Journal of Digital Information, Vol 1, No 8 (2001)

MetaNet - A Metadata Term Thesaurus to Enable Semantic Interoperability Between Metadata Domains

Jane Hunter
DSTC Pty Ltd, University of Queensland, Qld, 4072, Australia
Email: jane@dstc.edu.au Web: home page

Abstract

Metadata interoperability is a fundamental requirement for access to information within networked knowledge organization systems. The Harmony international digital library project [1] has developed a common underlying data model (the ABC model) to enable the scalable mapping of metadata descriptions across domains and media types. The ABC model [2] provides a set of basic building blocks for metadata modeling and recognizes the importance of 'events' to describe unambiguously metadata for objects with a complex history. To test and evaluate the interoperability capabilities of this model, we applied it to some real multimedia examples and analysed the results of mapping from the ABC model to various different metadata domains using XSLT [3]. This work revealed serious limitations in the ability of XSLT to support flexible dynamic semantic mapping. To overcome this, we developed MetaNet [4], a metadata term thesaurus which provides the additional semantic knowledge that is non-existent within declarative XML-encoded metadata descriptions. This paper describes MetaNet, its RDF Schema [5] representation and a hybrid mapping approach which combines the structural and syntactic mapping capabilities of XSLT with the semantic knowledge of MetaNet, to enable flexible and dynamic mapping among metadata standards.

1 Introduction

Networked knowledge organisation systems typically contain objects of mixed media types which are described using a multitude of diverse metadata schemas. Hence machine understanding of metadata descriptions which conform to schemas from different domains is a fundamental requirement for access to information within networked knowledge organization systems. In particular, there are three main scenarios in which interoperability among metadata descriptions is required:

- To enable a single search interface across heterogeneous metadata descriptions;
- To enable the integration or merging of descriptions which are based on complementary but possibly overlapping metadata schemas or standards;
- To enable different views of the one underlying and complete metadata description, depending on the user's particular interest, perpective or requirements.

Metadata descriptions from different domains are not semantically distinct but overlap and relate to each other in complex ways. Achieving interoperability between such metadata descriptions via manually-generated one-to-one crosswalks [6] is useful, but this approach does not scale to the many metadata vocabularies that will develop. A more scalable and cost-effective approach is to exploit the fact that many entities and relationships - for example, people, places, creations, organisations, events, etc. - occur across all of the domains. The Harmony project [1] has been investigating this more general approach towards metadata interoperability and in the process has developed the ABC model and vocabulary [2].

The hypothesis is that such an approach will lead to more efficient, scalable machine-translations between heterogeneous metadata descriptions. To test this hypothesis and to evaluate the interoperability capabilities of the ABC model, we applied it to some real multimedia examples and analysed the results of mapping from the ABC model to various different metadata domains using XSLT [3]. This work revealed serious limitations in XSLT's ability to support flexible dynamic semantic mapping. To overcome this, we developed MetaNet [4], a metadata term thesaurus which provides the additional semantic knowledge that is non-existent within declarative XML-encoded metadata descriptions.

This paper describes the optimum metadata mapping approach determined from applying the ABC model to a small test set of multimedia examples. This approach combines:

- the ABC event-aware metadata model, developed within the Harmony project, as the underlying model for scalable generic mappings between domain-specific vocabularies, with;
- XSLT for parsing XML descriptions and performing structural and syntactic mapping, and;
- MetaNet, a metadata term thesaurus, to provide the semantic knowledge required to enable semantic mapping between metadata terms from different domains or standards.

2 Definitions of Terms

This section defines the key terms used throughout the remainder of the paper:

- *Metadata* - data about data - or more commonly "descriptive information about Web resources". The use of standardized descriptive metadata can substantially improve the discovery and retrieval of relevant networked resources. Different communities or domains have developed their own standardized metadata vocaularies to meet their specific needs.
- *Vocabularies* - shared terminologies with commonly agreed-upon semantics for a domain. Common vocabularies enable search engines, agents, authors and users to communicate within a domain.
- *Schemas* - provide a standard way of defining standard domain-specific vocabularies by defining a common set of elements, their semantics and the relationships between the elements.
- *Ontology* - a formal description of the concepts, roles and relationships that exist for an agent or community of agents. Ontologies provide a shared and common understanding of a domain that can be communicated across people and applications, and play a major role in supporting information exchange and discovery.
- *Thesaurus* - the vocabulary of a controlled indexing language, formally organized so that the *a priori* relationships between concepts (for example "broader" and "narrower") are made explicit. [7]
- *Metadata Thesaurus* - a thesaurus (defined according to ISO 2788 standard for monolingual thesauri [7]) which defines the relationships between metadata terms from different domain vocabularies.

3 Related Work

Thesauri have been used to improve the precision and recall of information retrieval systems for over 30 years. The introduction of automated information retrieval has caused a dramatic increase in the demand for vocabulary control, particularly in the last decade. Examples of well known thesauri used to provide authority control over the terms used for indexing documents in the bibliographic, medical and cultural domains respectively are: the Library of Congress Subject Headings (LCSH) [9], the Medical Subject Headings (MeSH) [10] and the Art and Architecture Thesaurus (AAT). [11] In addition, thesauri have been used within information retrieval systems to improve retrieval effectiveness by providing semantic roadmaps. [12], [13], [14]

Since the emergence of the Internet, a great deal of effort has been invested in the development of metadata vocabularies to enable the exchange and discovery of information across different applications and domains. Metadata vocabularies such as Dublin Core [15], USMARC [16], INDECS [17], MPEG-7 [18], FGDC [19], IEEE LOM [20] and CIDOC CRM [21] provide standardized sets of descriptive elements to enable the exchange of resources for specific applications or domains. Although these standards enable interoperability within domains, they introduce the problem of incompatibility between disparate and heterogeneous metadata descriptions or schemas across domains.

A literature survey reveals many different proposals for improving interoperability between domain-specific vocabularies, thesauri and ontologies in the context of information retrieval and exchange. These range from database schema integration [22], to the use of ontologies in organizing and integrating networked information systems (e.g. OBSERVER [23], InfoSleuth [24], OntoSeek [25]) to the merging of monolingual [26] and multilingual thesauri. [8] Two of the major research issues have been categorizing

the complex kinds of interthesaurus semantic relationships which exist [27] and automating the detection of these relationships during the merging process. [28]

More recently the approach to merging thesauri has been to represent them formally using RDF Schemas [29] and to use inference engines to automate the merging - such as has been proposed in the Ontology Inference Layer (OIL). [30]

In this paper we are not so much concerned with the specific process by which MetaNet is generated or with expressing the complete set of possible term relations (as described in ISO 2788) in MetaNet. Our primary objective is to generate a thesaurus which specifies (an albeit simplified) set of semantic relationships between metadata terms from a number of different domain schemas relative to the ABC underlying vocabulary (the preferred terms) and hence also to each other. Our goal is then to demonstrate how this semantic knowledge can be represented in a machine-readable format (RDF Schema) and extracted and combined with the syntactic and structural mapping capabilities of XSLT to enable the implementation of flexible dynamic mappings between metadata descriptions from different domains.

4 Overview of the ABC Underlying Metadata Model

The Harmony Project [1] is investigating a generic approach to metadata interoperability through the development of an event-aware metadata model. The ABC model [2] defines a set of fundamental classes which provide the building blocks for expression (through sub-classing) of application-specific or domain-specific metadata vocabularies. The base classes, shown below, were determined by analysing commonalities between different communities' metadata models (including: Dublin Core [15]; INDECS [17]; MPEG-7 [18]; CIDOC CRM [21]; IFLA [31].)

- Resources
- Events
- Inputs and Outputs
- Acts
    - Agents
    - Roles
- Context
    - Time
    - Place
- Event Relations

ABC adopts an event-aware view for modeling the relationship between the various manifestations of a creation. This event-aware view provides semantically clear attachment points for the association of properties among the various manifestations, events and contributors (agents) involved in a resource's lifecycle. In addition, ABC provides a multiple views philosophy for metadata modeling and recipes for inter-conversion between those views. If life-cycle information is required, the event model can be used. When single resource metadata is needed, a resource-centric state model is used. Figure 1 shows the UML representation for the ABC metadata model.
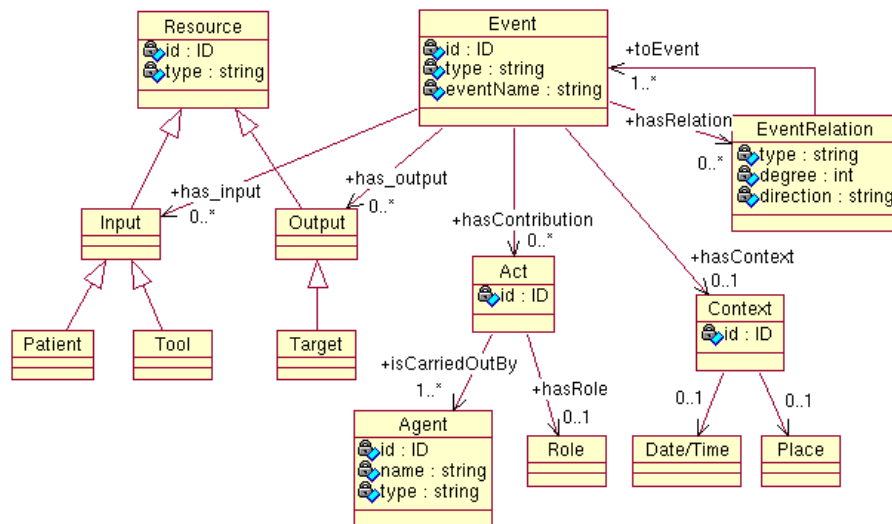

Figure 1. UML representation of the ABC metadata model

5 A Simple Example

To test the ABC model and evaluate XSLT for metadata mapping, we considered the following simple illustrative example:

> "A resource which is a 130 min audio (MP3) recording of a 'Live at Lincoln Center' performance. The Orchestra is the New York Philharmonic. The performance was on April 7, 1998 at 8 pm Eastern Time. The musical score performed is 'Concerto for Violin'. Copyright for the entire performance is held by Lincoln Center for the Performing Arts."

First we describe this resource using the ABC model. We then attempt to map from the ABC description to Dublin Core, MPEG-7 and ID3 [32] descriptions respectively, using XSLT. Figure 2 illustrates the two steps involved in mapping from the ABC metadata model to resource-centric models such as Dublin Core, MPEG-7 and ID3:

1. The structural mapping step involves transferring event properties to the output resource and creating a relationship between the output and input resources associated with the event.
2. The semantic mapping step involves mapping the properties attached to the output resource to semantically-equivalent properties in the output domain.

Appendix A contains the corresponding ABC, Resource-centric, Dublin Core, ID3 and MPEG-7 descriptions.
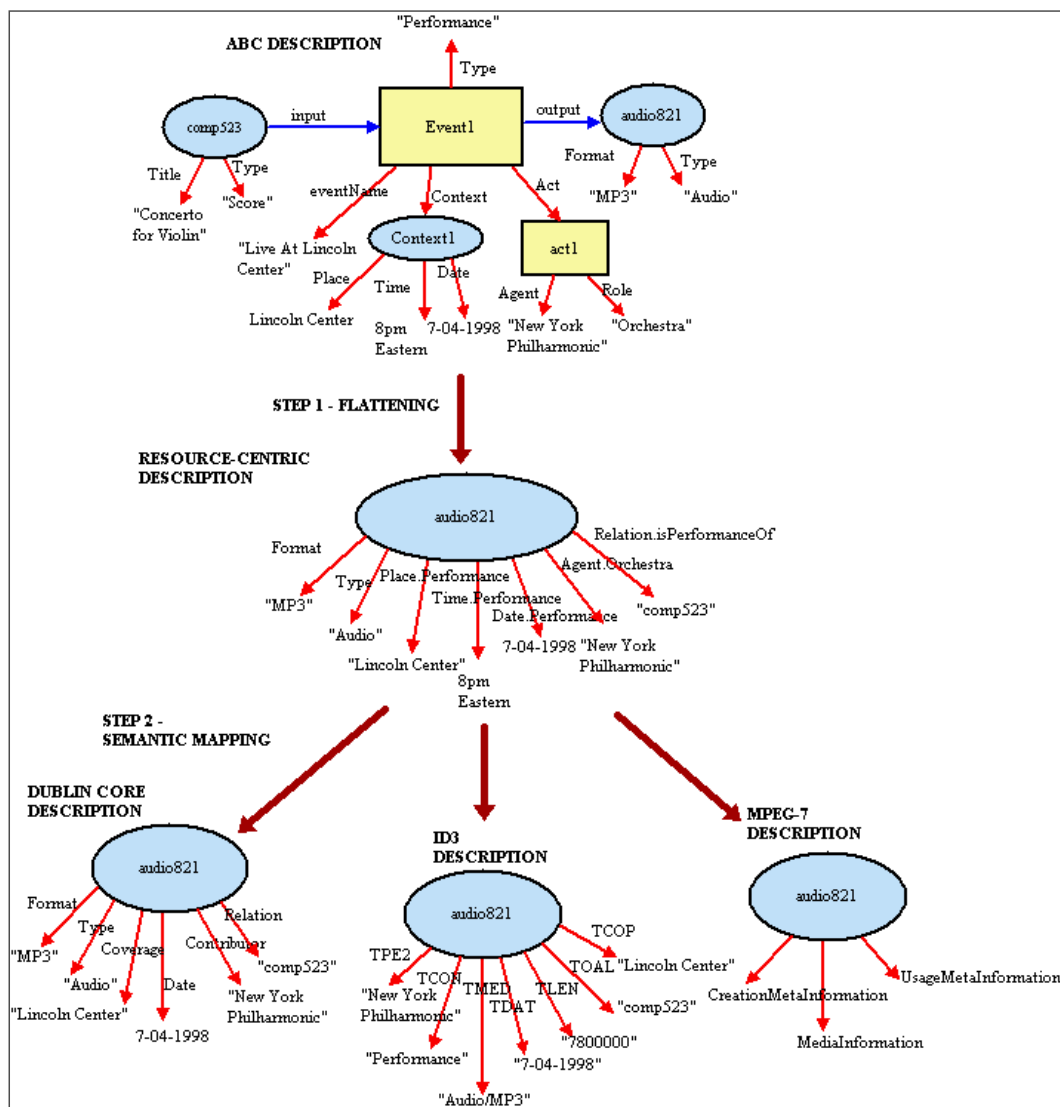
Figure 2. Transformation from the ABC event-aware model to three different resource-centric models

5.1 Structural Mapping Rules

For events which generate an output resource from an input resource, the transformation from an event-aware metadata model to a simple resource-centric metadata model consists of the following steps:

1. The Date, Time and Place properties within the Event's Context node can be qualified using the Event Type and transferred to the target output resource, e.g. Date.Performance, Time.Performance, Place.Performance;
2. The Role property of each Act associated with an event becomes a qualifier on the Agent property which is attached to the target output resource and its value is the Act's Agent Name, e.g. the Agent.Orchestra property has value "New York Philharmonic";
3. A Relation property arc is generated from the event type (e.g. Performance -> Relation.isPerformanceOf) and is attached to the target output resource. The value of this property is the patient input resource of the event (e.g. "comp523").
4. All other existing properties of the input and output resource remain the same.

Other inheritance and metadata derivation rules may be possible but these require further investigation.

For example, a Description property for the output resource can be generated from the Event Type and the input resource's Title e.g. "Performance of 'Concerto for Violin'". Or in many cases, the *Title* property can be inherited by the output resource directly from the Title property of either the input resource or the event.

6 An Evaluation of XSLT for Metadata Mappings

The Extensible Style Language (XSL) [3] consists of a transformation language (XSLT) and a formatting language. The transformation language XSLT (which acts independently of the formatting language) provides elements that define rules for how one XML document is transformed into another XML document. The transformed XML document may use the markup and DTD of the original document or it may use a completely different set of tags. The ability of XSLT to transform data from one XML representation to another makes it appear to be ideal for metadata interchange applications.

An XSL document contains a list of templates and rules. A template rule has a pattern specifying the trees it applies to and a template to be output when the pattern is matched. When an XSL processor formats an XML document using an XSL style sheet, it scans the XML document tree looking through each sub-tree in turn. As each tree in the XML document is read, the processor compares it with the pattern of each template rule in the style sheet. When the processor finds a tree that matches a template rule's pattern, it outputs the rule's template. This template generally includes some markup, some new data and some data copied out of the tree from the original XML document.

Using XSLT and the Xalan [33] XSLT processor we developed XSL programs for transforming the ABC description above to DC, ID3 and MPEG-7 descriptions, respectively. Appendix B shows the resulting XSL files.

The mapping implementations in Appendix B revealed that although XSLT works well for the structural mapping from an event model to a resource-centric model based on the set of rules described in Section 3.1, it is inadequate for implementing flexible dynamic semantic mappings between metadata vocabularies. This is due to:

- XSLT's limited capabilities for handling variable input descriptions based on schemas which are not tightly constrained;
- The non-existence of machine-understandable semantic information in declarative XML-encoded metadata descriptions;
- Processor-dependent handling of input parameters and procedural code extensions;
- Limited string manipulation and comparison functions, e.g. it is not possible to perform case-insensitive string comparisons within XSLT.

The mappings revealed that if the input XML descriptions are relatively fixed and tightly constrained, then the semantic mappings can be hardwired and XSLT is adequate. But if the input descriptions are at all variable or unpredictable (e.g. undefined domain specific sub-classing and attributes) then XSL simply cannot cope. Cawsey investigated the use of XSLT for customizing RDF descriptions, reaching similar conclusions. [34]

Below are listed a number of possible approaches to handling the semantic mapping problem. The approach chosen is a balance between simplicity on the one hand, and flexibility or scalability on the other. The wider the targeted scope of interoperability,

the more difficult it is to achieve accurate, precise mappings. Below is a list of mapping approaches in increasing order of both scope and difficulty:

1. Hardwire crosswalks between metadata terms from specific metadata domains (easy, but only works for fixed input);
2. Extract mappings from a pre-defined multiple-domain mapping matrix;
3. Determine the semantic mappings from a metadata term ontology;
4. Determine the semantic mappings from a generic ontology such as WordNet;
5. Determine the semantic mappings from a dynamically generated ontology created by using inferencing to merge multiple domain-specific ontologies.

By reducing the scope of the problem to interoperability between existing metadata standards, then the fully generic approaches (e.g., 4 and 5 above) become unnecessarily complex. Hence in the remainder of this paper we investigate the less complex but still moderately flexible approaches (2 and 3) based on a mapping matrix and a metadata term ontology, respectively.

7 Semantic Mapping via a Mapping Matrix

The second approach in the list above involves linking a mapping matrix to the XSLT processor. The mapping matrix explicitly defines the semantic mappings between a fixed set of metadata vocabularies from a number of different domains. Figure 3 illustrates such a mapping matrix. If XPath [35] is used to specify the elements, then to some extent both the structural and semantic mappings can be defined.

ABC Element	DC Element	ID3 Element	MPEG-7 Path
Resource/Title	Title	TIT2	CreationMetaInformation/Creation/Title/TitleText (@TitleType="original")
	Creator	TPE1	CreationMetaInformation/Creation/Creator (@role="creator")
	Publisher	TPUB	UsageMetaInformation/Publication/Publisher
Event/Act/Agent	Contributor	IPLS(involved People List), TCOM(Composer), TENC(Encoder), TEXT(Lyricist), TOLY(OriginalLyricist), TOPE(Original Artist), TPE2(Band, Orchestra, Accompaniment), TPE3(Conductor), TPE4 (Interpreter, Remixer, Modifier)	CreationMetaInformation/Creation/Creator (@role)
Resource/Subject	Subject	TIT1	CreationMetaInformation/Creation/Classification/PackagedType
Resource/Description	Description	TIT3	CreationMetaInformation/Creation/CreationDescription
	Date.Creation	-	CreationMetaInformation/Creation/CreationDate
Event/Context/Date	Date.Publication	-	UsageMetaInformation/Publication/PublicationDate
	Date.Recording	TRDA	-
Resource/Type	Type	TCON	CreationMetaInformation/Classification/Genre
	Format	TFLT	MediaInformation.MediaProfile/MediaFormat/FileFormat
Resource/Format	Format.length	TLEN	-
	Format.size	TSIZ	-
Resource/Identifier	Identifier	UFID	MediaInformation/MediaIdentification/Identifier
Event/Input	Source	TOAL (Title of original recording or source)	-
Event/Context/Place	Coverage.Place	-	-

Table 1. Metadata mapping matrix

This approach has certain debilitating limitations, however. A matrix is only capable of specifying mappings which involve fairly simple one-to-one mappings, and a two-dimensional matrix will only work if the mappings are symmetrical in both directions across all the domains. If the mappings are asymetrical then the matrix becomes highly complex and multi-dimensional. However, the primary limitation of this approach is that it simply does not scale - as the number of domains grows and the mappings become asymmetrical, then the matrix becomes excessively complex and unwieldy.

8 Development of MetaNet, a Metadata Term Thesaurus

Rather than limiting the semantic mapping to a fixed number of domains/vocabularies (i.e. the number of columns in the mapping matrix), a more generic approach is to extract the mapping dynamically from a thesaurus of metadata terms, generated by formally defining relationships between metadata terms from a number of different domains' standardized vocabularies.

8.1 Intrathesaurus and Interthesaurus Relations

The ISO2788 standard for the identification and documentation of monolingual thesauri [7] identifies the following types of intrathesaurus relations:

- hierarchical
- associative
- equivalence

The hierarchical relation occurs between concepts having "broader/narrower" meanings. This can be further specialized into the generic (BTG/NTG), whole-part (BTP/NTP) and instance (BT/NT) relations. For the sake of simplicity, we have chosen only to model the BTG/NTG relation (a common practice among thesauri developers) and the equivalence relation, and not to include associative relations within MetaNet.

The ISO5964 standard for the documentation and establishment of multilingual thesauri [8] identifies the following types of interthesaurus relations:

- exact equivalence
- partial equivalence
- single to multiple equivalence
- inexact equivalence.

These relations indicate that the semantic relations between terms from different metadata vocabularies are likely to be much more complex than one-to-one exact equivalence and that even "exact equivalence" will be an approximation. However, because the scope of our problem is limited to relations between terms in a number of standardized English metadata vocabularies, then we can expect the frequency of more complex mappings to be less than for general natural language thesauri. For the first draft of MetaNet, we decided only to consider exact and partial equivalence relations and to combine them in the ET relation which defines equivalent/overlapping terms. If two different domains use two different metadata terms which are ETs in our thesaurus then we make the assumption that the domains are referring to semantically equivalent concepts.

Consequently the metadata term thesaurus which we have developed, *MetaNet* [4], contains only preferred terms (the ABC core vocabulary), equivalent/overlapping terms (ET), narrower terms (NT) and broader terms (BT), and attempts to encompass terms from the most significant and widely-used metadata vocabularies (Dublin Core, IFLA, IEEE LOM, INDECS).

8.2 Description of MetaNet

The objective of the *MetaNet* thesaurus is to provide the semantic knowledge required to enable machine understanding of equivalence and hierarchical (subtyping) relationships between metadata terms from different domains. The scope of this thesaurus is limited to the most significant metadata models/vocabularies used for describing attributes and events associated with resources and their life cycles. This encompasses metadata vocabularies from the bibliographic, museum, archival, record keeping and rights management communities. It has been developed by performing WordNet [36] searches using the core terms from the ABC vocabulary, and extracting those synonyms and hyponyms which could conceivably be used in a metadata scheme to represent the original core term. In addition, the results have been compared with the vocabularies of the DC, INDECS, IFLA, IMS and CIDOC CRM vocabularies to check that the majority of the terms used in these metadata dictionaries have been incorporated into the thesaurus.

A machine-readable RDF Schema representation of this thesaurus has been developed. [37] The RDF and RDF Schema elements, *Class*, *subClassOf*, *property*, *subPropertyOf* are used to define the hierarchical/subtyping and entity/attribute relationships between metadata elements. The RDFS *label* element is used to specify semantically equivalent terms which may be used. The ABC core vocabulary is used as the top-level set of preferred terms. Although this thesaurus has been generated manually, it could conceivably be generated automatically by using inferencing mechanisms to merge RDF Schemas from different domains, as has been proposed in the Ontology Inference Layer (OIL). [30]

For example, consider "Agent", which is a core term of the ABC vocabulary and hence a preferred term in the MetaNet thesaurus. Semantically equivalent terms for "Agent", commonly used within other metadata vocabularies, include: *actor, contributor, creator, player, doer, worker, performer*

Possible narrower terms or *hyponyms* for "Agent" include: *author, composer, artist, musician, . . etc.*

Table 2 is an excerpt from the RDF Schema which illustrates the representation for the "Agent" metadata term as well as its equivalent terms and a partial hierarchy of its narrower terms.

Table 2. Excerpt from the RDF Schema

```
<?xml version="1.0"?>
<rdf:RDF xml:lang="en"
      xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
      xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#">

<rdfs:Class rdf:ID="Agent">

    <rdfs:comment  xml:lang="en">The resources which contribute to or act
        in an event. Typically agents are people, groups of people,
        organisations or instruments.</rdfs:comment>

    <rdfs:label xml:lang="en">Actor</rdfs:label>
    <rdfs:label xml:lang="en">Contributor</rdfs:label>
    <rdfs:label xml:lang="en">Creator</rdfs:label>
    <rdfs:label xml:lang="en">Player</rdfs:label>
    <rdfs:label xml:lang="en">Doer</rdfs:label>
    <rdfs:label xml:lang="en">Worker</rdfs:label>
    <rdfs:label xml:lang="en">Performer</rdfs:label>

    <rdfs:subClassOf
        rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>

</rdfs:Class>

<rdfs:Class rdf:ID="Author">
    <rdfs:label xml:lang="en">Writer</rdfs:label>
    <rdfs:label xml:lang="en">Wordsmith</rdfs:label>
    <rdfs:subClassOf
                rdf:resource="#Agent"/>
</rdfs:Class>

<rdfs:Class rdf:ID="Journalist">
    <rdfs:label xml:lang="en">Columnist</rdfs:label>
    <rdfs:label xml:lang="en">Reporter</rdfs:label>
    <rdfs:subClassOf
                rdf:resource="#Author"/>
</rdfs:Class>

</rdf:RDF>
```

A Web search and browse interface to MetaNet has also been developed. [4] Users can search on any common metadata term and retrieve a list of equivalent terms, broader terms and narrower terms. Figure 3 shows the results of a search on the term "author".

Figure 3. Results of MetaNet search

9 Linking MetaNet to XSLT

Using XSLT it is possible to parse an input XML description and for each element encountered call a Java procedural code

extension which determines the equivalent term in the output domain from the semantic realtionships specified in the MetaNet thesaurus.

For example, suppose the Java program, Mapping.java, contains an extension function *readMetaNet*. For each element encountered during parsing of the input metadata description, the input element name (e.g. abc:Agent) and the output domain schema definition (e.g. the Dublin Core schema) are passed to the *readMetaNet* function. This function searches the MetaNet RDF Schema file for an element in the output schema definition that is equivalent to the input element name (e.g. dc:contributor), and returns this value. XSL creates a new output element with this name in the output description. Figure 4 illustrates the program flowchart.
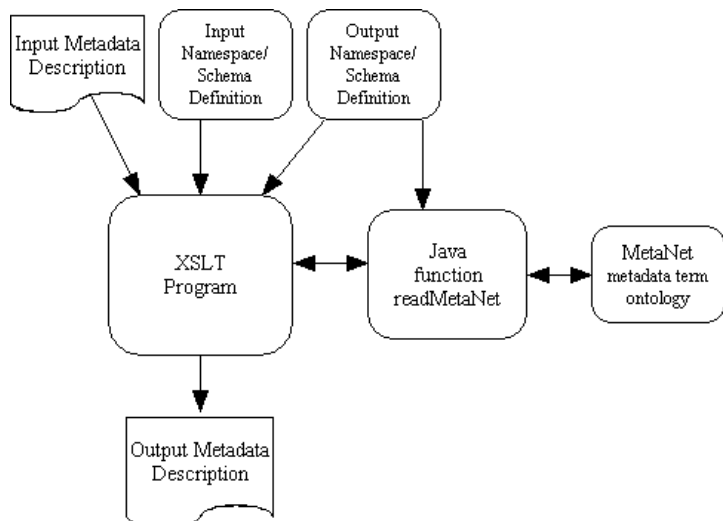

Figure 4. Program flow for metadata description mappings

The XSL code in Table 3 illustrates how to call a Java extension function, *readMetaNet*, from the main XSL file.

Table 3. XSL code to call a Java extension function, *readMetaNet,* from the main XSL file

```
<?xml version="1.0"?>
<xsl:stylesheet version="1.0"
        xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
        xmlns:dc ="http://purl.org/dc/elements/1.1/">
        xmlns:lxslt="http://xml.apache.org/xslt"
        xmlns:mapping="Mapping"
        extension-element-prefixes="mapping"
        version="1.0">

   <lxslt:component prefix="mapping" elements="*" functions="readMetaNet">
        <lxslt:script lang="javaclass" src="Mapping"/>
   </lxslt:component>

    <xsl:template match="ABC">
            <xsl:apply-templates />
    </xsl:template>

    <xsl:template match="*">
        <xsl:element name="mapping:readMetaNet(., 'dc')"/>
            <xsl:value-of select="."/>
        </xsl:element>
    </xsl:template>

</xsl:stylesheet>
```

Below is a high-level, simplistic algorithm describing the mapping process that is performed within the readMetaNet Java function in Figure 4.

Table 4. Algorithm describing the mapping process within the *readMetaNet* Java function in Figure 4

```
For each element in the input description
{
    Search for the input element name in the output domain schema;
    if (found) {
        Map the input element to the equivalent output domain element;
    }
    else {
        Extract the Equivalent Terms (ETs) for the input element from MetaNet;
        Search the output domain schema for each of the ETs;
        if (an ET is found)
        {
            Map the input element to the equivalent output domain element;
        }
        else {
            Extract the broader terms (BTs) for the input element from MetaNet;
            Search for each BT in the output domain namespace;
            if (a BT is found)
            {
                Map the input element to the broader output domain element;
            }
            else {
                Extract the narrower terms (NTs) for the input element from MetaNet;
                Search for each NT in the output domain namespace;
                if (a NT is found)
                {
                    Map the input element to the narrower output domain element;
                }
            }
        }
    }
}
```

```
} endFor
```

10 Conclusions, Limitations and Future Work

10.1 Conclusions and Limitations

Our evaluation of XSLT for mapping between metadata descriptions from different domains revealed that although XSLT is good for syntactical and structural mapping, semantic mappings need to be hardwired into the code. Flexible semantic mapping is only possible with the assistance of semantic knowledge bases provided by ontologies or thesauruses such as the MetaNet thesaurus described above.

The MetaNet thesaurus described here is a first draft English version, based on the vocabulary of the ABC model. Although it has only been applied to a relatively small sample set, some of the limitations of this thesaurus are already evident. These include its inability to support metadata vocabularies which use:

- Tokens, e.g. ID3 tags such as TPE2, which are semantically meaningless. This limitation can be overcome by either explicitly including such tags in the thesaurus or searching the definitions (rather than element names) in the output namespace for the input element name or its semantically equivalent terms;
- Abbreviations, e.g. acc.no.;
- Qualifers or hybrid words joined by a variety of connectors, e.g. UserClass, Assistant Editor, Art_Director, Time-span. This problem can be solved to some extent by including "associated terms" in the thesaurus and by ignoring typical "connectors".

In addition, the inherently ambiguous nature of language leads to the following problems:

- Metadata terms with multiple possible meanings, e.g."condition" - this could be the current state of an object or it could be a restriction on the permissable use of a resource. This can be overcome by the use of unambiguous metadata terms by schema designers.
- Multiple possible spellings for the same word, e.g. artefact/artifact, colour/color.
- This thesaurus is based on nouns, e.g. "creator", "publisher", and does not search for related verbs, adverbs, adjectives in various tenses which could be used to express the same semantics, e.g. "created_by", "published_by". This problem could, to some extent, be overcome through the use of stemming.

Currently only English is supported. However, we believe that this thesaurus could be extended to provide equivalent or overlapping terms for the ABC vocabulary in other languages by following the recommendations specified in ISO5964. [8]

10.2 Future Work

So far the ABC model has only been tested on a relatively small sample set. We intend carrying out more extensive evaluation of both the ABC model and the hybrid mapping approach, by applying them to metadata tranformations between large sets of sample records provided by a number of different CIMI [38] member organisations. The plan is to build a testbed using multimedia museum resources and metadata descriptions provided by CIMI members and to use this testbed to implement and evaluate metadata interoperability between different museums' descriptions.

The Harmony ABC model exhibits many similarities with the CIDOC Conceptual Reference Model (CRM) [21], a domain ontology developed by the CIDOC Committee of the International Council of Museums. In the near future we plan to investigate the possible merging of the Harmony model and the CIDOC CRM model into a single ontology. We plan to use the CIMI testbed described above to evaluate the "super" ontology resulting from harmonization of these two models.

The mapping implementations above have all involved mapping from an event-aware metadata model to a resource-centric metadata model. We are also interested in the rules and mechanisms required for machine translations between metadata descriptions based on different event-aware metadata models, e.g. from ABC to INDECS or CIDOC CRM.

We would also like to investigate mapping between "application profiles" or schemas which mix metadata elements imported from multiple different namespaces. The test examples considered so far only present the problem of mapping from a single domain's metadata description to another single domain's metadata description, e.g. pure DC to pure MPEG-7. A situation that will become increasingly common in the future is the need to map from a schema which imports elements from multiple namespaces to another schema which imports a different set of elements from multiple namespaces. In addition, each schema may impose its own local

- structural constraints, e.g. parent/child relationships
- cardinality/occurrence constraints
- datatyping, enumeration and formatting constraints on the element values.

We believe the approach proposed in this paper will support mapping between mixed-domain "application profiles", but need to test this through further research involving machine translations between metadata descriptions which conform with both complex local usage constraints, (defined by XML Schemas [39]), as well as namespace-specific semantic definitions (defined by RDF Schemas).

Acknowledgements

References

[1]The Harmony Project Home Page http://www.ilrt.bris.ac.uk/discovery/harmony/

[2] C. Lagoze, J. Hunter and D. Brickley (2000) "An Event-Aware Model for Metadata Interoperability". *ECDL 2000*, Lisbon, September

[3] XSL Transformations (XSLT) Version 1.0 (1999) W3C Recommendation, 16 November http://www.w3.org/TR/xslt.html

[4] MetaNet Search Page http://sunspot.dstc.edu.au:8888/Metanet/Top.html

[5] RDF Schema Specification 1.0 (2000) W3C Candidate Recommendation, 27 March http://www.w3.org/TR/rdf-schema/

[6] Dublin Core/MARC/GILS Crosswalk (1999) November http://lcweb.loc.gov/marc/dccross.html

[7] ISO 2788 (1986) Documentation -- Guidelines for the Development and Establishment of Monolingual Thesauri

[8] ISO 5964 (1985) Documentation -- Guidelines for the Development and Establishment of Multilingual Thesauri

[9] Library of Congress Subject Headings, Cataloging Distribution Service, Library of Congress http://lcweb.loc.gov/cds/lcsh.html

[10] Medical Subject Headings home page http://www.nlm.nih.gov/mesh/meshhome.html

[11] Art and Architecture Thesaurus Browser, Getty Research Institute http://shiva.pub.getty.edu/aat_browser/

[12] C. Paice (1991) "A Thesaural Model of Information Retrieval". *Information Processing and Management*, 27(5):433-447

[13] A.R. Aronson (1994) "Exploiting a Large Thesaurus for Information Retrieval". RIAO 94, New York, October

[14] W. Bruce Croft and J. Yufeng (1994) "An Association Thesaurus for Information Retrieval". RIAO 94, New York, October

[15] Dublin Core Metadata Initiative http://purl.org/dc/

[16] MARC Standards, Library of Congress Network Development and MARC Standards Office http://lcweb.loc.gov/marc/marc.html

[17] G. Rust and M. Bide (1999) "The indecs Metadata Schema Building Blocks". Indecs Metadata Model, November http://www.indecs.org/pdf/schema.pdf

[18] MPEG-7 Home Page http://www.darmstadt.gmd.de/mobile/MPEG7/index.html/

[19] Content Standard for Digital Geospatial Metadata (CSDGM) http://www.fgdc.gov/metadata/contstan.html

[20] IEEE Learning Technology Standards Committee's Learning Object Meta-data Working Group, Approved Working Draft WD5 Learning Object Meta-data Scheme http://ltsc.ieee.org/wg12/

[21] ICOM/CIDOC Documentation Standards Group (1999) Revised Definition of the CIDOC Conceptual Reference Model, September http://www.geneva-city.ch:80/musinfo/cidoc/oomodel

[22] C. Batini, M. Lenzerini and S.B. Navathe (1986) "A comparative analysis of methodologies for database schema integration". *ACM Computing Surveys*, 18(4):323-364, December

[23] E. Mena, V. Kashyap, A. Sheth and A. Illarramendi (1996) "OBSERVER: An Approach for Query Processing in Global Information Systems based on Interoperation across Pre-existing Ontologies". *Proceedings of the 1st IFCIS International Conference on Cooperative Information Systems* (CoopIS'96), Brussels, Belgium, June (IEEE Computer Society Press)

[24] R. Bayardo, *et al.* (1997) "InfoSleuth: Agent-based Semantic Integration of Information in Open and Dynamic Environments". *Proceedings of ACM SIGMOD Conference on Management of Data*, Tucson, Arizona, May, pp. 195-206

[25] N. Guarino, C. Masolo and G. Vetere (1999) "Ontoseek: Content-based Access to the Web". *IEEE Intelligent Systems*, Vol. 14, No. 3, May/June, 70-80

[26] H. Mili and R. Rada (1988) "Merging Thesauri: Principles and Evalauation". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(2):204-220

[27] M. Doerr and I. Fundulaki (1998) "A proposal on extended interthesaurus links semantics". Technical Report TR-215, Institute of Computer Science-FORTH, March

[28] M. Sintichakis and P. Constantopoulos (1997) "A Method for Monolingual Thesauri Merging". *Proceedings of the 20th ACM International Conference on Research and Development in Information Retrieval* (ACM SIGIR), Philadeplphia, PA, USA, July

[29] B. Amann and I. Fundulaki (1999) "Integrating Ontologies and Thesauri to Build RDF Schemas". In ECDL'99: Research and Advanced Technologies for Digital Libraries, Paris, France, September, Lecture Notes in Computer Science (Springer-Verlag), pp. 234-253

[30] Ontology Inference Layer http://www.ontoknowledge.org/oil/

[31] International Federation of Library Associations and Institutions (IFLA) (1998) Functional Requirements for Bibliographic Records, March http://www.ifla.org/VII/s13/frbr/frbr.pdf

[32] ID3 Tag Version 2.3.0 http://www.id3.org/id3v2.3.0.html

[33] Xalan-Java Overview http://xml.apache.org/xalan/overview.html

[34] A. Cawsey (2000) "Presenting tailored resource descriptions: Will XSLT do the job?". *WWW9 conference*, Amsterdam, May http://www.cee.hw.ac.uk/~alison/www9/paper.html

[35] XML Path Language (XPath) Version 1.0 (1999) November http://www.w3.org/TR/xpath

[36] WordNet - a Lexical Database for English http://www.cogsci.princeton.edu/~wn/online/

[37] RDF Schema Representation of the MetaNet Thesaurus (2000) October http://archive.dstc.edu.au/maenad/metanet.rdf

[38] CIMI Consortium for Interchange of Museum Information http://www.cimi.org/

[39] XML Schema Language http://www.w3.org/XML/Schema

Appendix A

A.1 ABC Description of Example Resource

```
<?xml version="1.0"?>
<ABC>

  <Event id="E1" Type="Performance">
    <Title>Live At the Lincoln Centre</Title>
    <Context>
      <Date>7/4/98</Date>
      <Time>20:00</Time>
      <Place>Lincoln Centre</Place>
    </Context>
    <Act id="Act1">
      <Agent>New York Philharmonic</Agent>
      <Role>Orchestra</Role>
    </Act>
    <Input id="comp523"/>
    <Output id="audio821"/>
    <Rights>
        Lincoln Center for Performing Arts
    </Rights>
  </Event>


  <Resource id="comp523">
    <Type>Musical Score</Type>
    <Title>Concerto for Violin</Title>
  </Resource>


  <Resource id="audio821">
```

```
    <Type>audio</Type>
    <Format>MP3</Format>
    <Length units="mins">
        130
    </Length>
</Resource>


<ABC>
```

A.2 Simple Resource-centric Description of Example Resource

```
<?xml version="1.0"?>
    <Resource id="audio821">
        <Title>Live At Lincoln Center</Title>
        <Date.Performance>1998-07-04</Date.Performance>
        <Time.Performance>20:00</Time.Performance>
        <Place.Performance>Lincoln Centre</Place.Performance>
        <Agent.Orchestra>New York Philharmonic</Agent.Orchestra>
        <Relation.isPerformanceOf>comp523</Relation.isPerformanceOf>
        <Description>Performance of 'Concerto for Violin'</Description>
        <Rights>
            Lincoln Center for Performing Arts
        </Rights>
        <Type>audio</Type>
        <Format>MP3</Format>
        <Length units="mins">
            130
        </Length>
    </Resource>
```

A.3 Dublin Core Description of Simple Example

```
<?xml version="1.0"?>
<!DOCTYPE rdf:RDF SYSTEM "http://purl.org/dc/schemas/dcmes-xml-20000714.dtd">
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
           xmlns:dc="http://purl.org/dc/elements/1.1/">
    <rdf:Description about="audio821">
        <dc:Title>Live At Lincoln Center</dc:Title>
        <dc:Date.Performance>1998-07-04T20:00-05:00</dc:Date.Performance>
        <dc:Coverage>Lincoln Centre</dc:Coverage>
        <dc:Contributor.Orchestra>New York Philharmonic</dc:Contributor.Orchestra>
        <dc:Relation.isPerformanceOf>comp523</dc:Relation.isPerformanceOf>
        <dc:Description>Performance of 'Concerto for Violin'</dc:Description.Performance>
        <dc:Rights>
            Lincoln Center for Performing Arts
        </dc:Rights>
        <dc:Type>audio</dc:Type>
        <dc:Format>MP3</dc:Format>
    </rdf:Description>
<rdf:RDF>
```

A.4 MPEG-7 Description

```
<?xml version="1.0"?>
<MPEG-7 id="audio821">
  <CreationMetaInformation>
    <Creation>
      <Title>Live at Lincoln Center</Title>
      <Creator>
        <Role>Orchestra</role>
        <Name>New York Philharmonic</Name>
      </Creator>
      <CreationDate>
        <day>7</day>
        <month>4</month>
        <year>1998</year>
      </CreationDate>
      <Location>
        <PlaceName>Lincoln Center</PlaceName>
      </Location>
    </Creation>
    <Classification>
      <Genre>Performance</Genre>
    </Classification>
  </CreationMetaInformation>
  <MediaInformation>
    <MediaProfile>
      <MediaFormat>
        <Medium>MP3</Medium>
        <Length><m>130</m></Length>
      </MediaFormat>
    </MediaProfile>
  </MediaInformation>
  <UsageMetaInformation>
    <Rights>
      <RightsId IdOrganization='Lincoln Center'/>
    </Rights>
  </UsageMetaInformation>
</MPEG-7>
```

A.5 ID3 Description

```xml
<?xml version="1.0"?>
<ID3>
<!-- Unique Identifier -->
  <UFID>audio821</UFID>

<!-- Title -->
  <TIT2>Live At Lincoln Center</TIT2>

<!-- Orchestra -->
  <TPE2>New York Philharmonic</TPE2>

<!-- Type or Genre -->
  <TCON>Performance</TCON>

<!-- Media Type sound originated from-->
  <TMED>Audio/MP3</TMED>

<!-- Date Recorded -->
  <TDAT>7/4/98</TDAT>

<!-- Time Recorded -->
  <TIME>2100</TIME>

<!-- Length in millisecs-->
  <TLEN>7800000</TLEN>

<!-- Original recording or source -->
  <TOAL>comp523</TOAL>

<!-- Copyright Message -->
  <TCOP>Lincoln Center of Performing Arts</TCOP>
</ID3>
```

Appendix B

B.1 XSL for Transforming from ABC to DC

```xml
<?xml version="1.0"?>
<xsl:stylesheet version="1.0"
      xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
      xmlns:dc ="http://purl.org/dc/elements/1.1/">

    <xsl:output method="xml" indent="yes"/>

    <xsl:template match="ABC">
        <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
                 xmlns:dc ="http://purl.org/dc/elements/1.1/">
          <rdf:Description>
            <xsl:apply-templates select="Event"/>
            <xsl:apply-templates select="Resource"/>
          </rdf:Description>
        </rdf:RDF>
    </xsl:template>

    <xsl:template match="Event">
        <xsl:apply-templates select="Output"/>
        <xsl:apply-templates select="Context"/>
        <xsl:apply-templates select="Act"/>
        <xsl:apply-templates select="Input"/>
        <xsl:apply-templates select="Title"/>
        <xsl:apply-templates select="Rights"/>
    </xsl:template>

    <xsl:template match="Output">
        <xsl:attribute name="about">
           <xsl:value-of select="@id"/>
        </xsl:attribute>
        <xsl:copy-of select="*"/>
    </xsl:template>

    <xsl:template match="Context">
        <xsl:apply-templates select="Date"/>
        <xsl:apply-templates select="Place"/>
    </xsl:template>

    <xsl:template match="Date">
        <xsl:element name="dc:Date.{../../@Type}">
           <xsl:value-of select="."/>
        </xsl:element>
    </xsl:template>

    <xsl:template match="Place">
        <xsl:element name="dc:Coverage">
           <xsl:value-of select='.'/>
        </xsl:element>
    </xsl:template>

    <xsl:template match="Act">
        <xsl:element name="dc:Contributor.{Role}">
           <xsl:value-of select='Agent'/>
        </xsl:element>
```

```
        </xsl:template>

        <xsl:template match="Input">
            <xsl:element name="dc:Relation.is{../@Type}Of">
                <xsl:value-of select="@id"/>
            </xsl:element>
        </xsl:template>

        <xsl:template match="Title">
            <xsl:element name="dc:Title">
                <xsl:value-of select="."/>
            </xsl:element>
        </xsl:template>

        <xsl:template match="Rights">
            <xsl:element name="dc:Rights">
                <xsl:value-of select="."/>
            </xsl:element>
        </xsl:template>

        <xsl:template match="Resource">
            <xsl:if test="@id=../Event/Output/@id">
                <xsl:apply-templates select="Type"/>
                <xsl:apply-templates select="Format"/>
            </xsl:if>
            <xsl:if test="@id=../Event/Input/@id">
                <xsl:element name="dc:Description">
                    <xsl:value-of select="../Event/@Type"/> of
                    <xsl:text>"</xsl:text>
                    <xsl:value-of select="Title"/>
                    <xsl:text>"</xsl:text>
                </xsl:element>
            </xsl:if>
        </xsl:template>

        <xsl:template match="Type">
            <xsl:element name="dc:Type">
                <xsl:value-of select="."/>
            </xsl:element>
        </xsl:template>

        <xsl:template match="Format">
            <xsl:element name="dc:Format">
                <xsl:value-of select="."/>
            </xsl:element>
        </xsl:template>

</xsl:stylesheet>
```

B.2 XSL for Transforming from ABC to ID3

```
<?xml version="1.0"?>
<xsl:stylesheet version="1.0"
    xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
    xmlns:id3 ="http://www.id3.org/id3v2.3.0">

    <xsl:output method="xml" indent="yes"/>

    <xsl:template match="ABC">
        <id3:ID3 xmlns:id3="http://www.id3.org/id3v2.3.0">
            <xsl:apply-templates select="Event"/>
            <xsl:apply-templates select="Resource"/>
        </id3:ID3>
    </xsl:template>

    <xsl:template match="Event">
        <xsl:apply-templates/>
    </xsl:template>

    <xsl:template match="Output">
        <xsl:element name="id3:UFID">
            <xsl:value-of select="@id"/>
        </xsl:element>
    </xsl:template>

    <xsl:template match="Context">
        <id3:TDAT>
            <xsl:value-of select="Date"/>
        </id3:TDAT>
        <id3:TIME>
            <xsl:value-of select="Time"/>
        </id3:TIME>
    </xsl:template>

    <xsl:template match="Act">
        <xsl:element name="id3:TPE2">
            <xsl:value-of select='Agent'/>
        </xsl:element>
    </xsl:template>

    <xsl:template match="Input">
        <xsl:element name="id3:TOAL">
            <xsl:value-of select="@id"/>
        </xsl:element>
```

```
            </xsl:template>

            <xsl:template match="Title">
                <xsl:element name="id3:TIT2">
                    <xsl:value-of select="."/>
                </xsl:element>
            </xsl:template>

            <xsl:template match="Rights">
                <xsl:element name="id3:TCOP">
                    <xsl:value-of select="."/>
                </xsl:element>
            </xsl:template>

            <xsl:template match="Resource">
                <xsl:if test="@id=../Event/Output/@id">
                    <xsl:apply-templates select="Format"/>
                    <xsl:apply-templates select="Length"/>
                </xsl:if>
            </xsl:template>

            <xsl:template match="Format">
                <xsl:element name="id3:TMED">
                    <xsl:value-of select="."/>/<xsl:value-of select="../Type"/>
                </xsl:element>
            </xsl:template>

            <xsl:template match="Length">
                <xsl:element name="id3:TLEN">
                    <xsl:value-of select=".*60*1000"/>
                </xsl:element>
            </xsl:template>


    </xsl:stylesheet>
```

B.3 XSL for Transforming ABC to MPEG-7

```
<?xml version="1.0"?>
<xsl:stylesheet version="1.0"
        xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
        xmlns:mpeg7 ="http://www.mpeg7.org/2000/MPEG7_schema/">

        <xsl:output method="xml" indent="yes"/>

        <xsl:template match="ABC">
            <MPEG-7>
                <xsl:apply-templates select="Event"/>
                <xsl:apply-templates select="Resource"/>
            </MPEG-7>
        </xsl:template>

        <xsl:template match="Event">
            <xsl:apply-templates select="Output"/>

            <CreationMetaInformation>
                <Creation>
                    <xsl:apply-templates select="Title"/>
                    <xsl:apply-templates select="Act"/>
                    <xsl:apply-templates select="Context"/>
                    <xsl:apply-templates select="Input"/>
                </Creation>

                <Classification>
                    <xsl:element name="Genre">
                        <xsl:value-of select="@Type"/>
                    </xsl:element>
                </Classification>
            </CreationMetaInformation>

            <xsl:apply-templates select="Rights"/>
        </xsl:template>

        <xsl:template match="Output">
            <xsl:attribute name="id">
                <xsl:value-of select="@id"/>
            </xsl:attribute>
        </xsl:template>

        <xsl:template match="Title">
            <xsl:element name="Title">
                <xsl:value-of select="."/>
            </xsl:element>
        </xsl:template>

        <xsl:template match="Act">
            <Creator>
            <xsl:element name="Role">
                <xsl:value-of select='Role'/>
            </xsl:element>
            <xsl:element name="Name">
                <xsl:value-of select='Agent'/>
            </xsl:element>
            </Creator>
        </xsl:template>
```

```
<xsl:template match="Context">
   <CreationDate>
      <xsl:variable name="date" select='Date'/>
      <xsl:variable name="my" select="substring-after($date,'/')"/>
      <xsl:element name="day">
         <xsl:value-of select="substring-before($date,'/')"/>
      </xsl:element>
      <xsl:element name="month">
         <xsl:value-of select="substring-before($my,'/')"/>
      </xsl:element>
   <xsl:element name="year">
   <xsl:value-of select="substring-after($my,'/')"/>
   </xsl:element>
   </CreationDate>
   <Location>
   <xsl:element name="PlaceName">
   <xsl:value-of select='Place'/>
   </xsl:element>
   </Location>
</xsl:template>

<xsl:template match="Input">
</xsl:template>

<xsl:template match="Rights">
<UsageMetaInformation>
<Rights>
<xsl:element name="RightsId">
<xsl:value-of select="."/>
</xsl:element>
</Rights>
</UsageMetaInformation>
</xsl:template>


<xsl:template match="Resource">
<xsl:if test="@id=../Event/Output/@id">
<MediaInformation>
<MediaProfile>
<MediaFormat>
<xsl:apply-templates select="Format"/>
<xsl:apply-templates select="Length"/>
</MediaFormat>
</MediaProfile>
</MediaInformation>
</xsl:if>
</xsl:template>

<xsl:template match="Format">
<xsl:element name="Medium">
<xsl:value-of select="."/>
</xsl:element>
</xsl:template>

<xsl:template match="Length">
<Length>
<xsl:if test="@units='mins'">
<xsl:element name="m">
<xsl:value-of select="."/>
</xsl:element>
</xsl:if>
</Length>
</xsl:template>

</xsl:stylesheet>
```