

International Journal of Semantic Computing  
© World Scientific Publishing Company

## A Pattern System for Describing the Semantics of Structured Multimedia Documents

Ansgar Scherp<sup>1,2</sup>, Carsten Saathoff<sup>1</sup>, Stefan Scheglmann<sup>1</sup>

<sup>1</sup>*Institute for Web Science and Technologies*, <sup>2</sup>*Institute for Information Systems Research*  
*University of Koblenz-Landau, Koblenz, Germany*  
*scherp,saathoff,schegi@uni-koblenz.de*

Accepted (4 January 2012)

Today's metadata models and metadata standards often focus on a specific media type only, lack combinability with other metadata models, or are limited with respect to the features they support. Thus they are not sufficient to describe the semantics of rich, structured multimedia documents. To overcome these limitations, we have developed a comprehensive model for representing multimedia metadata, the Multimedia Metadata Ontology (M3O). The M3O has been developed by an extensive analysis of related work and abstracts from the features of existing metadata models and metadata standards. It is based on the foundational ontology DOLCE+DnS Ultralight and makes use of ontology design patterns. The M3O serves as generic modeling framework for integrating the existing metadata models and metadata standards rather than replacing them. As such, the M3O can be used internally as semantic data model within complex multimedia applications such as authoring tools or multimedia management systems. To make use of the M3O in concrete multimedia applications, a generic application programming interface (API) has been implemented based on a sophisticated persistence layer that provides explicit support for ontology design patterns. To demonstrate applicability of the M3O API, we have integrated and applied it with our SemanticMM4U framework for the multi-channel generation of semantically annotated multimedia documents.

### 1. Introduction

Media management comprises the activities of planning, organizing, conducting, and controlling the media content along its production and distribution process [43]. The archival and retrieval of the multimedia content becomes very hard if not practically infeasible if there is no or only limited metadata and annotations provided during the production process. While there exist various different metadata models [24, 3, 36, 22, 23] and metadata standards [18, 31, 2, 27, 38], we argue that these are not sufficient to annotate and provide support for the management of rich, structured multimedia documents.

Structured multimedia documents are understood as the coherent arrangement of media assets in time and space. They combine at least one continuous media asset like audio and video and one discrete media asset such as text and image [50]. Structured multimedia documents can be represented and rendered in presentations formats such as Flash [1], HTML5 [60], Scalable Vector Graphics (SVG) [57],

Lightweight Application Scene Representation (LASeR) [29], Synchronized Multimedia Integration Language (SMIL) [56] that is used for the Multimedia Messaging Service (MMS), or Extensible MPEG-4 Textual Omega (XMT- $\Omega$ ) [33]. Such structured multimedia documents are produced and management by organizations like online media production companies or media companies producing e-learning content or interactive, multimedia-based instruction manual for large medical devices and others. The annotation of such structured multimedia documents is considered the association of metadata to the structured content and the media assets used for assembling the structured document [37]. The existing metadata models and metadata formats are not sufficient to describe the semantics of such rich, structured multimedia documents. They are either conceptually too narrow, semantically ambiguous, focus on a specific media type only, or cannot be used and combined with each other. This situation is very unfortunate as the production and management of multimedia content is typically quite expensive. Providing a comprehensive support for representing the annotations of structured multimedia documents not only improves archival, retrieval, and overall management of the content, but also allows for better reuse of the multimedia documents within the organization.

With the Multimedia Metadata Ontology (M3O), we propose a generic modeling framework that accommodates and integrates the features provided by the different multimedia metadata models and metadata standards [14]. It allows to describe and represent the semantics of structured multimedia documents in a coherent and semantically precise manner. The M3O bases on Semantic Web technologies and thus is by design supported by today's presentation formats like Flash, SVG, and SMIL, which foresee in their specification the use of the Resource Description Framework (RDF) [54] for representing the metadata. For the design of the M3O, we have analyzed and extracted the common features and data structures that underlie existing metadata models and metadata standards. Each feature has then been implemented as distinct *ontology design pattern* (ODP) [16]. An ODP provides—similar to design patterns in software engineering [15]—a generic modeling solution to a recurrent ontology design problem [16]. The patterns of the M3O are not independent from each other. Rather, the ODPs of the M3O highly relate to each other and are designed to be applied together. Such a set of related patterns is called a pattern system [10]. Besides the patterns themselves, the system describes the dependencies and responsibilities of the patterns and how they are combined. The patterns of the M3O are based on the foundational ontology DOLCE+DnS Ultralight (DUL) [9] and axiomatized using description logics [4]. Being a generic modeling framework for multimedia metadata, the M3O can be applied in various domains where the management of multimedia metadata is required. By its rich axiomatization, the M3O is particularly designed to support capturing the metadata that are created during the media production process [19]. For publishing the multimedia documents and their metadata on the web, we assume that the M3O-based annotations of the multimedia content will be mapped to other metadata formats

such as the recently released Schema.org [18] vocabulary. Schema.org is an initiative and common vocabulary developed by Microsoft, Google, and Yahoo! and serves as markup language to annotate media assets on the web. As Schema.org does not provide all features of the M3O, the transformation will be lossfull, however. This can be accepted, as the Schema.org descriptions aim to be crawled and indexed by today's search engines in order to make media assets findable on the web. Once a specific multimedia document is found and a more detailed description of the document is requested, the original M3O metadata might be retrieved.

This paper extends the initial M3O [44] in various ways. First, we make the relations between the patterns more explicit in the sense of a pattern system [10]. To this end, we have refactored the M3O and extracted a common Provenance Pattern from the Annotation Pattern and Decomposition Pattern. Second, we introduce the Collection Pattern to describe collections of media assets. We discuss confidence information as information that describes the level of trust about some annotation, decomposition, or collection. Thus, it is related to provenance information. We show in the sense of a pattern system, how the different patterns like Annotation Pattern, Decomposition Pattern, and Collection Pattern can be combined with the Provenance Pattern. In addition, we demonstrate how the M3O can be combined with ODPs from other ontologies such as the Participation Pattern of the Event-Model-F [47]. A brief scenario in Section 2 motivates the different features of the pattern system M3O. Based on the requirements described in Section 3, we introduce the pattern system M3O in Section 4. The application of the pattern system is demonstrated in Section 5 at various examples taken from the scenario. Finally, we introduce the design and describe the application of a generic access mechanism to the M3O in Section 6. This access mechanism abstracts from the complexity of the pattern system and provides application developers an easy to use application programming interface (API). The API can be integrated with arbitrary multimedia applications that need to deal with multimedia metadata. As proof of concept, we have integrated and applied the API with our SemanticMM4U framework [46] for the multi-channel generation of semantically-rich multimedia presentations. An extensive analysis and comparison of existing metadata models and metadata standards is conducted in Section 7, before we conclude the paper.

## 2. Scenario

John works for a web company producing multimedia content such as interactive multimedia applications and games. Today, he is working on an interesting project for his son's school principal to provide a multimedia presentation about the history of nuclear energy, discussing its benefits and risks. He prepares a multimedia presentation as shown in Figure 1. The presentation consists of different parts. The

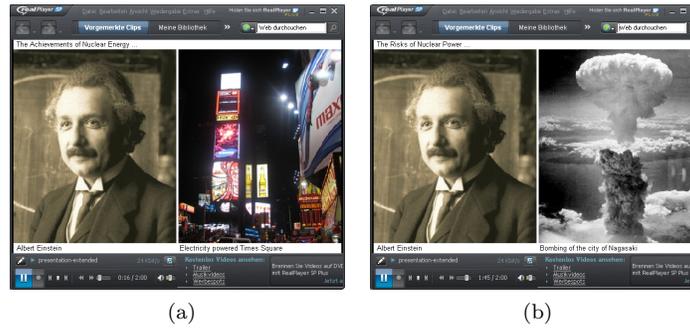


Fig. 1: Example of a Structured Multimedia Document

first part (cf. Figure 1a) shows a picture of Albert Einstein<sup>a</sup> and a photo of the Times Square in New York. It serves as metaphor for the achievements of nuclear energy. For the second part (cf. Figure 1b), John replaces the photo of the Times Square by a picture showing the atomic bombing of the city of Nagasaki in Japan in 1945. The contextual use of the picture of Einstein is now completely different and used as metaphor for the risks and destructive power of nuclear energy.

John uses a sophisticated multimedia authoring tool at work and starts annotating the different parts of his presentation (i) to express the positive and negative aspects of nuclear energy, respectively (ii). In addition, John annotates the pictures of Einstein, Times Square, and the atomic cloud with background information (iii). For example, he adds information about the historic event of the bombing of the city of Nagasaki in 1945. As John reuses some images from Wikipedia, he would like to point to the web resources where these images can be accessed (iv). Other images John uses are part of a larger collection he has been creating for a while (v) and wants to make his connection to the collection explicit (vi). John makes his confidence about the annotations explicit by adding corresponding descriptions to his presentation (vii). For example, the location annotation of the image showing the Times Square is annotated with a high level of trust as John has seen many other images of the times square before. This eases reuse of the image in the future, as its trustworthiness with respect to, e.g., the location where it has been captured, has already been confirmed. The authoring tool John uses adds him as creator of the presentation (viii), before the presentation together with the annotations is rendered in concrete presentation formats like SMIL and Flash (ix). Here, the authoring tool adds the information that the SMIL and Flash files are realizations of the same presentation to ease management of the multimedia documents (x).

<sup>a</sup>[http://en.wikipedia.org/wiki/File:Einstein1921\\_by\\_F\\_Schmutzer\\_4.jpg](http://en.wikipedia.org/wiki/File:Einstein1921_by_F_Schmutzer_4.jpg), the image is in the public domain.

### 3. Requirements on a Multimedia Metadata Model

From the scenario above and the related work discussed in Section 7, we identify the key features for representing and describing the semantics of structured multimedia documents. These requirements are:

*Information Objects and Information Realizations (REQ-1):* On the conceptual level, multimedia content conveys information to the consumer. As such, the multimedia content plays the role of a message that is transmitted to a recipient. Such a message can be understood as an abstract information object [9]. The multimedia content in our scenario above is, e.g., realized as SMIL and Flash presentation (cf. (ix) and (x)). This separation between information objects and information realizations is important for the management of multimedia documents.

*Annotation (REQ-2):* The M3O needs to support the annotation of both information objects and information realizations (cf. (ii)—(vi)). In our example, the image showing the Times Square is annotated with the geo-coordinates it was taken at. The location where an image has been taken does not depend on how it is realized, e.g., as JPG or other. Thus, the location annotation is applied on the information object level and not attached to the information realizations. On the contrary, some low-level metadata such as color information or the file size have to be attached to the information realization, since they depend on the concrete realization.

*Decomposition (REQ-3):* Structured multimedia documents can be decomposed into their constituent parts. In our example, the presentation is composed of the two parts discussing the chances of nuclear energy and its risks (cf. (i)). Each part can be again decomposed into the media assets they contain. In order to annotate only specific parts of a multimedia document, support for decomposition is needed. A clear separation between the information object and information realization is also important for the decomposition. For example, physically addressing the first part in a SMIL presentation is different from, e.g., Flash. Thus, addressing a component is depending on the concrete presentation format used as realization. In contrast, this does not matter for decomposing information objects.

*Collections (REQ-4):* Multimedia documents and in particular single media assets are often organized in collections that are about a specific topic. In the scenario, John uses media assets in his presentation that have been taken from a collection of images (cf. (v)). It shall be possible to annotate such collections of media assets. In addition, the provenance of contributions to such collections shall be made explicit, which is described next.

*Provenance (REQ-5):* Provenance is of crucial importance to assess the source of an information (cf. (viii)) and to judge its reliability. It is relevant for the annotation and decomposition of multimedia documents. In addition, it is relevant to collections of multimedia documents and media assets. In the scenario, John adds himself as creator of the image collection about the potentials and risks of nuclear energy. In addition, John is added as author of the presentation as he is the registered user of the authoring tool (viii).

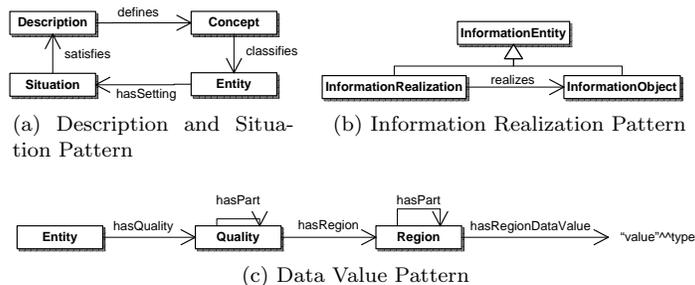


Fig. 2: Patterns of the foundational ontology DOLCE+DnS Ultralight

*Confidence (REQ-6)*: Confidence information is an explicit expression of a level of trust about some annotation, decomposition, or collection. Thus, it is related to provenance. In our scenario, John states with a high confidence that the image of the Times Square he uses in his presentation actually has been taken in New York at that place (cf. (vii)).

#### 4. Multimedia Metadata Ontology

For defining our Multimedia Metadata Ontology (M3O) based on the requirements above, we leverage Semantic Web technologies and follow a pattern-oriented ontology design approach [16]. Our goal is not to provide an ontological representation of a *specific* metadata standard or conceptual model but to provide a set of common ontology design patterns (ODPs) covering the description of multimedia documents in general. An ODP can be understood as a generic modeling solution to a recurrent ontology design problem [16]. The patterns of the M3O base on the foundational ontology DOLCE+DnS Ultralight (DUL) [9]. They have been specified using the Web Ontology Language (OWL) [53] and are formalized in Description Logics [4]. The patterns of the M3O highly relate to each other and are designed to be applied together. As such, the set of M3O patterns is called a pattern system [10] that describes the dependencies and responsibilities of the patterns. In Sections 4.1 to 4.3, we briefly introduce the patterns reused from DUL and provide some examples. Subsequently, we present the patterns of the M3O in Sections 4.4 to 4.7. Examples of the M3O patterns and their combination are provided in Section 5.

##### 4.1. Descriptions and Situation Pattern

The Descriptions and Situation (DnS) pattern [16] allows for a formally defined representation of contextualized views on relations. It assigns roles or types to a set of individuals. These roles or types are only valid within a particular context.

*Pattern Description*: The DnS pattern is depicted in Figure 2a and consists of a Situation that satisfies a Description. The Description defines the roles and types present in a context, called Concepts. Each Concept classifies an Entity. Each Entity is

connected to the situation via the `hasSetting` relation. Furthermore, the concepts can be related to other concepts by the `isRelatedToConcept` relation in order to express their dependency. The DnS pattern therefore expresses an n-ary relation among a set of entities.

*Example:* As an example, we consider a semantic annotation expressing a view on the relation of Albert Einstein and the development of a nuclear bomb. One might be interested in the source of this annotation. Using a simple relation is not suffice since additional information about the context in which this annotation is asserted needs to be provided. Using the DnS pattern, we express the fact that a relation between the media and its annotation is valid within the context of the user who provided the annotation. We can then restrict a search only to certain users or types of users we trust.

#### 4.2. Information Realization Pattern

The Information Realization Pattern models the distinction between information objects and information realizations [9]. Information objects represent abstract ideas that we aim to communicate. In order to communicate information objects, they have to be implemented, i.e., there have to be concrete information realizations of the information object.

*Pattern Description:* The Information Realization Pattern is depicted in Figure 2b and consists of the `InformationRealization` that is connected to the `InformationObject` by the `realizes` relation. Both are subconcepts of `InformationEntity`, which allows treating information in a general sense.

*Example:* The multimedia document in the scenario is realized as both a SMIL file and Flash file. But the message or abstract idea being communicated is the same. In other words, the same information can be realized in different ways and each realization communicates the same abstract idea or message.

#### 4.3. Data Value Pattern

In DUL there exists the concept `Quality` to represent intrinsic attributes of an `Entity`. `Regions` are used to represent the values of qualities. By this, we are able to model the kinds of data that describe entities as well as their concrete values.

*Pattern Description:* The Data Value Pattern depicted in Figure 2c assigns a concrete data value to an attribute of that entity. The attribute is represented by the concept `Quality` and is connected to the `Entity` by the `hasQuality` property. The `Quality` is connected to a `Region` by the `hasRegion` relation. The `Region` models the data space the value comes from. We attach the concrete value to the `Region` using the relation `hasRegionDataValue`. The data value is encoded using typed literals, i.e., the datatype can be specified using XML Schema Datatypes [5].

*Examples:* As an example, we model the EXIF metadata `brightnessValue` representing the brightness of an image. We model `Brightness` as a `Quality` since it is an attribute of the image. As `BrightnessRegion`, we use the real numbers and attach the

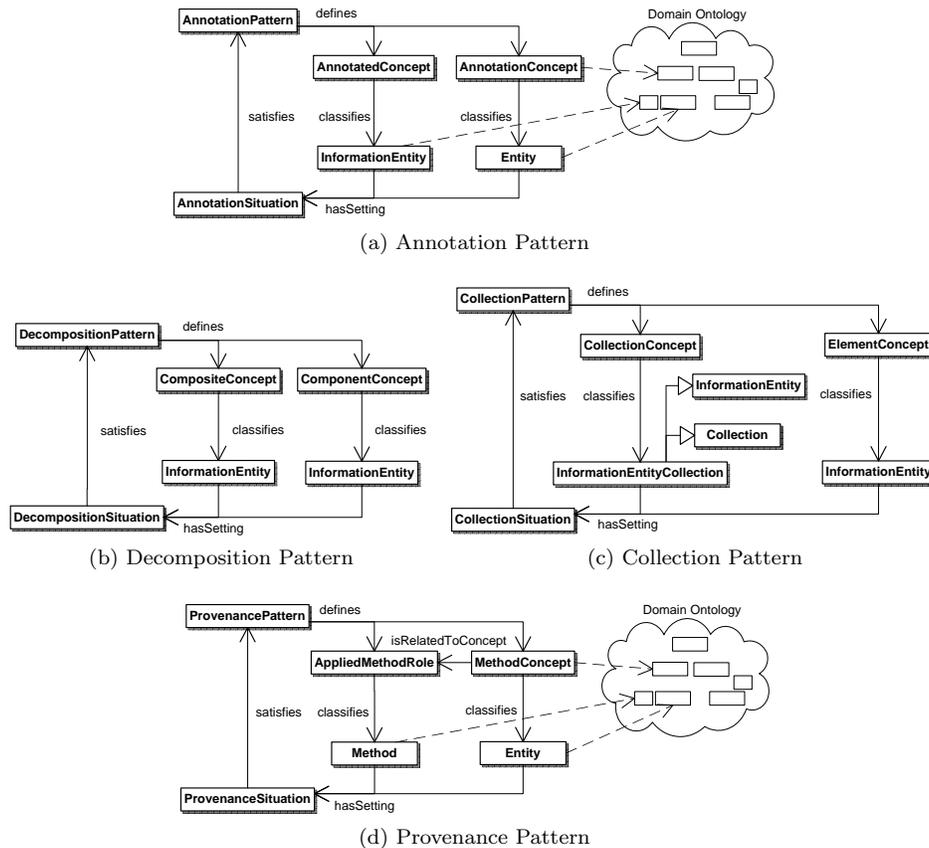


Fig. 3: Patterns of the Multimedia Metadata Ontology

concrete brightness value to the region using a specialization of `hasRegionDataValue` with a `minInclusive` and `maxInclusive` restriction for the interval  $[-99.99, 99.99]$ . Using the `hasPart` relation, we can also express structured data values like in MPEG-7.

#### 4.4. Annotation Pattern

Annotations are understood in M3O as the attachment of metadata to an information entity (see Section 1). Thus, annotations are metadata of information objects and information realizations, respectively. Metadata comes in various forms such as low-level descriptors obtained by automatic methods, non-visual information covering authorship and technical details, or semantic annotations (cf. Section 7). Our Annotation Pattern models the basic structures that underlies all types of annotation.

*Pattern Description:* The Annotation Pattern depicted in Figure 3a is a specialization of the DnS pattern. It consists of an `AnnotationSituation` that is a specialization

of the `Situation` concept and satisfies an `AnnotationPattern`, which is a specialization of the `Description` concept. The `AnnotationPattern` defines at least one `AnnotatedConcept` that classifies each `InformationEntity` that is annotated by an instantiation of this pattern. Thus, the annotation pattern allows to annotate multiple individuals of `InformationEntity` within a single instantiation of the pattern. The `InformationEntity` has the `AnnotationSituation` as its setting. Each metadata item is represented by an `Entity` that is classified by an `AnnotationConcept`. A metadata item is an `Entity` that is used for describing some `InformationEntity` [20]. There has to be at least one `AnnotatedConcept` defined by an `AnnotationPattern`.

*Discussion:* Instances of `Entity` can be defined in some domain ontology as indicated in Figure 3a. For example, an annotation entity of an image can be the `Creator` of an image defined in some domain ontology (cf. example in Figure 8). Besides the `Entity` that is used to annotate an `InformationEntity`, also the role this `Entity` plays in the annotation and the `InformationEntity` itself can be defined in some domain ontology. For example, there might be a specialization of the `AnnotationPattern` that introduces concrete media types such as `Image` or `Presentation` (cf. examples in Figures 5 and 8). The provenance of an annotation can be explicitly captured by using the `ProvenancePattern` described in Section 4.7.

#### 4.5. *Decomposition Pattern*

The `DecompositionPattern` models the decomposition of information entities, e.g., the decomposition of a SMIL presentation into its logical parts. After a decomposition, there is a whole, called the composite, and there are the parts, called the components. The provenance of a decomposition can be modeled using the `ProvenancePattern` described in Section 4.7.

*Pattern Description:* The `DecompositionPattern` shown in Figure 3b introduces a `DecompositionPattern` as specialization of the `Description` concept and a `DecompositionSituation` as specialization of the `Situation` concept. The `DecompositionPattern` defines exactly one `CompositeConcept` and at least one `ComponentConcept`. The `CompositeConcept` classifies an `InformationEntity`, expressing that it is the whole. Each `ComponentConcept` classifies an `InformationEntity`, asserting that they are parts of the whole. All classified entities have the `DecompositionSituation` as setting.

*Discussion:* It is important to note that in cases of structured multimedia documents there is already composition information available in the media itself. A SMIL file, e.g., contains information about how single media assets are arranged. However, with the M3O, we aim at representing metadata about parts of the media that are not necessarily equal to the physical structure of the presentation.

The W3C Media Fragments working group provides a mechanism to address specific parts of videos [52]. When using an URI in the syntax of the Media Fragments working group in the `DecompositionPattern`, a specialization `FragmentComponentConcept` of the `ComponentConcept` should be used to indicate the use of such a media fragment URI.

#### 4.6. *Collection Pattern*

The Collection Pattern allows to represent an arbitrary collection of information entities that are semantically coherent. This set of information entities share one or more common properties. For example, John has created a collection of images showing scientists.

*Pattern Description:* The Collection Pattern depicted in Figure 3c introduces a `CollectionPattern` as specialization of the `Description` concept and a `CollectionSituation` specializing the `Situation` concept of the DnS pattern. The `CollectionPattern` defines exactly one `CollectionConcept` classifying an `InformationEntityCollection`. An `InformationEntityCollection` represents the actual collection and is a specialization of `InformationEntity` and `Collection`. Finally, the `CollectionPattern` defines some `ElementConcepts` that classify `InformationEntity`, i.e., the elements of the collection.

*Discussion:* Please note that collections and decompositions express different mereological relations between entities. Thus, the `InformationEntityCollection` is a first-order entity. It might be annotated using the Annotation Pattern described in Section 4.4 or decomposed using the Decomposition Pattern as introduced in Section 4.5. Also provenance of the items in the collection and the collection itself can be added by using the Provenance Pattern, which is described below.

#### 4.7. *Provenance Pattern*

With the Provenance Pattern, we allow for an explicit representation of provenance information along with a reason or justification of this information. The Provenance Pattern can be applied to describe the source of annotations, decompositions, as well as collections of information entities.

*Pattern Discussion:* The Provenance Pattern is depicted in Figure 3d. It introduces a `ProvenancePattern` specialized from the `Description` concept and a `ProvenanceSituation` that is a subconcept of `Situation`. The `ProvenancePattern` defines one or more `AppliedMethodRoles` and optional `MethodConcepts`. The `AppliedMethodRole` classifies a `Method` that represents how an annotation, decomposition, or collection has been created. For example, it can be an algorithm for image segmentation or a manual annotation. The optional `MethodConcepts` defined by the `ProvenancePattern` provide further details of the applied `Method`. The `MethodConcept` classifies an `Entity` such as the parameters used when applying the `Method`. Thus, the `MethodConcepts` are related to the `AppliedMethodRole`, which is represented using the property `isRelatedToConcept`. Besides representing parameters used to run an automatic algorithm, the `Entity` can also point to a model used for some classifier, give a high-level explanation in case of a manual annotation, or represent the creator of a collection. In case of concrete data values for the `Entity` parameters, the Data Value Pattern described in Section 4.3 is used. To express to which entities the provenance information refers to, the relation `isRelatedToConcept` pointing from the `AppliedMethodRole` to, e.g., the `AnnotationConcept` of the Annotation Pattern is used.

*Confidence Information and Discussion:* Closely related to provenance is the notion of confidence (see requirements in Section 3). Confidence information can be provided by some algorithms and explicitly represented using the Provenance Pattern’s Entity. For example, an image classifier potentially knows its performance in classifying images in a set of concepts it supports. To represent concrete confidence values, the Data Value Pattern is applied.

Please note that the Provenance Pattern can incorporate existing provenance ontologies. For example, the Provenance Vocabulary Mappings [58] of the W3C Provenance Incubator Group can be used as demonstrated in Figure 5.

## 5. Modeling the Scenario using the M3O

Having introduced the M3O, we demonstrate the application of the pattern system to the scenario from Section 2. To this end, the following examples describe what happens when John creates and annotates his presentation and which patterns of the M3O are used. The concrete objects are referred to as individuals. Each individual has a type that refers to a concept in some ontology. Each box in the following diagrams represents an individual and its type. For example, the box labeled with `presentation-realization-1:SMILFile` in Figure 4 refers to the individual `presentation-realization-1` of type `SMILFile`. Concepts and individuals are identified by URIs. However, for presentation reasons we omit the namespace.

*Example 1 - Information Realization:* The example shows how to apply the Information Realization Pattern to represent information objects and information realizations. In the example, we consider two realizations of our presentation, namely one in SMIL and one in Flash. In Figure 4, we see the individual `presentation-1` of type `Presentation`, which is a subclass of `InformationObject`. The files are represented by the individuals `presentation-realization-1` and `presentation-realization-2`, which realize the presentation. They are of type `SMILFile` and `FlashFile`, which are subclasses of `InformationRealization`.

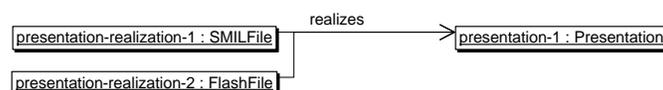


Fig. 4: Two realizations of a multimedia presentation

*Example 2 - Annotation of the Entire Presentation:* In the next step, we annotate the entire presentation with its topic, represented by a Wikipedia article on risk society using the Annotation Pattern. As shown in Figure 5, the `AnnotatedConcept` classifies the individual `presentation-1` and expresses that this is the information object being annotated. The `AnnotationConcept` classifies the individual `Risk.Society` from DBPedia, a Semantic Web representation of the Wikipedia article. We are not limited to DBPedia, but can use any source of domain knowledge.

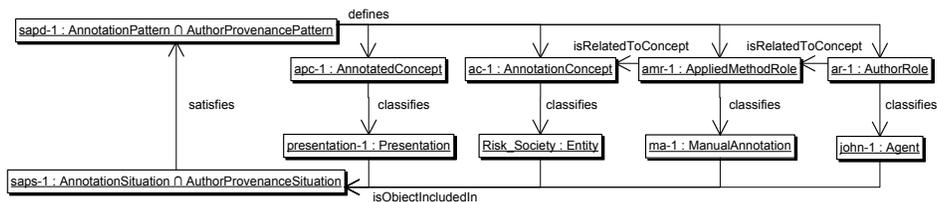


Fig. 5: Semantic annotation of the whole presentation

The example shows the benefit of using the DnS-based approach. We cannot only express the annotation as a relation between the information object and its annotation, but we can treat this relation within context. We exemplify the support for context by adding provenance information about the creator of the annotation. The `AppliedMethodRole` classifies a `ManualAnnotation`, and thus expresses that this presentation was labeled manually. We specify the author of this annotation by classifying some individual `john-1` using the `AuthorRole`, a specialization of `MethodConcept`. The `john-1` is an `Agent`, a concept taken from the Provenance Vocabulary Mappings [58] of the W3C Provenance Incubator Group. The `AuthorRole` `isRelatedToConcept` the `AppliedMethodRole`, expressing that `john-1` is the author of this manual annotation. The concepts `ManualAnnotation` and `AuthorRole` are subconcepts of the DOLCE+DnS Ultralight concepts `Method` and `Entity`. They are provided by some domain-specific ontology specializing the M3O. To ease retrieval of the patterns, the concepts `ProvenancePattern` and `ProvenanceSituation` are specialized to `AuthorProvenancePattern` and `AuthorProvenanceSituation`. However, as we do not introduce specializations of concepts defined in the Annotation Pattern, we do not specialize the `AnnotationPattern` and `AnnotationSituation` concept. In general, we introduce specializations of the M3O pattern when we introduce specializations of the concepts defined by the pattern.

*Example 3 - Decomposition:* In Figure 6, we show the logical decomposition of the presentation into two parts representing the positive and negative aspects of nuclear energy, respectively. We further demonstrate the decomposition of the first part into the images of Albert Einstein and the Times Square.

The upper part of Figure 6 shows the first composition, the lower half the second one. We see that the `DecompositionPattern` defines the `CompositeRole` and two `ComponentRoles`. The `CompositeRole` classifies the individual `presentation-1`, i.e., the information object representing our presentation. This relation represents the fact that the presentation is the `Composite`, i.e., the *whole*. The `ComponentRoles` classify the two `InformationObjects` named `part-1` and `part-2`, representing the two logical *parts* of the presentation. The lower part of Figure 6 shows how `part-1` is further decomposed into the two `Images`, represented by `einstein-1` and `times-square-1`. The concept `Image` is a specialization of `InformationObject` representing images. We see that the individual `part-1` plays the `ComponentRole` in the first composition and the `Compos-`

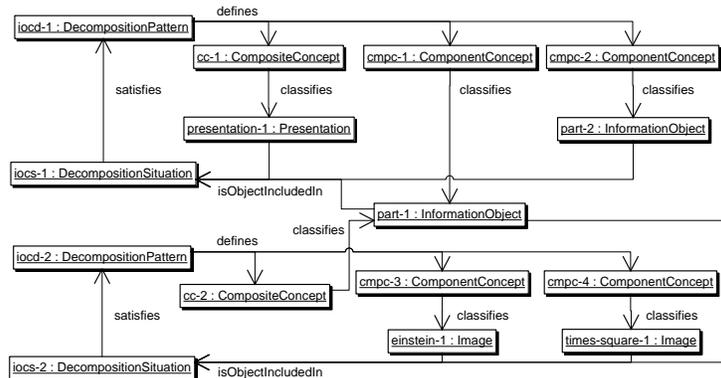


Fig. 6: Two-layered decomposition of the presentation

iteRole in the second. Being a component or a composite is therefore depending on the context.

*Example 4 - Annotation of an Information Object:* We demonstrate the annotation of an image with the location where it has been taken using the GPS-based location extracted from EXIF [31]. EXIF allows to embed mainly technical metadata directly into images, e.g., in JPEG files. As shown in Figure 7, the annotation is conducted by the `EXIFAnnotationPattern`, a specialization of the `AnnotationPattern` for EXIF. In addition, the `ProvenancePattern` is applied to provide the provenance of the annotation. EXIF metadata mainly focuses on the capturing conditions of an image. Thus, it might be tempting to attach all EXIF metadata to the information realization. But in fact, EXIF provides many metadata that does belong to the information object level such as camera model, camera make, location, date, and time. Even if we transform the original image into other formats, the camera model used to take the image and the other properties do not change. Other metadata such as the dimensions and color depth of the image, however, are properties of the realization. Existing ontologies for EXIF [32] just performed a one-to-one mapping of the EXIF tags to RDF properties and thus do not add further semantics. Consequently, we have decided to build our own ontology and have manually mapped the EXIF tags to information objects and information realizations [14].

Figure 7 depicts the annotation of an `Image` information object with the location where the image was taken. We attach the location information to the information object, since the location is independent of the image realization. In order to represent the coordinates, we employ the `Data Value Pattern` and introduce the `GeoLocationQuality`. In addition, the `AnnotationConcept` is specialized to `EXIFGeoLocationParameter` that parametrizes a `Point` from the WGS84 vocabulary [55]. The concept `Point` is used here as `Region` and represents the data space of all geo coordinates. We use the WGS84 properties `geo:lat` and `geo:long` to represent the latitude and longitude of the place, where the image has been taken. Please note that the

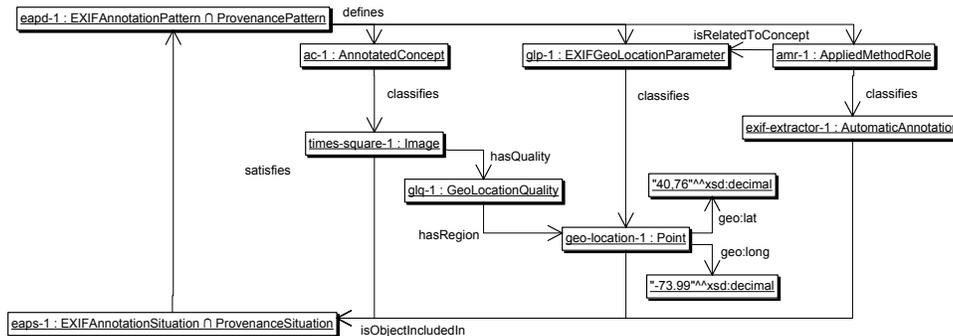


Fig. 7: Representing geo-information extracted from the EXIF header

Region, the Quality, and the WGS84 relations are not specific to the EXIF descriptor. They could be reused in other annotations that represent geolocations. The provenance of the location information is the `exif-extractor-1`, a common EXIF extractor tool. Thus, it is an `AutomaticAnnotation` classified by an `AppliedMethodRole`.

An alternative way of modeling the geolocation would be introducing subregions for latitude and longitude to the region `Point` using the `hasPart` relation of DUL. These sub-regions `LatitudeRegion` and `LongitudeRegion` would then each have a `hasRegionDataValue` for the latitude and longitude information. However, it is questionable that introducing two sub-regions for modeling the latitude and longitude of a geo point is of added value for the knowledge representation. Thus, a trade-off is needed between how deep one wants to model the knowledge while still allowing an efficient processing of the data.

*Example 5 - Annotation of an Information Realization:* The annotation of concrete realizations of the information object `times-square-1` is very similar to the annotation shown in Figure 7. Instead of annotating the information object `times-square-1`, the realizations of this image information object, namely individuals of `ImageRealizations` are annotated. Metadata that is typically associated with these realizations are the image resolution and the location where the file is stored. A figure depicting the annotation of image realizations is omitted for reasons of brevity.

*Example 6 - Annotation with Provenance and Confidence:* The example depicted in Figure 8 shows the annotation of the image of Einstein `einstein-1` with information about its `Creator`, namely Ferdinand Schmutzer represented by the individual `F.Schmutzer`. The provenance is a `ManualAnnotation` conducted by `john-1`, which is modeled using the `ProvenancePattern`. As the actual annotation information is provided by Wikipedia, John states a high confidence to his annotation. John's `ManualAnnotation` has a `ConfidenceQuality` that uses as `ConfidenceRegion` the interval of  $[0, 1]$ . The confidence is modeled using the `float` datatype [59] and applying `minInclusive` and `maxInclusive` restrictions on it.

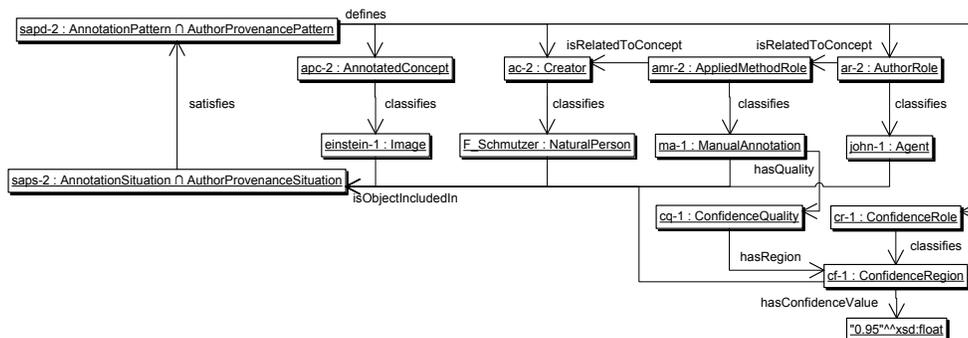


Fig. 8: Einstein image annotated with its creator and high confidence by John

*Example 7 - Annotation with Complex Information:* The Annotation Pattern can also be used to annotate an `Image` (or other information entity) with some complex knowledge. The example depicted in Figure 9 demonstrates the annotation of the image `nagasaki-cloud-1` with some complex annotations. The image `nagasaki-cloud-1` shows the atomic cloud of the bombing of the city of Nagasaki. It is annotated with the event of that bombing using the Participation Pattern of the ontology Event-Model-F [47]. The Participation Pattern captures the objects participating in an event. It also allows to represent the time and location when this event happened (not shown in the figure for reasons of brevity). As Figure 9 shows, the Participation Pattern is connected with the Annotation Pattern through the common `EventParticipationSituation`. The Participation Pattern defines the event `nagasaki-bombing-1` and the different objects that have participated in the event. These are `sweeney-1`, the 393rd Squadron commander Major Charles W. Sweeney who was the Pilot of the B-29 airplane `superfortress-bockscar-1` that dropped the bomb called `fat-man-1` onto the city of Nagasaki on August 9, 1945.

*Example 8 - Image Collection:* Finally, the example depicted in Figure 10 demonstrates the use of the Collection Pattern and Provenance Pattern. John’s colleague Marie has added some images to John’s `ImageCollection` of scientists. The `collection-1` has several elements of which two of them are shown in the figure. One is the individual `einstein-1`, an image of Einstein contributed by John. The other one is an image of Wernher von Braun `von-braun-1` provided by Marie.

## 6. Application Programming Interface to the M3O

In order to make use of ontologies in concrete scenarios, they need to be accessible from an object-oriented software application. To this end, we have developed an easy to use access mechanism to the M3O in form of an application programming interface (API). The API abstracts from the complexity of the pattern system and provides application developers full access to the functionality provided by the patterns such as annotation, decomposition, and information realization. The actual

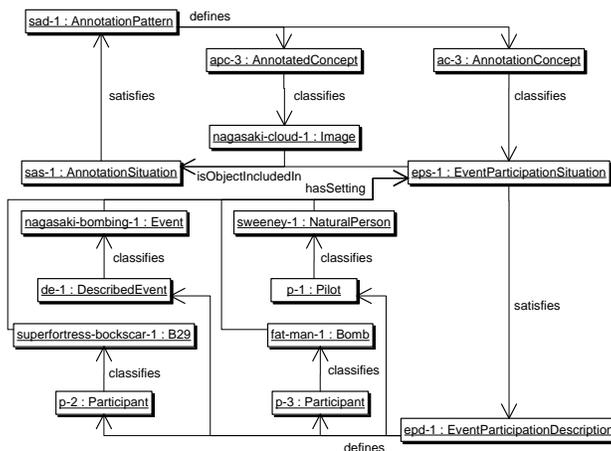


Fig. 9: Combined use of M3O’s Annotation Pattern with F’s Participation Pattern

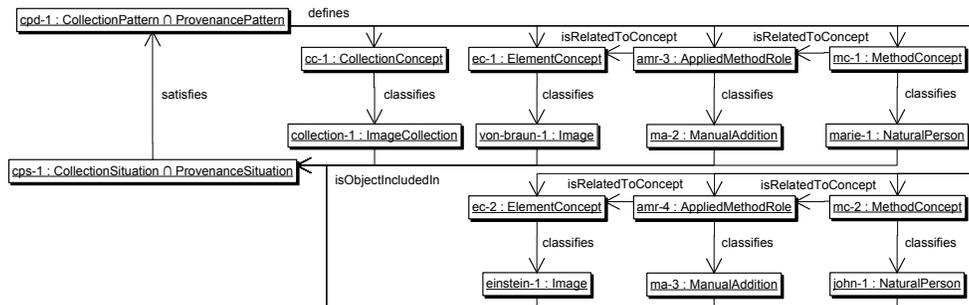


Fig. 10: Using the collection pattern for representing an image collection of scientists

design and axiomatization of the M3O is hidden from the application developer. Thus, the application developers can focus on developing the metadata management features of their concrete application.

For the design of the M3O API, a core API layer has been developed using our persistence layer Winter [45]. Winter is a generic tool for mapping RDF to Plain Old Java Objects (POJOs). It provides a declarative language to annotate Java classes with SPARQL graph patterns. This declarative language provides explicit support for ontology design patterns. By this, Winter exceeds the capabilities of other mapping tools like ActiveRDF [42] and Jastor (<http://jastor.sourceforge.net/>). We have introduced a single Java class for each M3O pattern and added Winter annotations to it. If needed, the classes of the API can be specialized by application developers to reflect more specific requirements of a particular domain or used as is. In the context of this work, we focus on describing how to use the M3O API without modifications.

Listing 1 shows the Winter annotations to the class `AnnotationPattern.java` implementing the Annotation Pattern. The annotation has two parameters, one for the annotation type and one for the query. In the example, the type is `PATTERN` and the query parameter contains a SPARQL graph pattern to retrieve an Annotation Pattern from the triple store (lines 1-8). Please note that `dul` is the namespace of DUL, namely `http://ontologydesignpatterns.org/ont/dul/DUL.owl`. The namespace of RDF `http://www.w3.org/1999/02/22-rdf-syntax-ns` is represented as `rdf`. The class `AnnotationPattern` implements an interface `RdfSerialisable` that abstracts from the access to the Sesame (`http://openrdf.org/`) triple store. The information entity being annotated is marked as `EXTERNALOBJECT` (lines 11-12). This means, that the information entity is provided as separate class in the API and thus is visible to the application developers. In contrast, other concepts like the `AnnotationDescription` and `AnnotationSituation` are hidden by the API implementation (not shown in the listing). By this, the API abstracts the complexity of the M3O to the application developers. The annotations themselves are represented using an internal class called `Annotation`, which is annotated with the Winter-type `MAPPING` (lines 14-17). A `MAPPING` describes the connection between shared parts of the knowledge representation provided by different classes of the API. The mapping is specified through the parameters `src` and `dst`. Provenance information is added to the annotation using the `ProvenancePattern` class (19-25). Further below, we find getter and setter methods for the entity being annotated, adding annotations, and adding the Provenance Pattern (lines 28-40).

```

1  @winter( type = winter.Type.PATTERN, query =
2    "?AnnotationDescription dul:defines ?AnnotatedIEConcept." +
3    "?AnnotationSituation dul:satisfies ?AnnotationDescription." +
4    "?AnnotatedIEConcept dul:classifies ?InformationEntity." +
5    "?InformationEntity dul:hasSetting ?AnnotationSituation." +
6    "?AnnotationSituation rdf:type ?AnnotationSituationType." +
7    "?AnnotationDescription rdf:type ?AnnotationDescriptionType." +
8    "?AnnotatedIEConcept rdf:type ?AnnotatedIEConceptType." )
9  public class AnnotationPattern implements RdfSerialisable {
10     // Information entity being annotated
11     @winter(type = winter.Type.EXTERNALOBJECT, var = "InformationEntity")
12     InformationEntity informationEntity = null;
13     // Annotations are represented internally as Set<Annotation>
14     @winter(type = winter.Type.MAPPING,
15       src = {"AnnotationDescription", "AnnotationSituation"},
16       dst = {"AnnotationDescription", "AnnotationSituation"})
17     Set<Annotation> annotations = new HashSet<Annotation>();
18     // Provenance information
19     @winter(type = winter.Type.MAPPING,
20       src = {"AnnotationDescription", "AnnotationDescriptionType",
21         "AnnotationSituation", "AnnotationSituationType"},
22       dst = {"Description", "DescriptionType",
23         "Situation", "SituationType"})
24     Set<ProvenancePattern> provenances =
25       new HashSet<ProvenancePattern>();
26
27     // Getter/setter of the information entity being annotated
28     public InformationEntity getInformationEntity() { ... }

```

```
29 public void setInformationEntity(InformationEntity infEntity) { ... }
30 // Adding annotations
31 public void addAnnotation(Entity annotation ,
32     URI annotationConcept ){ ... }
33 public Set<Entity> getAnnotations(){ ... }
34 ...
35 // Adding the Provenance Pattern
36 public Set<ProvenancePattern> getProvenance() { ... }
37 public void addProvenance(ProvenancePattern provenance) { ... }
38 ...
39 }
40
41 }
```

Listing 1: Winter annotations and query patterns of the Annotation Pattern

The Annotation class shown in Listing 2 is internally used for managing the annotations of the Annotation Pattern. It has been introduced to ease dealing with the annotations and covers the AnnotationConcept connected via the classifies relation to the Entity concept. Thus, the knowledge covered by the Annotation class can be considered a *sub-pattern* of the Annotation Pattern. The application developer, however, only uses the AnnotationPattern class and is not burden with this internal implementation detail. Like the AnnotationPattern class itself, the Annotation class is annotated as shown in Listing 2 with a Winter type and query (lines 1-5). The entity used for annotation and its type are defined (lines 8-14) and appropriate methods for accessing them are provided (lines 17-22).

```
1 @winter(type = winter.Type.PATTERN, query =
2     "?AnnotationDescription dul:defines ?AnnotationConcept." +
3     "?AnnotationConcept dul:classifies ?Entity." +
4     "?Entity dul:hasSetting ?AnnotationSituation." +
5     "?AnnotationConcept rdf:type ?AnnotationConceptType." )
6 public class Annotation implements RdfSerializable {
7     // Entity serving as annotation
8     @winter(type = winter.Type.EXTERNALOBJECT, var = "Entity")
9     Entity annotationEntity = null;
10    // URI of the annotation concept and the type of annotation
11    @winter(var = "AnnotationConcept")
12    protected IndividualURI annotationConcept = null;
13    @winter(var = "AnnotationConceptType")
14    protected ClassURI annotationConceptType = null;
15
16    // Methods for creating an annotation
17    public Annotation(Entity annotationEntity){ ... }
18    public Annotation(Entity annotationEntity ,
19        URI annotationConcept){ ... }
20    public Entity getAnnotationEntity() { ... }
21    public void setAnnotationEntity(Entity annotationEntity) { ... }
22    ...
23 }
```

Listing 2: Internal representation of an annotation

Finally, the implementation of the ProvenancePattern class is shown in Listing 3. The full SPARQL graph pattern is omitted for reasons of brevity. Like the previous

examples, it defines a provenance entity and provenance method (lines 4-8) and appropriate methods (lines 12-19). Below, we show how the API is used to instantiate and combine the Annotation Pattern and Provenance Pattern.

```
1 @winter(type = winter.Type.PATTERN, query = "...")
2 public class ProvenancePattern implements RdfSerialisable {
3     // Provenance entity
4     @winter(type = winter.Type.EXTERNALOBJECT, var = "ProvenanceEntity")
5     Entity provenanceEntity = null;
6     // Provenance method
7     @winter(type = winter.Type.EXTERNALOBJECT, var = "Method")
8     Method method = null;
9
10    // Methods for creating provenance information
11    public ProvenancePattern(Entity provenanceEntity,
12        Method method) { ... }
13    public void setProvenanceEntity(Entity provenanceEntity) { ... }
14    public Entity getProvenanceEntity() { ... }
15    public void setMethod(Method method) {
16        public Method getMethod() {
17        public void setAppliedMethodRole(IndividualURI methodRole) { ... }
18        public IndividualURI getAppliedMethodRole() { ... }
19        ...
20    }
```

Listing 3: Winter annotations and query patterns of the Provenance Pattern

The M3O API allows for adding support to represent and manage rich semantic descriptions of multimedia documents into arbitrary multimedia applications. To demonstrate applicability of the API, we have integrated and used it with the SemanticMM4U framework [46] for the multi-channel generation of semantically-rich multimedia presentations in formats like SMIL, SVG, Flash, and others. The framework has been successfully applied for the development and generation of rich, structured multimedia presentations in application domains such as personalized sports news, context-aware tourist guides, and the generation and semantic enrichment of personal photo albums [8]. The SemanticMM4U framework provides different composition operators such as `sequential(...)` and `parallel(...)` and allows to define domain-specific composition operators to create multimedia documents as the example multimedia presentation in the scenario in Section 2. Once the multimedia document is composed by the operators of the SemanticMM4U framework (not shown for reasons of brevity), the multimedia document can be annotated with metadata as shown in Listing 4. Finally, the multimedia document is transformed into various target formats like the multimedia presentation formats SMIL, SVG, Flash, and others.

An example of using the M3O API in the SemanticMM4U framework is shown in Listing 4. The `annotationBase` is the location of the presentation generated by the SemanticMM4U framework. The `presentationExtension` is a specialization of the M3O and defines concepts like `Presentation` as specialization of `InformationEntity`, `Image` and `ImageRealization` as specializations of `InformationObject` and `InformationRealization`, and

Point and GeoLocationQuality as specializations from the Point concept of the WGS84 vocabulary and the generic Quality concept. In the example, first, the information object of an image showing Albert Einstein is annotated with a resource from DBpedia using the AnnotationPattern class (lines 7-10). Subsequently, the provenance of this annotation is created using the ProvenancePattern class (lines 13-25). The ProvenancePattern is added to the AnnotationPattern using the method addProvenance(...) (line 26). The annotation and its provenance are made persistent using the addObject(...) method of Winter's metadata mapper (line 28).

```
1 String annotationBase = "http://m3o.semantic-multimedia.org/" +
2   "presentation/m3o.realPlayer.smil.rdf#";
3 String presentationExtension = "http://m3o.semantic-multimedia.org/" +
4   "m3o-presentation-extension.owl#";
5
6 // Annotation of an image showing Einstein
7 AnnotationPattern ap1 = new AnnotationPattern(annotationBase);
8 ap1.setInformationEntity( einsteinImageO );
9 ap1.addAnnotation(new EntityImpl( new IndividualURI(
10   "http://dbpedia.org/resource/Albert_Einstein" ) ), null, null);
11
12 // Add provenance information
13 ProvenancePattern provenance = new ProvenancePattern(
14   new MethodImpl(
15     new IndividualURI( Utils.createURI(annotationBase, "ManualAddition") ),
16     new ClassURI( presentationExtension + "ManualAddition" ) ),
17   new EntityImpl(
18     new IndividualURI( annotationBase + "john-1" ),
19     new ClassURI( "http://www.ontologydesignpatterns.org/ont/dul/" +
20       "DUL.owl#Agent" ) )
21 );
22 provenance.setDescription(new IndividualURI(
23   Utils.createURI(annotationBase, "ProvenancePattern" ) ));
24 provenance.setAppliedMethodRole(new IndividualURI(
25   Utils.createURI(annotationBase, "AppliedMethodRole" ) ));
26 ap1.addProvenance(provenance);
27
28 metadataMapper.addObject(ap1); // Make metadata persistent
```

Listing 4: Use of the M3O API in the SemanticMM4U framework

## 7. Related Work

Metadata is defined as "data about data" (cf. [20]). Numerous metadata models and metadata standards have been proposed in research and industry [14]. These models come from different backgrounds and with different goals. They vary in the domain for which they have been designed and can be domain-specific or general purpose. The existing metadata models also typically focus on a single media type such as image, text, or video. Thus, they are not designed for annotating rich, structured multimedia presentations like SMIL, SVG, and Flash. In addition, the metadata models differ in the complexity of the data structures they provide. With standards like

EXIF [31], XMP [2], and IPTC [27], we find metadata models that provide (typed) key-value pairs to represent metadata of the media type image. Harmonization efforts like the Metadata Working Group [39] are very much appreciated. However, they remain on the same technological level and do not extend their effort beyond the single media type of image. Similar standards exist for other media types, such as ID3 [41] for audio files. Like EXIF, ID3 provides a predefined list of key-value-pairs to annotate audio files and allows for defining custom metadata fields. Yahoo! SearchMonkey Media [61] is an RDF vocabulary providing a set of classes and a set of properties describing media assets. It does not provide any additional functionality compared to EXIF or ID3. However, it covers further media types like video and text. In addition, it allows to create collections of images and videos. Recently, the providers of the three largest web search engines Microsoft, Google, and Yahoo! have agreed on a common microformat called Schema.org [18] to annotate media assets on the web. Like SearchMonkey Media, Schema.org supports annotation of different media types, i.e., audio, image, and video, with technical properties as well as additional properties like editor, genre, and reviews. However, also Schema.org does not consider the semantic description of structured multimedia content. Other metadata models like Dublin Core [13] support hierarchical modeling of key-value pairs. It can be used to describe arbitrary resources. However, it is designed to annotate entire documents and not parts of it. In addition, as it is very generic it covers only a small fraction of the metadata needed to sufficiently annotate rich, structured multimedia content. With MPEG-7 [38], we find a comprehensive metadata standard created in collaboration with different communities [40]. It aims at covering mainly the decomposition and description of low-level features of audiovisual media content and also provides basic means for semantic annotation. MPEG-7 suffers from many redundancies introduced in the standard. For example, there are several ways to annotate image regions in MPEG-7 such as using semantic labels, keyword annotation, and structured annotation [49, 3]. Several approaches have been published providing a formalization of MPEG-7 as an ontology [12], e.g., by Hunter [25, 24], Bloehdorn et al. [6], Garcia and Celma [17], Isaac and Troncy [28], DS-MIRF [51], or the Core Ontology on Multimedia (COMM) [3]. In contrast to other approaches to modeling MPEG-7 as an ontology [25, 6, 17, 28, 51], COMM is not designed as a one-to-one mapping from MPEG-7 to an ontological representation. Instead, COMM aims at resolving the redundancy problems of MPEG-7 by providing a set of ontology design patterns that cover the core and repetitive building blocks of MPEG-7. Although these ontologies provide clear semantics for the multimedia annotations, they still focus on MPEG-7 as the underlying metadata standard.

Annotation of structured multimedia documents is in principle possible with MPEG-7 by considering the multimedia content as a media stream that can be decomposed. However, conducting such a decomposition for a complex structured multimedia presentation is not very practical in MPEG-7 due to the nature of annotations in MPEG-7 and the complexity involved with these annotations. In MPEG-7

a structured multimedia presentation such as a Flash or SMIL presentation is considered an audiovisual media stream. In order to associate MPEG-7 annotations with this stream, the audio-visual stream would be appropriately decomposed and annotated with the MPEG-7 metadata. This way of annotating media content is not enforced by MPEG-7 and could be done differently. However, it is likely that it would be done in this way in order to provide support for annotations similar to the way metadata are associated with Flash or SMIL. For example, the multimedia presentation from Section 2 rendered by the Real Player would be temporally decomposed into some time-variant streams of the first and second part of the presentation. In addition, the streams would be spatially decomposed into the left image, right image, and the textual media assets involved. Each stream would be annotated in MPEG-7 with the appropriate metadata at the time when it is rendered to the users. This approach is not very practicable, as the annotations have to be associated to the content each time the presentation is rendered. However, the multimedia annotations are available a-priori to the rendering of the presentations. Thus, they can already be associated with and stored together with the multimedia content beforehand. This approach is followed by today's presentation formats such as SMIL and SVG.

From the existing metadata standards and metadata models, only few models like the Functional Requirements for Bibliographic Records (FRBR) [26], Conceptual Reference Model (CIDOC CRM) [11], and COMM consider the separation of information objects and information realizations [9]. Examples of information objects are stories, stage plays, or narrative structures. John's presentation, e.g., could be seen as an information object of a narrative structure telling the history of nuclear energy. Each information object is realized by different so-called information realizations [9]. Only a realization brings something abstract such as a message into the real world and makes it perceivable by humans (or any kind of agent). However, it is not fully supported in FRBR as the annotations can only be applied on the information objects. Also COMM does not fully support the separation of information objects and information realizations. In the Semantic Annotation Pattern of COMM, MPEG-7 descriptors cannot be associated with information realizations. All descriptions, including format specific low-level features of the realizations, are attached to the information object. Thus, it is not possible to annotate the different realizations of the same image information object with different metadata. All annotations of the image realizations, also those that are specific to the realization such as resolution, aperture, and others, have to be associated with the information object in COMM. In addition, COMM has no support for collections of information entities and no explicit pattern for provenance information that can be applied to annotations, decompositions, as well as collections like in M3O. The existing metadata models and metadata formats also hardly integrate low-level and high-level features, i.e., the integration of representing both the features that can be extracted from the media assets as well as the annotation with semantic background knowledge. This is unfortunate, as studies have shown the need for semantic anno-

tation [21, 48] and conceptual queries, e.g., in image retrieval [36, 22, 23]. Finally, a multitude of conceptual models for multimedia metadata have been proposed such as [30, 34]. However, they have not been operationalized or published as a standard or ontology.

The list of metadata models and metadata standards is far from being complete and is beyond the scope of this work. The examples discussed have been selected as representative to show the variety of the different metadata models and metadata standards for multimedia documents that exist today. A comprehensive overview of multimedia metadata models and standards can be found in a report [7] of the W3C Multimedia Semantics Incubator Group. Subsequently, the current W3C Media Annotations Working Group has published an ontology for media resources [35] that aims at providing a mapping vocabulary for a number of existing standards, including EXIF, IPTC, ID3, or Yahoo! SearchMonkey Media. However, the technical drawbacks of the existing standards are not tackled by this working group. Nonetheless, the proposed mapping vocabulary is a great source for defining refinements and specific extensions of the M3O [14]. Building on top of this work will provide a core of mappings to many existing standards.

## 8. Conclusions and Future Work

The Multimedia Metadata Ontology (M3O) is a generic modeling framework for rich, structured multimedia documents. It provides different, connected ontology design patterns each providing a specific feature for representing multimedia metadata. We have refactored the initial M3O [44] and extracted a common Provenance Pattern from the Annotation Pattern and Decomposition Pattern. These patterns are enriched by a Collection Pattern describing collections of multimedia documents and media assets. Provenance information can be added to any kind of annotations, decompositions, and collections. We have shown that the patterns of the M3O can be combined with design patterns from other ontologies and combined with existing ontologies such as the W3C Provenance Vocabulary Mappings [58]. In order to make use of the M3O in concrete multimedia applications, we have developed an application programming interface (API) based on our sophisticated persistence layer Winter [45]. The API has been integrated and used with our SemanticMM4U framework [46] for the multi-channel generation of semantically-rich multimedia presentations.

All resources of the M3O are publicly available from its homepage: <http://west.uni-koblenz.de/m3o>. This includes the patterns in OWL, the example presentation, all examples of applying the M3O, mappings to different existing metadata models and metadata standards, and the implementation of our persistence layer Winter, the M3O API, and its use within the SemanticMM4U framework.

## References

- [1] Adobe, Inc. Flash file format, July 2008. <http://www.adobe.com/licensing/developer/>.

## 24 REFERENCES

- [2] Adobe, Inc. XMP Specifications, 2008. <http://www.adobe.com/devnet/xmp/>.
- [3] R. Arndt, R. Troncy, S. Staab, L. Hardman, and M. Vacura. COMM: designing a well-founded multimedia ontology for the web. In *ISWC+ASWC*, pages 30–43, 2007.
- [4] F. Baader, D. Calvanese, D. L. McGuinness, D. Nardi, and P. F. Patel-Schneider, editors. *The Description Logic Handbook*. Cambridge University Press, 2003.
- [5] P. V. Biron and A. Malhotra. XML Schema Part 2: Datatypes Second Edition, W3C Recommendation. October 2004. <http://www.w3.org/TR/xmlschema-2/>.
- [6] S. Bloehdorn, K. Petridis, C. Saathoff, N. Simou, V. Tzouvaras, Y. Avrithis, S. Handschuh, Y. Kompatsiaris, S. Staab, and M. G. Strintzis. Semantic annotation of images and videos for multimedia analysis. In *ESWC*. Springer, May 2005.
- [7] S. Boll, T. Bürger, O. Celma, C. Halaschek-Wiener, E. Mannens, and R. Troncy. Multimedia Vocabularies on the Semantic Web, July 2007. <http://www.w3.org/2005/Incubator/mmsem/XGR-vocabularies/>.
- [8] S. Boll, P. Sandhaus, A. Scherp, and U. Westermann. Semantics, content, and structure of many for the creation of personal photo albums. In *ACM Multimedia*, 2007.
- [9] S. Borgo and C. Masolo. *Handbook on Ontologies*, chapter Foundational choices in DOLCE. Springer, 2nd edition, 2009.
- [10] F. Buschmann, R. Meunier, H. Rohnert, P. Sommerlad, and M. Stal. *Pattern-Oriented Software Architecture, Volume 1: A System of Patterns*. Wiley, 1996.
- [11] CIDOC. The cidoc conceptual reference model, 2011. <http://www.cidoc-crm.org/>.
- [12] S. Dasiopoulou, V. Tzouvaras, I. Kompatsiaris, and M. G. Strintzis. Enquiring MPEG-7 based multimedia ontologies. page 331–370, Oct. 2009.
- [13] Dublin Core Metadata Initiative. DCMI Metadata Terms, Jan. 2008. <http://dublincore.org/documents/dcmi-terms/>.
- [14] D. Eißing, A. Scherp, and C. Saathoff. Integration of existing multimedia metadata formats and metadata standards in the M3O. In *SAMT*. Springer, 2010.
- [15] E. Gamma, R. Helm, R. Johnson, and J. Vlissides. *Design patterns: elements of reusable object-oriented software*. Programmer’s Choice. Addison-Wesley, July 2004.
- [16] A. Gangemi and V. Presutti. *Handbook on Ontologies*, chapter Ontology Design Patterns. Springer, 2009.
- [17] R. Garcia and O. Celma. Semantic integration and retrieval of multimedia metadata. In *Knowledge Markup and Semantic Annotation Workshop*, Galway, Ireland, 2005.
- [18] Google, Microsoft, Yahoo! schema.org, 2011. <http://schema.org/>.
- [19] L. Hardman, Z. Obrenovic, F. Nack, B. Kerhervé, and K. W. Piersol. Canonical processes of semantically annotated media production. *Multimedia Syst.*, 14(6):327–340, 2008.
- [20] B. Haslhofer and W. Klas. A survey of techniques for achieving metadata interoperability. *ACM Comput. Surv.*, 42(2):1–37, 2010. ISSN 0360-0300.
- [21] L. Hollink, G. Nguyen, G. Schreiber, J. Wielemaker, B. Wielinga, and M. Worring. Adding spatial semantics to image annotations. In *Knowledge Markup and Semantic Annotation*, 2004.
- [22] L. Hollink, A. T. Schreiber, B. J. Wielinga, and M. Worring. Classification of user image descriptions. *Int. J. of Human-Computer Studies*, 61(5):601 – 626, 2004.
- [23] L. Hollink, G. Schreiber, and B. Wielinga. Patterns of semantic relations to improve image content search. *Web Semantics*, 5(3):195–203, 2007.
- [24] J. Hunter. Enhancing the semantic interoperability of multimedia through a core ontology. *IEEE Trans. on Circuits and Systems for Video Technology*, 13(1), 2003.
- [25] J. Hunter. *Multimedia Content and the Semantic Web: Standards, Methods and Tools*, chapter Adding Multimedia to the Semantic Web: Building an MPEG-7 ontology, pages 261–283. John Wiley & Sons, 2005. ISBN 0470857536.

- [26] Int. Federation of Library Associations and Institutions. Functional requirements for bibliographic records. Technical report, IFLA, 2009. <http://www.ifla.org/publications/functional-requirements-for-bibliographic-records>.
- [27] International Press Telecommunications Council. “IPTC Core” Schema for XMP Version 1.0 Specification document, 2005. <http://www.iptc.org/>.
- [28] A. Isaac and R. Troncy. Designing and using an audio-visual description core ontology. In *Workshop on Core Ontologies in Ontology Engineering, CEUR Proceedings Vol. 118*, Whittlebury Hall, Northamptonshire, UK, October 2004.
- [29] ISO/IEC. Lightweight Application Scene Representation (LASER) and Simple Aggregation Format (SAF), Apr. 2007. [http://www.mpeg-laser.org/documents/DRAFT-LASER\\_2ND\\_ED.pdf](http://www.mpeg-laser.org/documents/DRAFT-LASER_2ND_ED.pdf).
- [30] A. Jaimes and S.-F. Chang. A conceptual framework for indexing visual information at multiple levels. In *IS&T/SPIE Internet Imaging*, volume 3964, 2000.
- [31] JEITA. Exchangeable image file format for digital still cameras, April 2002. <http://www.jeita.or.jp/>.
- [32] M. Kanzaki. Exif data description vocabulary, 2003. <http://www.kanzaki.com/ns/exif>, last update in 2007.
- [33] M. Kim, S. Wood, and L.-T. Cheok. Extensible MPEG-4 textual format (XMT). In *ACM workshops on Multimedia*, pages 71–74, New York, NY, USA, 2000. ACM.
- [34] S. S. Layne. Some issues in the indexing of images. *Journal of the American Society for Information Science*, 45(8):583–588, 1994.
- [35] W. Lee, T. Bürger, F. Sasaki, V. Malaisé, F. Stegmaier, and J. Söderberg. Ontology for Media Resource 1.0. W3C Media Annotations Working Group, Working Draft, June 2009. <http://www.w3.org/TR/2009/WD-mediaont-10-20090618/>.
- [36] M. Markkula and E. Sormunen. End-user searching challenges indexing practices in the digital newspaper photo archive. *Information Retrieval*, 1(4):259–285, 2000.
- [37] Merriam-Webster, Inc. Metadata, 2009. <http://www.m-w.com/dictionary/metadata>.
- [38] MPEG-7. Multimedia content description interface. Technical report, Standard No. ISO/IEC n15938, 2001.
- [39] MWG. Metadata working group, 2010. <http://www.metadataworkinggroup.org/>.
- [40] F. Nack and A. Lindsay. Everything you wanted to know about MPEG-7: Part 1. *IEEE MultiMedia*, 6(3):65–77, 1999.
- [41] M. Nilsson and M. Mutschler. ID3, 2009. <http://www.id3.org/>.
- [42] E. Oren, R. Delbru, S. Gerke, A. Haller, and S. Decker. ActiveRDF: object-oriented semantic web programming. In *WWW*. ACM, 2007.
- [43] P. Pringle, M. Starr, and W. McCavitt. *Electronic Media Management*. Focal, 1999.
- [44] C. Saathoff and A. Scherp. Unlocking the semantics of multimedia presentations in the web with the multimedia metadata ontology. In *WWW*. ACM, 2010.
- [45] C. Saathoff, S. Scheglmann, and S. Schenk. Winter : Mapping RDF to POJOs revisited. In *Poster & Demo Session; ESWC*, Heraklion, Greece, 2009.
- [46] A. Scherp. Canonical processes for creating personalized semantically rich multimedia presentations. *Multimedia Syst.*, 14(6):415–425, 2008.
- [47] A. Scherp, T. Franz, C. Saathoff, and S. Staab. F—A Model of Events based on the Foundational Ontology DOLCE+ Ultralight. In *Knowledge Capturing*, 9 2009.
- [48] G. Schreiber, I. Blok, D. Carlier, W. van Gent, J. Hokstam, and U. Roos. A mini-experiment in semantic annotation. pages 404–408, 2002.
- [49] S. Staab, A. Scherp, R. Arndt, R. Troncy, M. Grzegorzec, C. Saathoff, S. Schenk, and L. Hardman. Semantic multimedia. In *Reasoning Web*. Springer, 2008.
- [50] R. Steinmetz and K. Nahrstedt. *Multimedia Systems*. Springer, 2004.

26 REFERENCES

- [51] C. Tsinaraki, P. Polydoros, and S. Christodoulakis. Integration of OWL ontologies in MPEG-7 and TVAnytime compliant semantic indexing. *Advanced Information Systems Engineering*, pages 398–413, 2004.
- [52] D. Van Deursen, R. Troncy, E. Mannens, S. Pfeiffer, Y. Lafon, and R. Van de Walle. Implementing the media fragments URI specification. In *WWW*. ACM, 2010.
- [53] W3C. Web ontology language, January 2004. <http://www.w3.org/TR/owl-features/>.
- [54] W3C. RDF Primer, Feb. 2004. <http://www.w3.org/TR/REC-rdf-syntax/>.
- [55] W3C. Basic geo vocabulary, 2006. <http://www.w3.org/2003/01/geo/>.
- [56] W3C. SMIL 3.0, Dec. 2008. <http://www.w3.org/TR/SMIL/>.
- [57] W3C. SVG, Apr. 2009. <http://www.w3.org/TR/SVG/>.
- [58] W3C. Provenance vocabulary mappings, 2011. [http://www.w3.org/2005/Incubator/prov/wiki/Provenance\\_Vocabulary\\_Mappings](http://www.w3.org/2005/Incubator/prov/wiki/Provenance_Vocabulary_Mappings).
- [59] W3C. XML schema, July 2011. <http://www.w3.org/TR/xmlschema11-2/>.
- [60] W3C. HTML5, 2011. <http://www.w3.org/TR/html5/>.
- [61] Yahoo! SearchMonkey Media, 2009. <http://developer.yahoo.com/searchmonkey/smguidesearchmonkey-media.html>.