout [22]. For representing the multimedia content's semantics, the SemanticMM4U framework allows for annotating any structural node of the representation tree, i. e., the media assets and composition operators. These annotations provide for a semantically-rich description of the created multimedia content. An example of multimedia content and its annotations in the representation tree is depicted in Figure 3. It shows a simple slideshow of images that are played together with a background music. As shown in the figure, each media asset and composition operator can have arbitrary meta data. This meta data not only describes the representation node it is attached to, but can also refer to other notes of the content tree (not shown in the diagram). In other words, there can be references to subnodes of the composition node or to any other media asset and composition operator of the created multimedia content. For example, a meta data entry of the Parallel operator could indicate that the playback of the music is semantically related with the order in which the images are displayed. Meta data entries cannot be associated with the projectors of the representation tree as these are already considered as (very basic) meta data providing information such as the width and height of an image or the volume of an audio.

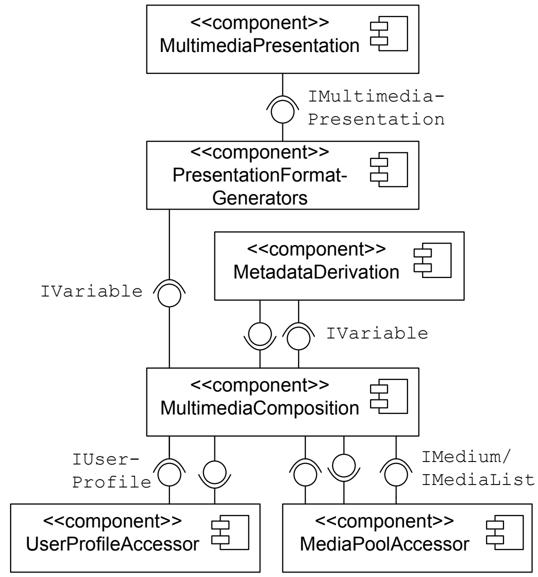

Figure 2. UML component diagram of the SemanticMM4U framework

The meta data annotations of the composition nodes are organized in a key-value fashion. The key is used to identify the meta data entry. For the value of the key, sim-

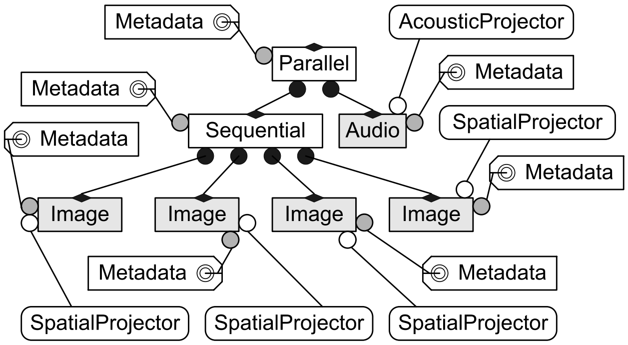

Figure 3. Multimedia Content Representation Tree with Semantic Annotation

ple data types such as string and integer are provided as well as more sophisticated data types. These allow for describing, e. g., the change of importance of an image that has been used in different presentations over a longer period of time. The multimedia content representation tree is also able to model the reference from a text asset to an image saying that the text is a semantic description of the image. As not all semantics derivation during the assemble process is perfectly reliable, a value of reliability is provided for each meta data entry. This value of reliability is in the interval of $]0, 1]$. The value of 1 is interpreted as hundred percent reliable, whereas values close to 0 are considered as very unreliable annotations. In addition, each meta data entry has a timestamp attribute in form of time and date. The timestamp stores the point in time when the meta data entry is created and the semantic information derived. This allows for tracking a possible change of a media asset's semantics over time (often referred to as *emergent semantics* [13, 19]). Also the source of creating the meta data entry is stored, i. e., the name of the concrete instance of the Metadata Derivation component that derived the particular piece of semantic information. The source of derivation is a unique name to avoid conflicts with other semantics derivation components. By this, each derivation component defines its own *namespace* in which it can add arbitrary meta data entries to the created multimedia content. Finally, the meta data entries associated to the composition tree nodes can be nested. Here, the value of a meta data entry is another meta data entry. By this, arbitrary hierarchies of semantical descriptions can be created.

4.3 Deriving Presentation Semantics

While organizing the media assets into structured multimedia content, new meta data can be derived for the created content at any point in time. This semantics derivation leverages the users' profile information and context information, the used media assets' meta data, as well as any other sources of semantically-rich information. This semantics derivation is provided by the Metadata Derivation component. It can be conducted for both the created multimedia presentations as well as for the single media assets used for the presentation (cf. Section 3).

The actual implementation of the instances of the Metadata Derivation components can employ different (semi-)automatic semantics derivation methods and techniques. A concrete Metadata Derivation component can internally use, e. g., rules, knowledge-bases such as taxonomies, thesauri, and ontologies, or might be plain programmed. Ontologies and taxonomies describe relations between different concepts of a domain. As shown in Section 2, they can be used to derive semantics based on the presence of some of the domain concepts. Rules can be applied, e. g., on the occurrence of a specific event. If a certain condition is present, a corresponding action in form of semantics derivation is conducted (Event-Condition-Action rules). An example of semantics derivation using such rules is presented with our demonstrator application in Section 5. The different semantics derivation methods and techniques are encapsulated by the interfaces of the Metadata Derivation component. This means that neither the concrete application using these components nor any other semantics derivation component knows about the internal realization of how the semantics is derived. For each piece of semantic information, a distinct derivation component should be implemented. This allows for an easy combination of different semantics derivations. In addition, the components can be—by their nature—easily exchanged and replaced by other ones. For example, in the domain of photo albums there might be a semantics derivation component detecting a title of a page. Based on the title detection, another component tries to detect sub-albums within the photo album and annotates the photo album with a table of content. At the same time, the title of a photo album page can be used as semantic description of the set of media assets placed on that page.

The semantics derivation components are triggered by the Multimedia Composition component. They can be triggered off at any point in time during the assemble process. In fact, there can be multiple assemble and derivate loops between the Multimedia Composition component and the different Metadata Derivation components. As the representation model of the multimedia content is tree-based, it naturally supports these loops by allowing to pass independent parts of the composition tree to the semantics derivation components.

4.4 Making the Semantics Explicit

The multimedia content and its semantics in the representation model is transformed by the Presentation Format Generators component into the syntax and features of the different presentation formats like SMIL, SVG, and Flash. For it, the component applies three application-independent transformation algorithms. The first two algorithms map the content and its layout to the target formats and have been elaborated in [23]. The third algorithm processes the semantics of the content representation and maps it to target formats. Finally, the results of all algorithms are assembled into the final multimedia presentation.

In the following, we describe in detail how the semantics of the created multimedia content in the representation tree is transformed and embedded into the syntax of the final presentation formats such as SMIL, SVG, and Flash. As output format of the semantics, we chose the Resource Description Framework (RDF) [29]. It supports references to other media nodes and composition nodes within the same document. In addition, RDF allows to easily model the additional attributes of the meta data entries such as reliability, source of derivation, and time and date. Finally, it can be serialized into the eXtensible Markup Language (XML) [32] and by this easily embedded into today's XML-based multimedia presentation formats. For example, SMIL provides with its Metainformation module (see [33, Sec. 8]) an approach that explicitly foresees the integration of XML-serialized RDF. As we will show, XML-serialized RDF can also be used with SVG as well as the binary presentation format Flash, which itself does not provide for any describing meta data of the presentation.

Mapping the Semantics to RDF The algorithm for processing and mapping the semantics into RDF starts with the root node of the representation tree containing the presentation's meta data. It traverses the tree in breadth first search, gathering each node's meta data and mapping them to RDF. Once all meta data entries of the multimedia content tree are transformed, the resulting RDF-document is serialized to XML using the Jena libraries[1].

An example of the transformation of the root node's meta data, i. e., the presentation properties is shown in Listing 1. First, the namespaces used by the SemanticMM4U framework's semantics derivation components are linked (lines 2 to 10). This is followed by a <rdf:Description> -block, containing the actual presentation properties. It contains basic presentation information such as the width, height, format, and creation time. In the example, the presentation is in Flash format, has a spatial extension of 800×600 pixels, and is created in October 2007 (lines 15 to 19). For these basic presentation properties, we use a predefined namespace provided by the SemanticMM4U framework (lines 6 to 7). As they are directly extracted from the presentation, no additional information such as reliability needs to be stored. However, the root node's meta data also comprises a title, description, and other information of the presentation such as the creator. As the title (as well as the description) might be (semi-)automatically derived from the creation process, it carries besides its actual value (line 22) also the source of derivation (line 23), its reliability (line 24 and 25), and its derivation time (line 26 and 27). The same applies for the description, creator, and others. For storing the title, description, and other information, we make use of Dublin Core elements. These are linked through the corresponding namespace (lines 4 to 5). Although, it might weaken the concept of separate namespaces for each derived piece of semantic information, we decided to use Dublin Core elements here as they are well supported by other systems and approaches.

Listing 1. RDF output of the presentation properties

```
   <rdf:RDF
     xmlns:rdf="http://www.w3.org/1999/02/22-
       rdf-syntax-ns#"
     xmlns:dc=
 5     "http://purl.org/dc/elements/1.1/"
     xmlns:pp="http://myframework/
       presentationProperties/1.0/"
     xmlns:me="http://myframework/
       metadataEntry/1.0/"
10   ...
     >
     <!-- root node properties -->
     <rdf:Description
       rdf:about="vacation.swf">
15     <pp:Width>800</pp:Width>
       <pp:Height>600</pp:Height>
       <pp:Format>Flash</pp:Format>
       <pp:Created>09.10.2007-14:33:18
         </pp:Created>
20     <dc:Title>
         <rdf:Description>
           <me:value>Norderney</me:value>
           <me:source>User</me:source>
           <me:reliability>100
25           </me:reliability>
           <me:creation_time>09.10.2007-
             14:33:16</me:creation_time>
         </rdf:Description>
       </dc:Title>
30     <dc:description> ...
         <me:value>Weekend Trip to
           Island Norderney</me/value>
         ...
       </dc:description>
35     ...
     </rdf:Description>
     ...
   </rdf:RDF>
```

As for the root node of the content representation tree, also for all other nodes of the tree a <rdf:Description> -block is created. An example of such a RDF-entry for an image is shown in Listing 2. The about -attribute indicates which part of the presentation is described, here the image with the ID image_4bc4321 (line 4). This internal ID used in the RDF document is also used in the content section of the concrete presentation formats. For example, the image would be defined in SVG as <image id="image_4bc4321" ...> and in SMIL as <img id="image_4bc4321" ...> . Thus, even in the target presentation formats, the meta data entries still point to the composition units to which they are associated in the representation model. In the example, a media asset stored at the location .../Norderney/Dunes.JPG is described (line 5). It has the caption "Beautiful Dunes" (lines 9 and 10). This is derived by the semantics derivation component ImageCaptionDerivation (lines 11 and 12). The reliability of the component is set to 90 percent (line 13). In future, this value will be refined based on analyses of real photo album data [4]. For storing the derived caption, we do not use a newly defined namespace, but exploit again the capabilities of Dublin Core by using the description element.

The second part of the listing shows the derivation of the physical location of an image with GPS

[1] http://jena.sourceforge.net/

data. This meta data derivation is conducted by applying the LocationDerivation component (lines 22 and 23). For the component, we defined the namespace drloc. It is embedded into the RDF document header as xmlns:drloc="http://.../derivationRules/drloc/1.0/" . Besides the actual name, the namespace also comes with version information (here 1.0). The example also shows, how meta data entries can be nested. The <rdf:Description> -block for the physical location name of the image has sub-entries for the concrete GPS values (lines 27 to 32).

Listing 2. RDF output describing an image

```
1  ...
2  <rdf:Description
3    rdf:about="http://.../
4      presentation/image_4bc4321">
5    <pp:URL>.../Norderney/Dunes.JPG</pp:URL>
6    <!-- Derived image caption -->
7      <dc:description>
8        <rdf:Description>
9          <me:value>Beautiful Dunes
10           </me:value>
11         <me:source>ImageCaptionDerivation
12           </me:source>
13         <me:reliability>90</me:reliability>
14         <me:creation_time>09.10.2007-
15           14:33:13</me:creation_time>
16       </rdf:Description>
17     </dc:description>
18   <!-- Derived location -->
19   <drloc:location>
20     <rdf:Description>
21       <me:value>Norderney</me:value>
22       <me:source>LoctionDerivation
23         </me:source>
24       <me:reliability>80</me:reliability>
25       <me:creation_time>09.10.2007-
26         14:33:13</me:creation_time>
27       <drloc:gpsData>
28         <rdf:Description>
29           <me:value>...
30           ...
31         </rdf:Description>
32       </drloc:gpsData>
33     </rdf:Description>
34   </drloc:location>
35   ...
36 </rdf:Description>
37 ...
```

Embedding RDF in the Presentation Formats Having described the mapping of the multimedia content's meta data into XML-serialized RDF, we now present how the semantics is integrated with the concrete presentation formats SMIL, SVG, and Flash. For embedding XML-serialized RDF in SMIL, the SMIL format's Metainformation Module is used [33]. With this module, additional meta data can be embedded to describe the SMIL presentation. The XML-serialized RDF document is embedded into the SMIL presentation's <header> tag by using the <metadata> tag. SMIL allows the embedded RDF document to refer to the single elements used in the presentation such as video elements and image elements. However, it also allows to point at SMIL's <seq> and <par> tags for the internal representation model's composition operators Sequential and Parallel . By this, the specific parts of the SMIL presentation are annotated by the RDF document. The same approach is applied for the mobile profiles of SMIL such as the SMIL Extended Mobile Profile.

The integration of XML-serialized RDF statements into SVG works in principle similar to the integration into SMIL. Like SMIL, it provides a <metadata> -tag that can be used to embed RDF into a SVG presentation. This tag is also part of the <header> -tag of a SVG presentation. The distinct parts of the SVG presentation such as the media assets are directly referred to by the semantic description. Also the mobile profiles of SVG, i. e., SVG Tiny and SVG Basic support embedded meta data.

Finally, the binary presentation format Flash does not allow for integrating any kind of meta data with the presentation content. Thus, we defined a markup language version of Flash, the FML (Flash Markup Language). The FML is basically a straightforward, XML-based serialization of the binary Flash content created. A full specification and description of this markup language is published in [20]. The RDF-based meta data serialized in XML is integrated with the FML. This document is then delivered with the binary Flash content as an additional, external file.

4.5 Summary

In this section, we presented the component architecture of the SemanticMM4U framework. We described in detail the semantics representation, derivation, and transformation into XML-serialized RDF, which is then embedded in the target presentation formats. If there is no additional semantics derived for the multimedia content during the assemble process, the RDF document created and embedded in the final presentations carries at least the already existing meta data of the media assets employed for the presentation. However, if there are meta data derived during the assemble process and associated with the content's nodes, the presentations in the final formats carry all these. By this, both the existing meta data as well as the derived meta data is preserved and made explicit with the created multimedia content in the final presentation formats. This allows other applications to process and to use the semantics and to overcome the semantics barrier.

Finally, the representation of multimedia content and their meta data in the presented SemanticMM4U framework can be embedded into the development of the MPEG-21 [5] standard. MPEG-21 provides a normative infrastructure for transferring and using digital (multimedia) content. Central part of this standard is the so-called Digital Item, a structured digital object that provides a combination of (media-)resources and meta data. This Digital Item could be used to ship the content and meta data provided by the SemanticMM4U framework.

5 Semantics Derivation in Personalized Multimedia Albums

So far, we have presented the architecture and components of the SemanticMM4U framework. We described in detail, how semantics is represented, derived, and made explicit in form of XML-serialized RDF. In this section, we present with the *x*SMART authoring tool a concrete application using the SemanticMM4U framework. It allows for a context-driven creation of multimedia albums [21]. A multimedia album is an extension of the traditional photo book by providing support for navigational interaction and continuous media assets such as audio and video [25]. For creating multimedia albums, some specific rules can be identified and applied regarding the album's semantics. These rules have been implicitly available with the thoughts and brain processes of the authors of multimedia albums. The challenge is to make this implicit semantics explicit and machine-processable with as less as possible intrusion and disturbance of the user. This is done with the *x*SMART authoring tool. For a detailed description of the identification and definition of the presented semantics derivation rules for multimedia albums, we refer to [25, 4]. A semi-formal specification of the rules in the Object Constraint Language (OCL) can be found in [4].

In the next Section 5.1, we briefly introduce the context-driven *x*SMART authoring tool for multimedia albums. In Section 5.2, we present how multimedia albums are created and annotated with semantically-rich information.

5.1 Authoring Tool for Multimedia Albums

In order to provide users with a powerful support for authoring multimedia albums, we developed the context-driven *x*SMART authoring tool [21]. This authoring tool uses the SemanticMM4U framework and different of its components. It provides an authoring wizard that supports the users in a step-wise creation of multimedia albums. This wizard employs different concrete Metadata Derivation components for semantically enriching the multimedia album. Once the wizard is finished with creating the album, the users can still manually edit and polish the presentation.

The *x*SMART authoring tool provides for calling the Metadata Derivation components during any step of the multimedia content creation process. This can be either during the wizard-guided creation of the album or when manually (post-)editing the content. For the latter, the *x*SMART authoring tool allows for firing events based on user input. These events are used to trigger appropriate semantics derivation components. For example, for a page of a multimedia album, the *x*SMART authoring tool specifies several events including those for adding, removing, and altering the page. It also defines media assets' events that are triggered when a media assets is added or removed. For visual media assets, the tool provides events for moving and resizing the assets. Acoustic media assets trigger a semantics derivation event, e. g., when the duration of an audio or video is changed. For example, when an image is enlarged or decreased in a multimedia album, an image_resize event is fired. This event triggers, e. g., a semantics derivation component valuating the image's importance based on its size [4]. When a text is entered, the text_added event is triggered. This calls, e. g., a derivation component detecting whether the text entered might be a caption of the image. Another derivation component determines whether the text is the title of a album page. If the text is detected as page title, it is added to the presentation's table of content.

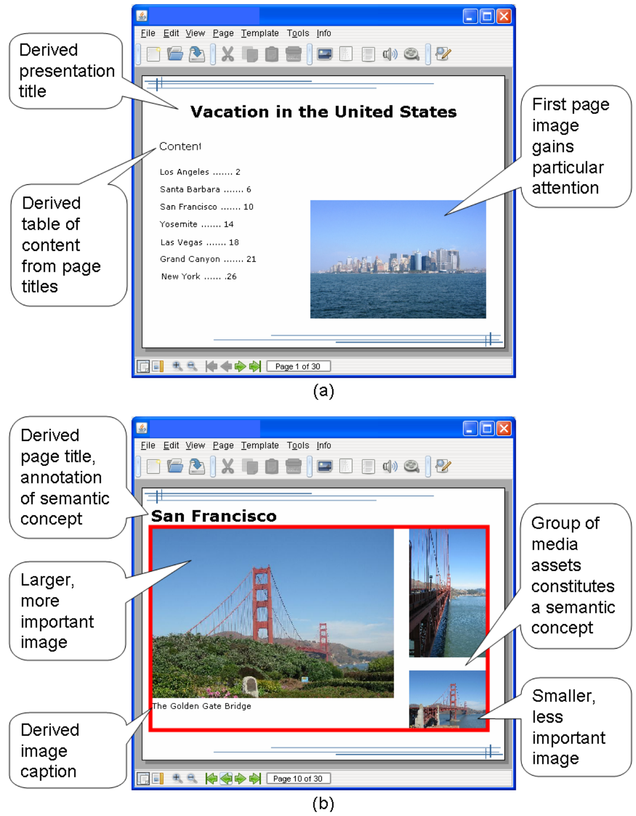

Figure 4. Semantics derivation for multimedia albums

5.2 Semantics in Multimedia Albums

The *x*SMART authoring tool provides semantics derivation components for many different aspects of multimedia albums' semantics. For example, we allow for deriving the page title, dynamically creating the table of content, and detecting image and video captions while authoring the multimedia albums. The importance of images and videos is determined based on their spatial size, i. e., larger images and videos are more important than smaller ones, and an image on the first page gains particular attention. If a set of media assets is placed on a distinct page of the album, these media assets are considered forming a semantic concept. The title of the album page is used as description of this semantic concept. Figure 4 shows these semantics derivation features for creating semantically-rich multimedia albums at the example of a group's vacation in the United States. The first page of the album is shown in Figure 4 (a). It illustrates the presentation title derivation and creation of table of content. Figure 4 (b) depicts the *x*SMART authoring tool features for determining a semantic concept, annotating it with the detected page title, determining whether a text is an image caption, and valuating the importance of images based on their size.

6 Conclusion and Future Work

In this paper, we presented the SemanticMM4U framework for authoring semantically-rich personalized multimedia presentations. The framework aims at helping application developers in designing and implementing their concrete systems for generating semantically-rich personalized multimedia content such as the presented *x*SMART authoring tool. With the help of the SemanticMM4U framework, semantics can be exploited and derived during the assemble process. In contrast to existing systems and approaches, this semantics is not thrown away once the content has been created but is made explicit and available by integrating it into the syntax and features of the different presentation formats such as SMIL, SVG, and Flash. By this, the created presentations' semantics becomes machine-readable and machine-processable. Thus, we overcome the semantics barrier. In addition, the SemanticMM4U framework supports back-storing the meta data derived for the media assets used in the presentations. Thus, it allows for using the derived semantics for future multimedia authoring tasks. The SemanticMM4U framework is available under open source license. It can be downloaded from *http://semanticmm4u.sourceforge.net/*.

Acknowledgments

The author would like to thank Holger Cremer for his efforts in implementing the SemanticMM4U framework and *x*SMART authoring tool. This research was supported by a Marie Curie International Fellowship within the sixth European Community Framework Programme.

References

[1] Adobe. Flash File Format Specification 7, 2004. http://www.adobe.com/licensing/developer/.

[2] Amazon, Inc, USA. amazon.com, 2006. Retrieved March 6, 2006, from http://www.amazon.com/.

[3] F. Bes, M. Jourdan, and F. Khantache. A Generic Architecture for Automated Construction of Multimedia Presentations. In *Multimedia Modeling*, Nov. 2001.

[4] S. Boll, P. Sandhaus, A. Scherp, and U. Westermann. Semantics, content, and structure of many for the creation of personal photo albums. In *Multimedia*. ACM, 2007.

[5] I. Burnett, R. van de Walle, K. Hill, J. Bormans, and F. Pereira. MPEG-21: Goals and Achievements. *IEEE MultiMedia*, 10(4):60–70, 2003.

[6] K. Cheverst, K. Mitchell, and N. Davies. The role of adaptive hypermedia in a context-aware tourist GUIDE. *Communications of the ACM*, 45(5), 2002.

[7] D. Ding, J. Yang, Q. Li, L. Wang, and L. Wenyin. Towards a flash search engine based on expressive semantics. In *World Wide Web Conf.* ACM, 2004.

[8] Dublin Core Metadata Initiative. Dcmi type vocabulary, Aug. 2006. Retrieved January 18, 2007, from http://dublincore.org/documents/dcmi-type-vocabulary/.

[9] G. P. Faconti and T. Rist, editors. *Proc. of ECAI-96 Workshop Towards a Standard Reference Model for Intelligent Multimedia Presentation Systems*, 1996.

[10] K. Falkovych and F. Nack. Context Aware Guidance for Multimedia Authoring: Harmonizing domain and discourse knowledge. *Multimedia Systems Journal*, 11(3), Mar. 2006.

[11] K. Falkovych, F. Nack, J. van Ossenbruggen, and L. Rutledge. Sample: Towards a framework for system-supported multimedia authoring. In *Multimedia Modelling*, page 362. IEEE Computer Society, 2004.

[12] J. Geurts, J. van Ossenbruggen, and L. Hardman. Application-specific constraints for multimedia presentation generation. In *Multimedia Modeling*. IEEE, Nov. 2001.

[13] W. I. Grosky, D. V. Sreenath, and F. Fotouhi. Deriving emergent semantics. In *Association for Information Systems, SIG on Semantic Web and IS*, Nov. 2004.

[14] R. Hjelsvold, S. Vdaygiri, and Y. Lé auté . Web-based personalization and management of interactive video. In *World Wide Web Conf.*, pages 129–139, 2001.

[15] T. Lemlouma and N. Layaïda. Adapted Content Delivery for Different Contexts. In *Symposium on Applications and the Internet*. IEEE, Jan. 2003.

[16] T. Lemlouma and N. Layaïda. Context-Aware Adaptation for Mobile Devices. In *Mobile Data Management*, Jan. 2004.

[17] S. Little, J. Geurts, and J. Hunter. Dynamic generation of intelligent multimedia presentations through semantic inferencing. In *Conf. on Research and Advanced Technology for Digital Libraries*. Springer, Sept. 2002.

[18] OAI. Open archives initiative, 2007. http://www.openarchives.org/.

[19] S. Santini, A. Gupta, and R. Jain. Emergent semantics through interaction in image databases. *IEEE Transactions on Knowledge and Data Engineering*, 13(3), 2001.

[20] A. Scherp. *A Component Framework for Personalized Multimedia Applications*. PhD thesis, Carl von Ossietzky University of Oldenburg, 2007. Retrieved January 3, 2007, from http://ansgarscherp.net/dissertation.

[21] A. Scherp and S. Boll. Context-driven smart authoring of multimedia content with xSMART. In *Proc. of the 13th annual ACM Int. Conf. on Multimedia; Hilton, Singapore*, pages 802–803. ACM, 2005.

[22] A. Scherp and S. Boll. MM4U - A framework for creating personalized multimedia content. In S. Nepal and U. Srinivasan, editors, *Managing Multimedia Semantics*, chapter 11, pages 246–287. Idea Group Publishing, 2005.

[23] A. Scherp and S. Boll. Paving the Last Mile for Multi-Channel Multimedia Presentation Generation. In Y.-P. P. Chen, editor, *Int. Conf. on Multimedia Modeling; Melbourne, Australia*, pages 190–197. IEEE, Jan. 2005.

[24] A. Scherp, S. Boll, and H. Cremer. Emergent semantics in personalized multimedia content. *J. of Digital Information Management*, 5(2), Apr. 2007.

[25] A. Scherp and R. Jain. Towards an ecosystem for semantics. In *Workshop on multimedia information retrieval on The many faces of multimedia semantics*. ACM, 2007.

[26] Special Issue of Comput. Stand. Interfaces on Standard Reference Model for Intelligent Multimedia Presentation Systems, Dec. 1997. vol. 18, no. 6–7.

[27] M. Umlauft, G. Pospischil, G. Niklfeld, and E. Michlmayr. LoL@, a Mobile Tourist Guide for UMTS. *Information Technology & Tourism*, 5(3):151–164, 2003.

[28] J. van Ossenbruggen, L. Hardman, J. Geurts, and L. Rutledge. Towards a multimedia formatting vocabulary. In *World Wide Web*. ACM, 2003.

[29] W3C. RDF/XML Syntax Specification, Feb. 2003. http://www.w3.org/TR/rdf-syntax-grammar/.

[30] W3C. Mobile SVG Profiles: SVG Tiny and SVG Basic, Mar. 2004. http://www.w3.org/TR/SVGMobile12/.

[31] W3C. SVG, Oct. 2004. http://www.w3.org/TR/2004/WD-SVG12-20040510/.

[32] W3C. XML, Feb. 2004. http://www.w3.org/TR/2004/REC-xml-20040204/.

[33] W3C. SMIL, Dec. 2005. http://www.w3.org/TR/2005/REC-SMIL2-20051213/.