Mapping Audiovisual Metadata Formats Using Formal Semantics

Martin Höffernig, Werner Bailer, Günter Nagler, and Helmut Mülner

JOANNEUM RESEARCH Forschungsgesellschaft mbH
DIGITAL – Institute of Information and Communication Technologies
Steyrergasse 17, 8010 Graz, Austria
{firstname.lastname}@joanneum.at

Abstract. Audiovisual archives hold enormous amounts of valuable content. However, many of these archives are difficult to access, as their holdings are documented using a range of different metadata formats. Being able to exchange metadata is the key to ensuring access to these collections and to establish interoperability among audiovisual collections, between audiovisual collections and other cultural heritage institutions as well as portals such as Europeana. In this work we attempt to model mappings between metadata formats based on a high-level intermediate concept representation in order to avoid hand-crafted one-to-one mappings between metadata formats. In addition, we define mapping templates on data type level, from which the code for mapping instructions between a pair of formats can be derived. The high-level intermediate concept representation is based on the existing meon ontology and the resulting mapping instructions are expressed using XSLT. As a proof of concept, mappings between two metadata formats are formalized and integrated in a web based prototype application.

1 Introduction

There are millions of hours of audiovisual content in collections of dedicated audiovisual archives, other archives, libraries and museums. Preserving digital audiovisual content means keeping it accessible and usable for a long time, including the metadata describing it. This requires migrating metadata to models currently in use and enabling metadata mapping to formats used in the systems which need to retrieve, deliver, display and process archived content. Being able to exchange metadata is the key to ensuring access to audiovisual collections, establishing interoperability among audiovisual collections, and between audiovisual collections and other cultural heritage institutions. Metadata exchange is often hindered by the diversity of metadata formats and standards that exist in the media production process and in different communities.

The work described in this paper builds on an existing multimedia ontology used for mapping metadata concepts in audiovisual media production workflows. The contributions of this paper are the extension of this ontology to include format-specific concepts and the support for data type conversions in the mapping system.

The rest of this paper is organized as follows. The remainder of this section discusses the motivation for developing a novel mapping approach and analyzes the problem and the resulting requirements. Section 2 discusses related work. In Section 3 we propose our approach for mapping and demonstrate a prototype implementation in Section 4. Section 5 concludes the paper.

1.1 Motivation

There is a number of use cases in the audiovisual archive domain that require metadata mapping. On one side these use cases include in-house scenarios such as the conversion of legacy technical metadata as part of preservation processes, access to legacy content descriptions in order to map between material and editorial entities, or the extraction metadata embedded in file headers and convert it to the data structures needed for import into a preservation system. On the other side there are use cases across institutions, such as content exchange with other cultural heritage institutions, content provision to access portals (e.g. Europeana[1]) and outsourcing of annotation and access services, with potentially different data models between the archive's and service provider's infrastructure.

Tools for metadata mapping are needed to overcome the existing interoperability issues on both syntactic and semantic level. However, with n formats existing in a given environment, we need in the worst case $O(n^2)$ mappings if we go for a simple approach considering only pair-wise mappings. Chaining mappings is also not a useful approach, as transitivity of relations cannot be ensured due to the incompleteness of mappings. We thus propose an approach that uses a high-level intermediate representation, together with mapping templates on data type level, from which the code for a mapping problem between a pair of standards can be derived. This would ideally allow us to solve the problem with $O(2n)$ definitions.

1.2 Mapping Problem and Resulting Requirements

The metadata formats encountered across the audiovisual archive domain differ in various aspects.

Coverage MPEG-7 [10] for example aims to be domain independent while TV Anytime [7] focuses on broadcast metadata for consumers.

Comprehensiveness For example, MPEG-7 aims to provide comprehensive descriptions of multimedia content ranging from low-level features that can be extracted automatically to fine-grained semantic description of a scene, while the Dublin Core Element Set[2] provides a simple list of general annotation elements.

Complexity Metadata formats also differ in the complexity of their description syntax. For example, the Dublin Core `dc:creator` element is a simple name

[1] `http://www.europeana.eu`

[2] `http://dublincore.org/documents/dces/`

or an URI identifying an agent whereas the creator's name in MPEG-7 is divided into a complex nested structure of titles, family names and given names along with the definition of her role in the creation process.

The problem of metadata interoperability exists on two levels:

Syntactic interoperability Even if metadata is represented and processed in the same syntactic format, today typically some XML-based format, syntactic interoperability issues concerning e.g. data type representations and different structural granularity still exist. Note that this does not imply that all metadata formats need to be defined as XML documents, only that they can be rendered as such (with services or wrappers).

Semantic interoperability Metadata elements in different formats are often not fully congruent w.r.t. to their semantics, but are defined to be narrower or wider than the related elements in other formats. In addition, some formats leave room for interpretation, so that different organization will use the elements in a different way. Also, controlled vocabularies can be used for the values of several elements, thus mapping requires alignment of (possibly in-house) vocabularies.

We define the following requirements for our mapping tool. Given an input metadata document in some supported, XML represented format the mapping tool must provide a best practice mapping of the metadata document to another supported, XML represented target format. The target document must be valid w.r.t. to the standards and must convey the semantics of the input document as far as possible. For each supported format, it is assumed that the mapping tool has access to the following definitions:

- A formal description of the set of system-wide supported metadata elements and their relations. This is independent of a format.
- A formal description of the set of metadata elements in the format and their relations among them (if applicable) and their relations to the set of elements supported by the system.
- A set of mapping rules to be applied.

The mapping tools should be deployed as services in order to allow integration into a wide range of components (which may be located remotely).

There are a number of foreseen difficulties and limitations:

- For some pairs of formats, mapping will be incomplete and lossy in one direction, and will have only loosely defined semantics (or need appropriate additional rules justified by the use case) in the opposite direction. This also means, that bidirectional mapping for such a pair of standards is only possible in a very limited way.
- There are ambiguities in mappings that need default or use case specific rules to be resolved.
- If formats use specific data types, their mappings to (structures of) standard data types needs to be defined. This will reduce the saving in the number of mapping definitions needed.

2 Related Work

There are different approaches described in literature for the problem of mapping between different metadata models, mostly focusing on XML-based representations. Xing et al. [12] present a system for automating the transformation of XML documents using a tree matching approach. However, this method has an important restriction: the leaf text in the different documents has to be exactly identical. This is hardly the case when combining different metadata standards. Likewise, Yang et al. [13] propose to integrate XML Schemas. They use a more semantic approach, using the ORA-SS data model to represent the information available in the XML Schemas and to provide mappings between the different documents. The ORA-SS data model allows to define objects and attributes to represent hierarchical data, however, more advanced mappings involving semantic relationships cannot be represented.

Cruz et al. [2] introduced an ontology-based framework for XML semantic integration. For each XML source integrated, a local RDFS ontology is created and merged in a global ontology. During this mapping, a table is created that is further used to translate queries over the RDF data of the global ontology to queries over the XML original sources. The authors assume that every concept in the local ontologies is mapped to a concept in the global ontology. This assumption can be hard to maintain when the number and the degree of complexity of the incorporated ontologies increases. Poppe et al. [11] advocate a similar approach to deal with interoperability problems in content management systems. An OWL upper ontology is created and the different XML-based metadata formats are represented as OWL ontologies and mapped to the upper ontology using OWL constructs and rules. However, the upper ontology is dedicated to content management system and, as such, is not as general as the approach proposed in this paper.

In [9] the authors propose a format for describing mappings between data models in the domain of database schema conversion. They define a very simple format expressed as a XML DTD which can contain mapping definitions. They suggest a graphical tool to make the process more intuitive. Of course mapping database schemas is a subset of the problem of mapping XML schemas. The approach covers a set of common heterogeneity issues encountered during mapping, to which extensions and exceptions can be defined when needed. The common cases covered by the mapping language are introduction of intermediate nodes, contraction of compounds, transformation from parallel to nested structures (including a specific case with an intermediate node), using the same instance in multiple mappings and conditional mappings.

The problem of mapping between different multimedia metadata formats is also encountered in different standardization initiatives. JPSearch[3] is a project issued by the JPEG standardization committee to develop technologies that enable search and retrieval capabilities among image archives, consisting of five parts. Part 2 introduces an XML based core metadata schema and transforma-

[3] `http://www.jpsearch.org`

tion rules for mapping descriptive information (e.g., core metadata to MPEG-7 or core metadata to Dublin Core) between peers [4], and part 3 adapts a profile of the MPEG Query Format [5] for ensuring standardized querying. The focus of this approach is clearly querying, thus the approach is concerned with mapping source format to a common representation, but not to map between an arbitrary pair of formats.

The W3C Media Annotations Working Group (MAWG)[4] has the goal of improving the interoperability between multimedia metadata schemas. The proposed approach is to provide an interlingua ontology and an API designed to facilitate cross-community data integration of information related to media resources in the web, such as video, audio, and images. The set of core properties that constitute the Media Ontology 1.0 is based on a list of the most commonly used annotation properties from media metadata schemas currently in use. Like with MPEG Query Format, the aim of mapping is in this case the Media Ontology and mapping between a pair of formats is not supported.

Finally, there are several initiatives in the cultural heritage domain to provide services for mapping between library and museum metadata formats. The Vocabulary mapping framework[5] is a project connected with JISC[6]. It is an extension of the RDA/ONIX framework, which was developed in 2006 for libraries and the publishing industry as a framework for categorizing resources. It facilitates the transfer and use of resource description data.

The OCLC crosswalk service[7] is part of the Metadata Schema Transformation project, which tries to support library staff who often have to translate between different metadata standards. The project designed a self-contained crosswalk utility that can be called by any application that has to translate (supported) metadata records. The translation logic is executed by a dedicated XML application called the Semantic Equivalence Expression Language (SEEL), a language specification and a corresponding interpreter that is used to execute the transformations. The system is closed source and the publicly available information is really sparse.

The Athena mapping service[8] is an online service to convert in different XML formats (e.g. museum data) to the LIDO and to the Europeana format. Users can create a document collection on an Athena server by uploading single documents or sets of documents The service infers the schema from the presented documents. The definition of a mapping happens in two steps: (i) Identifying "items" in a tree view of a document and naming them, and (ii) define a mapping between the items and the target schema elements. Both steps are done by using drag and drop. Mapping possibilities include mapping structural elements, by repeating input elements, constant value assignment, a concatenate function and conditional mapping. When a mapping definition is finished, the transfor-

[4] http://www.w3.org/2008/WebVideo/Annotations

[5] http://cdlr.strath.ac.uk/VMF/background.htm

[6] http://www.jisc.ac.uk

[7] http://xwalkdemo.oclc.org/

[8] http://www.athenaeurope.org

mation of the input documents can be initiated. The service currently supports concatenating values, but not splitting. Management and mapping of controlled vocabularies is supported.

3 Approach

Our approach uses a high-level intermediate representation of metadata elements. Metadata elements from a specific metadata format are described in relation to these generic elements. In addition, mapping templates on data type level are used. From these sources the code for a mapping problem between a pair of formats can be derived. We use an existing ontology, which has been extended for our purpose, to represent the formal semantics of the high-level intermediate representation. Since mapping instructions are derived from the ontology description, these mapping instructions are easier to maintain than hard coded mapping instructions. Furthermore the high-level intermediate representation serves as a hub for mapping between formats. Therefore hand-crafted one-to-one mappings between each pair of metadata formats are avoided, the mappings can be created automatically. Adding a new metadata format is done without side effects to existing definitions.

The core of this approach is the *meon* ontology[9] [8] which describes generic metadata elements and the relations between them. *meon* was originally developed to model metadata elements used throughout the audiovisual media production workflow in a format independent way in order to support content exchange and automation. Since concepts and relations coming from the *meon* ontology are reused and extended in our approach, the relations between metadata elements or groups of metadata elements used in this ontology are briefly introduced.

Definition relation A metadata element (or group of elements) A defines another metadata element (or group of elements) B, if B can be derived without any semantic ambiguity from A by some mapping/conversion.

Equivalence relation A metadata element (or group of elements) B is equivalent to another metadata element (or group of elements) B, if A defines B and B defines A.

Subtype relation is used to describe that a metadata element is a more specific one of another metadata element.

In order the express these relations, the *meon* ontology, expressed in OWL-DL [3] models the classes `meon:Concept`, with subclasses `meon:AtomicConcept` and `meon:CompoundConcept` and three data type properties, namely `meon:contains`, `meon:defines` and `meon:subTypeOf`. The class `Concept` models the general concept represented by a metadata element in the ontology. Class `AtomicConcept` represents all single metadata elements, while class `CompoundConcept` describes groups of metadata elements containing at least two metadata elements. Property `contains` describes that an instance of class `AtomicConcept` is

[9] prefix `meon`: `http://www.20203dmedia.eu/meon#`

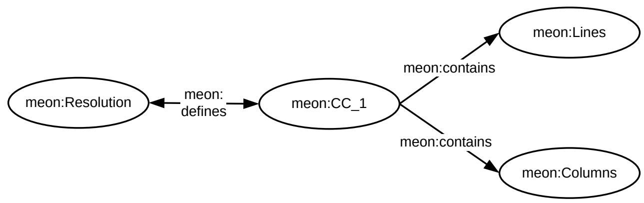

Fig. 1. *meon* example.

part of a group of metadata elements (i.e. part of an instance of class `Compound-Concept`). In order to describe that a metadata element (or group of elements) can be derived from another metadata element (or group of elements) by some mapping/conversion, the property `defines` is used. This property is used between instances of class `Concept`. Since the equivalence relation is just the result of a bidirectional definition relation, an appropriate use of this property is expressive enough to model also the equivalence relation. The property `subTypeOf` defines a relation between two `Concept` instances, of which one is a specialization of the other.

An example of the *meon* ontology for describing metadata elements and their relations between is shown in Figure 1. In this example *meon* is used to describe that the metadata element resolution is equivalent to the group of metadata elements containing lines and columns. Therefore resolution, lines, and columns are modeled as instances of class `AtomicConcept`.

`CC_1`, which is an instance of class `CompoundConcept`, represents the group of metadata elements containing lines and columns (denoted by the `contains` relation). To model the equivalence relation, two `defines` relations between `meon:Resolution` and `CC_1` are used.

The *meon* ontology has been extended to express mapping relations between metadata formats. In addition to the ontology of generic metadata concepts, specific ontologies are created for each format. They follow the same pattern as the *meon* ontology and include `meon:defines` relations between the format specific and the generic concepts. Figure 3 shows an example for formalizing the mapping relations of metadata elements expressed in MPEG-7 and Dublin Core via generic *meon* concepts, in this case `meon:creator`, `meon:producer`, `meon:contributor`, `meon:performer`, `meon:resolution`, `meon:lines` and `meon:columns`. It also models their relations, i.e. `meon:producer` is a subtype of `meon:creator`, `meon:performer` is a subtype of `meon:contributor`, and the compound of `meon:lines` and `meon:columns` is equivalent to `meon:resolution`. In the same manner the format specific concepts are defined. In MPEG-7, the creator element is used in combination with a role classification scheme to define types of creators and contributors. Therefore any specialized type of producer or contributor according to the used classification scheme is modeled as an subtype of class `mpeg7:Creator`. In our case, `mpeg7:Producer` and `mpeg7:Performer` are subtypes of `mpeg7:Creator`. In addition, another subtype mpeg7:UnqualifiedCreator is used to describe subtypes of mpeg7:Creator which are not explicitly modeled in the ontology. For describing the resolution in MPEG-7, mpeg7:Width and mpeg7:Height are added to the ontology. Furthermore the Dublin Core concepts dc:Creator, dc:Contributor and the DC Terms concept dcterms:Extent are added.

Definition relations are introduced to link these concepts to the *meon* concepts. dc:Creator is aligned with meon:Creator and dc:Contributor with meon:Contributor. Similarly, mpeg7:Producer is equivalent to meon:Producer and mpeg7:Performer is equivalent to meon:Performer. Furthermore there exists a equivalence relation between mpeg7:UnqualifiedCreator and meon:Creator, while there is only a definition relation between meon:Contributor and mpeg7:UnqualifiedCreator. In addition mpeg7:Height (resp. mpeg7:Width) is identical to meon:Lines (resp. meon:Columns) and dcterms:Extent is also equivalent to meon:Resolution.

Now it is possible to infer how concepts from different metadata formats are related by observing the relations among generic concepts and to the format specific concepts. In our example, it can be derived that mpeg7:Producer defines dc:Creator and mpeg7:Performer defines dc:Contributor. Since dc:Creator is the top-level term for all kind of creators, it can only define the mpeg7:UnqualifiedCreator. The same applies to dc:Contributor. This simple example thus also demonstrates typical limitations, that can only be overcome with additional, possible application specific information. Due to the lack of role qualification in Dublin Core, only a very coarse mapping to MPEG-7 is possible. If the classification scheme used by the MPEG-7 document is known, the situation can be improved. Then it is at least possible to decide whether an MPEG-7 role maps to creator or contributor, and in the other direction, a contributor could be mapped to a contributor term in the classification scheme, if that exists.

In order to retrieve mapping instructions between formats, format specific concepts are extended to carry additional information needed for mapping. Therefore the XPath location [1] and information about the data type representation of the metadata element are added to the ontology. For this purpose a new class meon:DataTypeRepresentation and new properties meon:hasDataTypeRepresentation, meon:hasXPath and meon:hasDataTypeFormat are introduced. This information is used to locate elements in the input and output documents and to apply the appropriate data type conversion. In Figure 4 the data type representations for the concepts mpeg7:Producer and dc:Creator are shown as an example. There can be several data type representations for a concept, typically associated with different XPath expressions. For example, the mpeg7:Creator element could contain the name of a person or organization, indicated by the value of its mpeg7:Agent/@xsi:type attribute.

We propose to use of XSL templates [6] for representing the data type conversion instructions. We define a table of such templates, which are annotated with their input and output data type formats. Once the inference is done, the input and output data types of the elements in the respective source and target formats are known and the appropriate template is found by looking up in the

```
<xsl:template  name="Mpeg7_PersonType__TaggedText">
  <xsl:param name="tag">xxx</xsl:param>
  <xsl:for-each select="mpeg7:Name">
    <xsl:element name="{$tag}">
      <xsl:call-template name="Mpeg7_Name__SimpleText"/>
    </xsl:element>
  </xsl:for-each>
</xsl:template>

<xsl:template name="Mpeg7_Name__SimpleText">
  <xsl:for-each select="mpeg7:GivenName">
    <xsl:value-of select="."/>
    <xsl:text> </xsl:text>
  </xsl:for-each>
  <xsl:for-each select="mpeg7:FamilyName">
    <xsl:value-of select="."/>
    <xsl:text> </xsl:text>
  </xsl:for-each>
</xsl:template>
```

Fig. 2. Example of XSL template for transforming the name of a structured MPEG-7 person type into a string.

XSLT	Input data type format	Output data type format
T1	XML_Structured_GivenName_FamilyName	String_FirstName_LastName
T2	XML_Structured_Organization_Name	String_Organization_Name
T3	XML_Attributes_Width_Height	String_Width_Height

Table 1. Example excerpt of data type template table.

table. An example excerpt of such a table is shown in Table 1. Additionally, the required XSL templates to transform the name of a structured MPEG-7 person type into a simple string are depicted in Figure 2.

In our example, it can be derived that `dcterms:Extent` defines both `mpeg7:Height` and `mpeg7:Width`, and also that the compound of `mpeg7:Height` and `mpeg7:Width` defines `dcterms:Extent`. However, we need to apply a single template (T3 in Table 1) that concatenates the values of the width and height attributes in the MPEG-7 documents to the string value of `dcterms:Extent`. This template is associated with both the `mpeg7:Width` and `mpeg7:Height` concepts. The inference process yields a new compound concept containing these two concepts which defines the desired output (`dcterms:Extent`) via `meon:Resolution` (cf. Figure 3). We then look for an appropriate data type template that is associated with all the concepts contained in the inferred compound concept. This data type template is applied to convert the group of source elements (`mpeg7:Height` and `mpeg7:Width`) to the target element (`dcterms:Extent`).

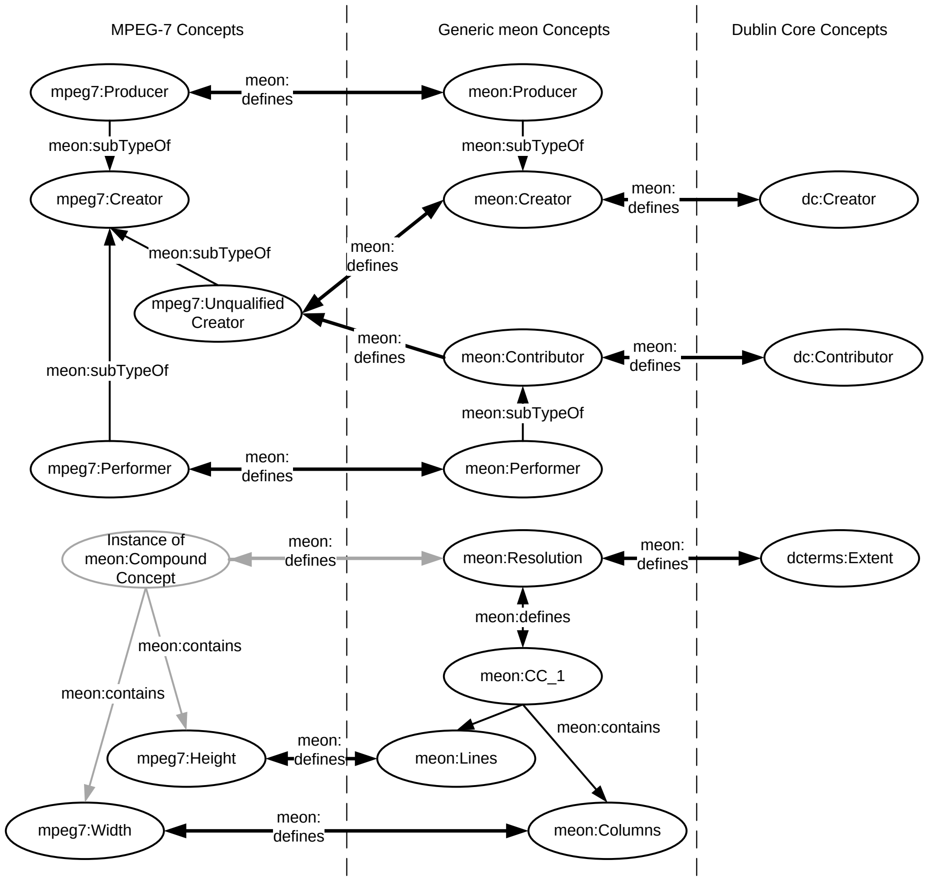

Fig. 3. Examples of mapping elements between MPEG-7 and Dublin Core.

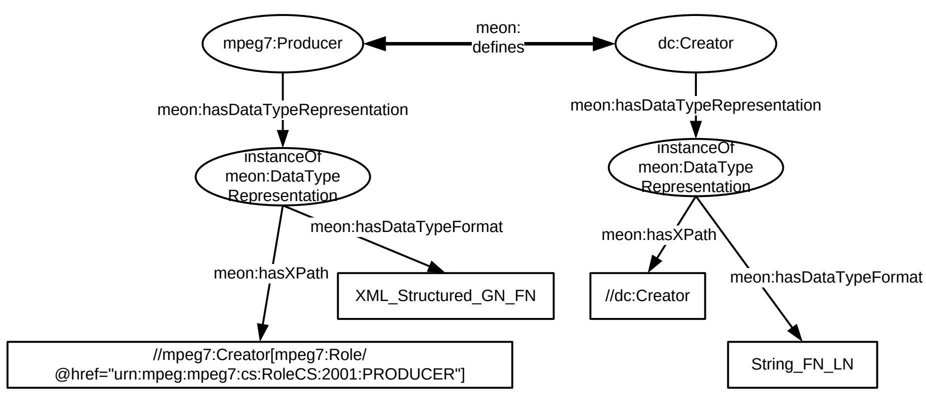

Fig. 4. Data type representation in the ontology.

4 Prototype Application

Following the mapping approach described above, the current prototype implementation of the mapping service is based on an ontology describing how metadata concepts coming from different metadata formats are related. This formalization is used to infer concrete mapping instructions between a pair of metadata formats. While the ontology is expressed in OWL-DL, the mapping instructions are encoded as XSL transformations [6] in order to perform the mapping of an instance document. The workflow of the current prototype implementation is as follows. The metadata concepts and mapping relations in the ontology are annotated with additional information needed for mapping, i.e. format specific metadata with the corresponding XPath information and information on data type representation. Using reasoning techniques over the ontology it is now possible to infer which data type templates are needed for the mapping between different format specific concepts. A SPARQL[10] query is used to retrieve all necessary mapping information from the ontology. Then the next step is to generate a XSL template out of the mapping information. The XSL template document with mapping instructions for all input elements can be applied to a input document instance.

Figures 5 and 6 show the MPEG-7 and Dublin Core XML fragments corresponding to the example described above. As we have discussed earlier, mapping of the MPEG-7 fragment in Figure 5 yields the Dublin Core fragment in Figure 6, but not vice versa. Without further knowledge, both `dc:Creator` and `dc:Contributor` would map to `mpeg7:UnqualifiedCreator`.

The metadata semantic converter application (Figure 7) allows the user to choose an example data set in MPEG-7 XML format and to convert it to Dublin Core XML format. The application displays the conversion steps together with the XML data involved. The application is written in Java using the GWT[11]. The conversion steps are deployed as asynchronous web services on Apache Tomcat 6.0[12]. The server application uses the Jena[13] library for reading and writing RDF and OWL files. The XSLT transformation is done using the Xalan[14] library. The prototype application is available at `http://prestoprime.joanneum.at/`.

5 Conclusion and Future Work

We have presented an approach for mapping between multimedia metadata formats using an ontology, that represents high-level format independent metadata concepts, as well as format specific concepts and their relations to the high-level concepts, together with data type templates. By using an intermediate format independent representation, the approach attempts to avoid pair-wise mappings

[10] `http://www.w3.org/TR/rdf-sparql-query/`
[11] Google Web Toolkit, `http://code.google.com/webtoolkit/overview.html`
[12] `http://tomcat.apache.org/index.html`
[13] `http://jena.sourceforge.net/`
[14] `http://xml.apache.org/xalan-j/`

between formats, thus breaking the need for a quadratic number of mappings. We have implemented mapping between MPEG-7 and Dublin Core in a prototype application. This example demonstrates the feasibility of the approach but also shows limitations due to limited expressiveness of formats.

The proposed approach can support complete mappings between two metadata formats as well as partial mappings in case of formats that do not sufficiently overlap. Partial mappings occur in cases where the *meon* ontology does not include appropriate concepts for format specific properties or mappings are incomplete due to limitations of the expressiveness of the formats involved. This also means that adding new formats does not require changing the *meon* ontology, but only adding concepts and relations if necessary. In cases where a property is specific to a format, there is no need to add it to *meon*, as it will not be possible to map it to any other format. In cases where a property is more general, it can be added to *meon*, and relation with existing concepts can be defined. Note that this does not require any modifications in the format specific ontologies for exisiting formats nor in exiting mapping templates.

The definition of the data types formats via long unique names is a temporary solution. In the next version, we will integrate the description of the data types and their related properties (e.g., the specifics of a person name string) in the ontology to have this information available for machine processing.

It has been discussed in Section 1.2 that mappings between formats with different expressivity or incongruent semantics can be ambiguous or incomplete. In a practical application some of these limitations can be overcome by defining rules based on additional context knowledge, such as about vocabularies (e.g., MPEG-7 classification schemes) used in the documents used as values or qualifiers of elements, or knowledge about the specific flavour of a format used. Such rules can help to reduce the potential information loss in the mapping process. In order to support users in the definition of such rules, a tool for defining theses rules will be necessary.

One further issue that needs to be addressed in future works are mappings between formats that differ w.r.t. the structure of the description, e.g. mapping from a flat to a hierarchical scheme. Like with incongruent semantics, there might be cases where such issues can only be resolved unambiguously with addition application specific knowledge, especially when mapping from simpler to more complex structures.

Acknowledgments

The research leading to this paper was partially supported by the European Commission under the ICT contract FP7 231161, "PrestoPRIME" (`http://www.prestoprime.eu`).

References

1. James Clark and Steve DeRose (eds.). XML Path Language (XPath) Version 1.0. W3C Recommendation, 1999. `http://www.w3.org/TR/xpath/`.

2. Isabel Cruz, Huiyong Xiao, and Feihong Hsu. An Ontology-Based Framework for XML Semantic Integration. In *International Database Engineering and Applications Symposium*, pages 217–226, Coimbra, Portugal, 2004.
3. Mike Dean and Guus Schreiber. OWL Web Ontology Language: Reference. W3C Recommendation, 10 February 2004. `http://www.w3.org/TR/owl-ref/`.
4. Mario Döller, Florian Stegmaier, Harald Kosch, Ruben Tous, and Jaime Delgado. Standardized Interoperable Image Retrieval. In *ACM Symposium on Applied Computing (SAC), Track on Advances in Spatial and Image-based Information Systems (ASIIS)*, 2010.
5. Mario Döller, Ruben Tous, Matthias Gruhne, Kyoungro Yoon, Masanori Sano, and Ian Burnett. The MPEG Query Format: On the way to unify the access to Multimedia Retrieval Systems. *IEEE Multimedia*, 15(4):82–95, 2008.
6. Michael Kay (ed.). XSL Transformations (XSLT) Version 2.0. W3C Recommendation, 2007. `http://www.w3.org/TR/xslt20/`.
7. ETSI. ETSI TS 102 822-3-1 V1.5.1: Broadcast and On-line Services: Search, select, and rightful use of content on personal storage systems ("TV-Anytime") - Part 3: Metadata. `http://www.etsi.org`, May 2009.
8. Martin Höffernig and Werner Bailer. Formal Metadata Semantics for Interoperability in the Audiovisual Media Production Process. In *Workshop on Semantic Multimedia Database Technologies (SeMuDaTe 2009), co-located with the 4th International Conference on Semantic and Digital Media Technologies (SAMT2009)*, Graz, Austria, 2009, to appear.
9. Haridimos Kondylakis, Martin Doerr, and Dimitris Plexousakis. Mapping language for information integration. Technical Report TR385, Institute of Computer Science, FORTH-ICS, 2006.
10. MPEG-7. Multimedia Content Description Interface. ISO/IEC 15938, 2001.
11. Chris Poppe, Gaëtan Martens, Erik Mannens, and Rik Van de Walle. Personal Content Management System: A Semantic Approach. *Journal of Visual Communication and Image Representation*, 20(2):131–144, 2009.
12. Guangming Xing, Zhonghang Xia, and Andrew Ernest. Building automatic mapping between XML documents using approximate tree matching. In *ACM Symposium on Applied Computing (SAC)*, pages 525–526, Seoul, Korea, 2007.
13. Xia Yang, MongLi Lee, Tok Wang Ling, Lee Tok, and Wang Ling. Resolving Structural Conflicts in the Integration of XML Schemas: A Semantic Approach. In *International Conference on Conceptual Modeling (ER)*, pages 520–533, Chicago, USA, 2003.

```
...
<Creator>
  <Role href="urn:mpeg:mpeg7:cs:RoleCS:2001:PERFORMER">
    <Name>Performer</Name>
  </Role>
  <Agent xsi:type="PersonType">
    <Name>
      <GivenName>Mick</GivenName>
      <FamilyName>Jagger</FamilyName>
    </Name>
  </Agent>
</Creator>
...
<Creator>
  <Role href="urn:mpeg:mpeg7:cs:RoleCS:2001:PRODUCER">
    <Name>Producer</Name>
  </Role>
  <Agent xsi:type="OrganizationType">
    <Name>Acme Inc.</Name>
  </Agent>
</Creator>
...
<Creator>
  <Role href="urn:mpeg:mpeg7:cs:RoleCS:2001:">
    <Name>Creator</Name>
  </Role>
  <Agent xsi:type="PersonType">
    <Name>
      <GivenName>Yoshiaki</GivenName>
      <FamilyName>Shibata</FamilyName>
    </Name>
  </Agent>
</Creator>
...
<VisualCoding>
  <Frame height="640" width="480"/>
</VisualCoding>
```

Fig. 5. Fragments from an MPEG-7 example document.

```
...
<dc:contributor>Mick Jagger</dc:contributor>
<dc:creator>Acme Inc.</dc:creator>
<dc:creator>Yoshiaki Shibata</dc:creator>
<dcterms:Extent>640x480 px</dcterms:Extent>
...
```

Fig. 6. Dublin Core example fragment.

Fig. 7. Screenshot of the metadata converter prototype application.