This project is funded under the
ICT Policy Support Programme part of the
Competitiveness and Innovation Framework Programme.

Project

Project Acronym:	AthenaPlus
Grant Agreement number:	325098
Project Title:	Access to cultural heritage networks for Europeana

Deliverable

Deliverable name:	**Description of the LIDO to EDM mapping**
Deliverable number:	**D3.2**
Delivery date:	9
Dissemination level:	Public
Status	Final
Authors (organisation)	Nikolaos Simou (NTUA)
Contributors (organisation)	Regine Stein (UNIMAR) Eleni Tsalapati (NTUA) Nasos Drosopoulos (NTUA)
Reviewers (organisation)	Sam H. Minelli (META) Dan Matei (INP)

Revision History

Revision	Date	Author	Organisation	Description
V0.1	2013-12-6	Nikolaos Simou	NTUA	Outline
V0.2	2013-12-15	Nikolaos Simou	NTUA	First Draft
V0.3	2014-01-02	Sam H. Minelli	META	Document review
V0.4	2014-01-02	Dan Matei	INP	Document review
V0.5	2014-01-10	Eleni Tsalapati, Nikolaos Simou	NTUA	Final version
V1	2014-01-10	Maria Teresa Natale	ICCU	Formal Check

Project Coordinator: Istituto centrale per il catalogo unico delle biblioteche italiane
Address : Viale Castro Pretorio 105 – 00185 Roma
Phone number: +3906 06 49210 425
E-mail: info@athenaplus.eu
Project WEB site address: http://www.athenaplus.eu

Table of Contents

1 EXECUTIVE SUMMARY

The main objective of this deliverable is to describe the mapping methods from the Lightweight Information Describing Objects (LIDO) metadata model to Europeana Data Model (EDM). The mapping from LIDO to EDM is necessary for the project since LIDO is a widely used standard in the museum domain with which most of the AthenaPlus partners are already familiar, while EDM is the metadata standard that is currently used by Europeana for harvesting and presentation purposes and most of the AthenaPlus partners do not use it inside their institutions/organizations.

The mapping from one model to the other is not a trivial task. Despite the existence of an EDM harvesting schema –which may be misleading- EDM is an ontology the instances of which have to fulfil certain criteria, like being consistent to domain and range restrictions. On the other hand LIDO is based on an XSD having different validation rules for the instances. Thus, the first difficulty encountered was the transition from a tree structure data model such as LIDO to a graph – RDF structured model such as EDM. Furthermore, the full expressiveness of the two data models needs to be exploited as much as possible. LIDO uses some complex types/elements having semantics that either cannot be found in EDM (e.g. actorInRole) or are not currently supported (e.g. event).

There are two ways to map LIDO data into EDM. The first method requires the creation of an XSLT transforming a valid LIDO instance to a valid EDM instance -according to the EDM schema Europeana provides[1]. This is the current approach used in the AthenaPlus project thought the MINT mapping tool that provides a sufficient but not exhaustive (in terms of semantic expressiveness) solution. The second and most flexible / expressive way, which is currently under exploration, includes the development of an EDM – LIDO profile (i.e. an EDM-LIDO ontology) that could enable the partners to fully exploit the semantic expressiveness of both models.

[1] http://pro.europeana.eu/edm-documentation

2 INTRODUCTION

This deliverable describes the mapping method from the Lightweight Information Describing Objects (LIDO) metadata model to Europeana Data Model (EDM). LIDO is a widely used standard in the museum domain successfully used in other Europeana feeder projects like Athena[2], Linked Heritage[3], Judaica[4] and others. Therefore most of the AthenaPlus partners that have been probably involved in one of the aforementioned projects are already familiar with it.

On the other hand EDM is the metadata model that Europeana currently uses for harvesting and presentation purposes. Therefore the transformation of LIDO valid instances to EDM valid instances is very important for the project since its main objective is to deliver 3,6 million metadata records to Europeana.

The rest of the document is structured as follows: section 3 provides a small introduction to LIDO and EDM data models, then section 4 describes the two approaches that can be used to transform valid LIDO files to EDM. Section 5 provides the conclusions, also including the future activities. Finally, the appendixes provide additional information: In particular appendix 3 provides the current version of the LIDO to EDM XSLT; appendix 4 provides a LIDO sample created using MINT mapping tool and appendix 5 provides an EDM sample created using MINT mapping tool.

2.1 Background

The reader of this deliverable should be well familiar with both the LIDO and the EDM model. Therefore the interested reader is referred to the following documents:

- LIDO specification http://www.lido-schema.org/schema/v1.0/lido-v1.0-specification.pdf
- EDM primer http://pro.europeana.eu/documents/900548/770bdb58-c60e-4beb-a687-874639312ba5,
- EDM definition http://pro.europeana.eu/documents/900548/0d0f6ec3-1905-4c4f-96c8-1d817c03123c
- EDM mapping guidelines http://pro.europeana.eu/documents/900548/60777b88-35ed-4bae-8248-19c3696b81fb.

2.2 Role of this Deliverable in the Project

The main objective of the project is the delivery of 3.6 million metadata records to Europeana. Therefore the implementation of the LIDO to EDM mapping is vital for the project since the metadata need to be transformed to EDM. In other words the LIDO to EDM mappings ensures the achievement of the following milestones:

- **MS4** 10-20% content uploaded in MINT
- **MS5** 60-70% content uploaded in MINT
- **MS6** 100% content uploaded in MINT
- **MS7** MINT/AthenaPlus ready for delivering content to Europeana

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

[3] http://www.linkedheritage.eu/

[4] http://www.judaica-europeana.eu/

3 A short introduction to LIDO and EDM

During the last few years digital evolution of the Cultural Heritage field has accelerated rapidly, huge digital libraries and aggregation services are being built like the Europeana virtual library giving access to millions of books, paintings, films, museum objects and archival records that have been digitised throughout Europe. Other examples are the evolving Digital Public Library of America, or the various national portals feeding their content into Europeana: the Italian "CulturaItalia", the French "Moteur Collections", the Finnish "Culture Sampo", the German "Deutsche Digitale Bibliothek", to name just a few ones. All these services are collecting data and metadata about cultural objects, including museum objects.

LIDO harvesting schema (Coburn, E. et al, 2010) plays a major role in this digital evolution as it has been and is being used successfully in several EU-funded projects to aggregate cultural content mainly from museum collections for Europeana, and furthermore in various national, regional and thematic online services and research projects.

LIDO, being developed under the auspices of CIDOC, is the result of a collaborative effort of international stakeholders in the museum sector to create a common solution for contributing cultural heritage content to portals and other repositories of aggregated resources, as well as exposing, sharing and connecting data on the web. It is an application of the CIDOC Conceptual Reference Model (CRM) (Crofts, N. et al, 2010) and provides an explicit XML harvesting schema to deliver museum's object information in a standardized way.

The strength of LIDO lies in its ability to support the full range of descriptive information about museum objects. It can be used for all kinds of object, e.g. art, architecture, cultural history, history of technology, and natural history. Moreover it supports multilingual portal environments. LIDO allows for a cost-effective solution to supply museum object information originally stored in collections management systems and cataloguing databases with each one potentially being based on different descriptive metadata formats.

LIDO provides a much richer view of museum content compared to Dublin Core (DC) metadata format that is the most common format used in cultural heritage service environments and also forms the basis of the Europeana Semantic Elements (ESE).

In the museum community DC derived metadata schemas is not considered as appropriate: museum metadata is 'flatten out', with most of the data going into a limited subset of elements. For example, a number of different persons and institutions are usually associated with a museum object: the creator or finder of an object, important persons who have used it, the museum currently holding it, previous owners, and so on. All this qualified information is lost in a DC based format. Moreover, the lack of structure that allows elements to be grouped according to their semantic content leads to substantial information loss.

In contrast LIDO provides sufficiently detailed and well-defined semantics while integrating this information on a reasonable level for online services. Looking into current EU project statistics it can be estimated that around 5,5 million object descriptions are by now delivered to Europeana in LIDO format, increasing with running projects to more than 7,5 million items from several hundred institutions across more than 20 countries and languages. Europeana-related projects using LIDO are: ATHENA[5], MIMO[6], Judaica Europeana[7], Linked Heritage[8], Digitising Contemporary Art[9], Partage Plus[10], and EuropeanaPhotography[11]. This makes LIDO the secondly most used format in the Europeana environment after the DC based ESE format.

[5] http://www.athenaeurope.org/
[6] http://www.mimo-db.eu/
[7] http://www.judaica-europeana.eu/
[8] http://www.linkedheritage.eu/
[9] http://www.dca-project.eu/
[10] http://www.partage-plus.eu/
[11] http://www.europeana-photography.eu/

Europeana recently evolved from ESE to Europeana Data Model (EDM) (Isaac, A. et al 2010) that is a more expressive model based on ontology. The main reason for doing that is because ESE reduced the different heritage sectors represented in Europeana to the lowest common denominator. Hence by using EDM, Europeana attempts to reverse this reductive approach to transcend the respective information perspectives of the represented sectors – museums, archives, audio-visual repositories and libraries. EDM is not built on any particular community standard but rather adopts an open, cross-domain Semantic Web-based framework that can accommodate the range and richness of particular community standards such as LIDO for museums, EAD1 for archives or METS2 for digital libraries.

EDM attempts to not only support the full richness of the content providers' metadata but also to enable data enrichment from a range of third party sources. EDM also supports more complex objects than ESE is able to. In terms of a digitised book, the individual chapters, illustrations and index can be understood both individually and collectively; in terms of an archival finding aid or fonds, the constituent letters, deeds, manuscripts or other items can be similarly understood.

3.1 The AthenaPlus case

In the AthenaPlus project LIDO acts as a bridge between the providers' source metadata schemas and the Europeana Data Model (EDM). The consortium consists of many content providers who use their in-house metadata management systems and metadata formats, resulting in diversity of content. Therefore, in order to effectively manage and deliver different kinds of metadata to Europeana a common reference schema is required. In detail, the transformation to EDM is a two-stage process. First, the source metadata are transformed to LIDO, which will subsequently be transformed to EDM.

The reasons for this two step process are as follows:

- EDM is not fully mature yet. If significant investment were to be made, mapping all source metadata to EDM, all these mappings would need to be updated, each time EDM evolved. By mapping to LIDO that is a stable interim standard, the content providers are protected from any changes in EDM, and maintaining the ingestion process requires just the updating of the mapping from the interim standard to EDM.

- EDM is a relatively complex RDF based metadata model. While this ensures that EDM has the richness and capacity to capture the information from other rich and complex metadata standards, it also means that a great deal of expertise is needed to generate the best possible mapping to EDM. By limiting the AthenaPlus partners to work with a single mapping to EDM, a good quality of EDM metadata is ensured within the project.

- EDM, due to its relative newness, enjoys relatively low levels of expertise and experience within the Digital Cultural Heritage (DCH) community. LIDO is well documented and there is extensive experience available both within and outside of the project, to provide knowledge transfer, training and support.

4 LIDO to EDM mapping

There are two ways of making a mapping from LIDO to EDM. The first one is by using an XSLT for the transformation of a valid LIDO instance to a valid EDM instance -according to the EDM schema that Europeana provides, while the second way includes the development of an EDM – LIDO profile (i.e. an EDM-LIDO ontology) that would permit us to fully exploit the semantic expressiveness of both models.

4.1 LIDO to EDM mapping using XSLT

4.1.1 *Mapping description*

At this point it is important to present the way that EDM's basic structure has been satisfied. EDM is ontology and as such it defines classes and properties for describing a domain the objects/individuals of which must have an identifier. The two basic classes of EDM around which all the others are glued are the edm:ProvidedCHO and the ore:Aggregation that are related as shown in the figure below.

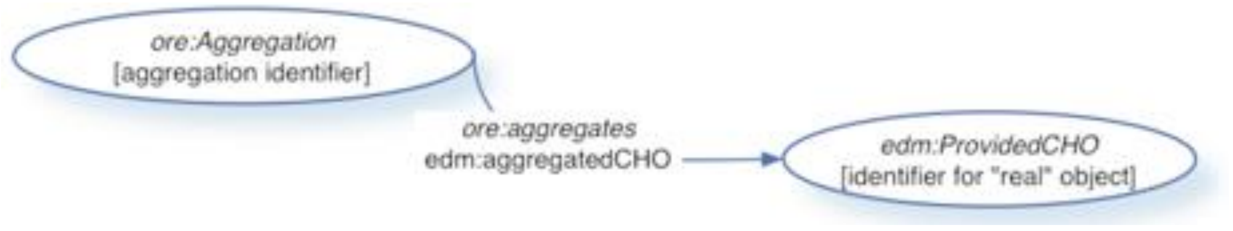

Figure 1: EDM basic structure

Therefore the very first thing that had to be addressed for the LIDO transformation to EDM was the creation of the identifiers that would permit the utilization of the EDM structure. In theory these identifiers have to be unique valid URIs – since only valid URIs can be used as identifiers in RDF. In the AthenaPlus case however this is not a major requirement since the EDM instances produced through the MINT mapping tool are harvested by Europeana that creates new URI identifiers. What is important however is to create unique identifiers among the metadata of a data provider because these identifiers are used by Europeana for detecting duplicates. For doing so the name of the provider along with the record local identifier **that has to be unique** within a content provider's metadata have been used. So the identifier of the ProvidedCHO is

http://mint-projects.image.ntua.gr/Athena_Plus/ProvidedCHO/+dataProvider+/ lido:recordID

that is also the value of the EDM xpath,

rdf:RDF/ore:Aggregation/edm:aggregatedCHO@rdf:resource

while the identifier of the respective instance of type ore:Aggregation is

http://mint-projects.image.ntua.gr/Athena_Plus/Aggregation/+dataProvider+/ lido:recordID

The following table presents the LIDO bookmarks that are used in the MINT mapping tool to assist providers during the mapping process, their LIDO xpaths and the EDM schema xpaths - in other words the properties that are attached to edm:ProvidedCHO and ore:Aggregation - to which they are mapped. The LIDO xpaths highlighted in bold green indicate some of the XSL conditions, while those in red specify the xpaths that cannot be mapped to EDM. The resulting XSLT is integrated into the MINT mapping tool, in that way the AthenaPlus content providers only have to make a valid LIDO instance using MINT and automatically a valid EDM instance will be made from it (see Appendix 3 for the full XSLT and appendixes 4 and 5 for a sample LIDO file and its transformation to EDM).

Table 1: LIDO to EDM mapping

LIDO Bookmark	LIDO Xpaths	EDM Xpaths
Identifier	/lido/administrativeMetadata/recordWrap/recordID	edm:ProvidedCHO/ dc:identifier
Descriptive metadata language	/lido/descriptiveMetadata/@ lang	All the values of the LIDO elements the belong to Descriptive Metadata are mapped with this language attribute in EDM.
Administrative metadata language	/lido/administrativeMetadata/@ lang	All the values of the LIDO elements the belong to Administrative Metadata are mapped with this language attribute in EDM.
Europeana type	/lido/descriptiveMetadata/objectClassificationWrap/ classificationWrap/classification**@lido:type='europeana:type'**/term	edm:ProvidedCHO/ edm:type
Object/Work Type	/lido/descriptiveMetadata/objectClassificationWrap/ objectWorkTypeWrap/lido:objectWorkType/term	edm:ProvidedCHO/ dc:type
Object Title/Name	/lido/descriptiveMetadata/objectIdentificationWrap/titleWrap/ titleSet/appellationValue**@lido:pref='preferred'**	edm:ProvidedCHO/ dc:title
	/lido/descriptiveMetadata/objectIdentificationWrap/titleWrap/ titleSet/appellationValue**@lido:pref='alternative'**	edm:ProvidedCHO/ dcterms:alternative
Object Description	/lido/descriptiveMetadata/objectIdentificationWrap/objectDescriptionWrap/ objectDescriptionSet/descriptiveNoteValue	edm:ProvidedCHO/ edm:description
Dimension	/lido/descriptiveMetadata/objectIdentificationWrap/objectMeasurementsWrap/ objectMeasurementsSet/displayObjectMeasurements	edm:ProvidedCHO/ dcterms:extent
Producer	/lido/descriptiveMetadata/eventWrap/eventSet/event/ eventActor/actorInRole/actor/nameActorSet/appellationValue **&** **/lido/descriptiveMetadata/eventWrap/eventSet/event/eventType/conceptID = http://terminology.lido-schema.org/lido00007**	edm:ProvidedCHO/ dc:creator
Producer's	/lido/descriptiveMetadata/eventWrap/eventSet/event/eventActor/	NOT MAPPED

role	actorInRole/roleActor **&** **/lido/descriptiveMetadata/eventWrap/eventSet/event/eventType/conceptID = http://terminology.lido-schema.org/lido00007**	
Production Date	/lido/descriptiveMetadata/eventWrap/eventSet/event/eventDate/displayDate **&** **/lido/descriptiveMetadata/eventWrap/eventSet/event/eventType/conceptID = http://terminology.lido-schema.org/lido00007**	edm:ProvidedCHO/dc:created
Production Place Name	/lido/descriptiveMetadata/eventWrap/eventSet/event/eventPlace/displayPlace **&** **/lido/descriptiveMetadata/eventWrap/eventSet/event/eventType/conceptID = http://terminology.lido-schema.org/lido00007**	edm:ProvidedCHO/dcterms:spatial
Material (Production Event)	/lido/descriptiveMetadata/eventWrap/eventSet/event/eventMaterialsTech/ materialsTech/termMaterialsTech@**lido:type = 'material'**/conceptID **&** **/lido/descriptiveMetadata/eventWrap/eventSet/event/eventType/conceptID = http://terminology.lido-schema.org/lido00007**	edm:ProvidedCHO/dcterms:medium
Technique (Production Event)	/lido/descriptiveMetadata/eventWrap/eventSet/event/eventMaterialsTech/ materialsTech/termMaterialsTech**@lido:type = 'technique'**/term **&** **/lido/descriptiveMetadata/eventWrap/eventSet/event/eventType/conceptID = http://terminology.lido-schema.org/lido00007**	edm:ProvidedCHO/dc:format
Subject / Theme (Concept)	/lido/descriptiveMetadata/objectRelationWrap/subjectWrap/subjectSet/ subject/subjectConcept/term	edm:ProvidedCHO/dc:subject
Repository	/lido/descriptiveMetadata/objectIdentificationWrap/repositoryWrap/ repositorySet**@lido:type ='current'**/repositoryName/ legalBodyName/appellationValue	edm:ProvidedCHO/dcterms:provenance
Rights Information for Work	/lido/administrativeMetadata/rightsWorkWrap/rightsWorkSet	edm:ProvidedCHO/dc:rights
Record Type	/lido/administrativeMetadata/recordWrap/recordType/term	edm:ProvidedCHO/dc:type
Record Source (Name)	/lido/administrativeMetadata/recordWrap/recordSource@**lido:type ='europeana:dataProvider'/** legalBodyName/appellationValue	ore:Aggregation/edm:dataProvider
Record Link (Link to metadata)	/lido/administrativeMetadata/recordWrap/recordInfoSet/recordInfoLink	ore:Aggregation/edm:isShownAt

Link to DCHO (Master)	/lido/administrativeMetadata/resourceWrap/resourceSet/ resourceRepresentation**@lido:type='image_master'**/linkResource**[1] (the first occurrence)**	ore:Aggregation/edm:isShownBy
	/lido/administrativeMetadata/resourceWrap/resourceSet/ resourceRepresentation**@lido:type='image_master'**/linkResource**[>1] (the remaining occurrences)**	ore:Aggregation/edm:hasView
Link to DCHO (Thumbnail)	/lido/administrativeMetadata/resourceWrap/resourceSet/ resourceRepresentation**@lido:type='image_thumb**/linkResource	ore:Aggregation/edm:object
Resource Rights Type	/lido/administrativeMetadata/resourceWrap/resourceSet/ rightsResource/rightsType/term@**lido:pref='preferred'**	ore:Aggregation/edm:rights

4.1.2 *Mapping choices*

The mapping approach followed was to strictly respect the actual specifications of both models, LIDO and EDM in order to avoid as far as possibly any noise in the resulting EDM. A rather small, however core subset of the LIDO elements is mapped to EDM (i.e. the elements that are mostly used in LIDO were used in our XSL mapping). Some LIDO elements are mapped without compromising the LIDO model (e.g. both lido:recordType and lido:objectWorkType of LIDO are mapped to property dc:type that is generic, this is also the case with lido:subjectPlace, lido:subjectActor that are mapped to dc:subject). Finally, some of the LIDO elements are not mapped to EDM (the complete set of lido xpaths that are mapped to EDM can be found at Appendix 3).

Creation of EDM resources vs. literals:

- If a resource in LIDO is identified by an http URI this URI is referenced by the EDM property and a respective EDM resource is created.
- Resources are created without any literals, dereferencing of URIs is left to Europeana mechanisms.
- If a resource in LIDO is identified by a name for the resource and not by an http URI, an EDM property with the literal value is created.
- For literal values, if both index and display names for a LIDO resource are given the display variant is preferred as literal value of the EDM property.

Specific information for (multiple) WebResources is lost.

For museum and particularly architectural objects often multiple (digital) representations of the object exist with relevant distinct information that is essential to understand the object. But it (currently) cannot be mapped and displayed in the portal as properties of edm:WebResource.

Therefore, if specific information such as perspectives, creation date, creator (e.g. photographer) are considered essential, the (digital) representation itself – typically a photograph- has to be provided as CHO.

Moreover, the edm:type classification does not hold when for one physical object distinct digital representations are provided to Europeana, e.g. a 2D and a 3D scan: Is edm:type="IMAGE" or "3D"?

Qualifying information for agents (dc:contributor), dates (dc:date), places (dcterms:spatial) is lost.

LIDO is an event-oriented model in that way the cultural object's history, particularly including relations to agents, dates, places, is represented as a series of events that are qualified by an event type. Especially in this context relevant information to understand the object cannot be mapped to EDM.

Let’s see for example a LIDO based example record in Europeana (ingested through Partage Plus project) http://europeana.eu/portal/record/2026116/Partage_Plus_ProvidedCHO_Bildarchiv_Foto_Marburg_t2_20064721.html

Date: 1928; 1935; 1938; 1945; 2006/2007 – it remains unrevealed to the user that this series of dates tells about a history of modifications, part additions, and restoration of the villa.
Contributor: Dürckheim-Montmartin, Friedrich von (Graf) – in this case the kind of contribution is not clear. (In this example, the Contributor is the commissioner.)

Materials and techniques used to produce the object are in museum documentation closely tied and therefore both mapped to dc:format, material to sub-property dcterms:medium.

However, this mapping stresses the specification of dc:format.

Current location of the CHO is mapped to dcterms:provenance

Since edm:currentLocation requires a resource, but (current) locations are often given in the metadata only as literals, this information is (following a request of Europeana ingestion team) mapped to dcterms:provenance. However, this is slightly misleading especially for architectural objects.

4.2 LIDO to EDM mapping using an EDM-LIDO profile

The Europeana Data Model provides constructs that support community standards of cultural heritage content. At the same time, it allows the representation of metadata of either object-centric or event-centric approach. Both CRM and LIDO follow the event-centric approach and therefore EDM can accommodate both initiatives. A representative example of the way that a LIDO record is mapped to EDM is shown below.

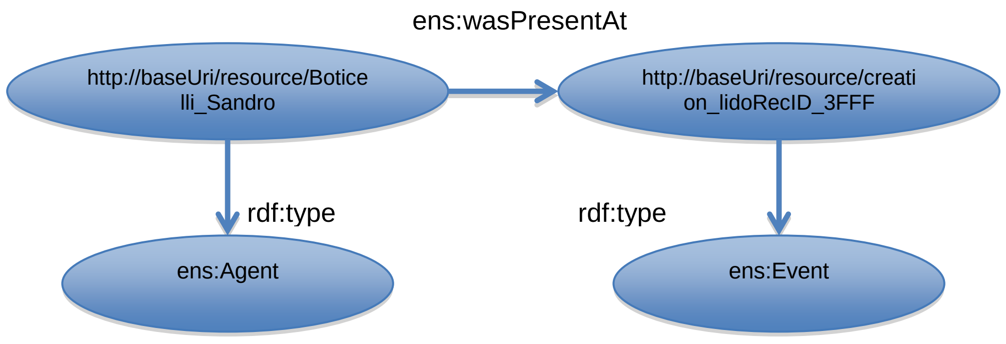
ens:wasPresentAt
http://baseUri/resource/Botticelli_Sandro
http://baseUri/resource/creation_lidoRecID_3FFF
rdf:type
rdf:type
ens:Agent
ens:Event

Figure 2: A LIDO record Representation in RDF using EDM

Likewise, a significant part of the knowledge covered in most LIDO records can be represented through EDM.

However, as the goal of EDM is to accommodate the range and richness of several community standards, not necessarily museum oriented, it constitutes a schema which is even more generic than CRM. Therefore the process of “strict” mapping from LIDO to EDM introduces several drawbacks.

An example that shows the bottlenecks rising from mapping LIDO to EDM is the representation of a relation’s type between a pair of objects. Any information about the relation’s type between objects or works can hardly be modelled with the use of EDM. Particularly, EDM defines specific kinds of relations, like “is successor of”, “is similar to”, “has part”, “incorporates”, between the described objects/works, or the general relation “ens:isRelatedTo”, omitting specific types of relation like “larger context for”, “model of”, “model for”, “study of”, “study for”, “rendering of”, “copy of” that LIDO anticipates as possible types. At the same time, EDM does not provide any specific construct to represent the type of the relation between two events.

Another issue that arose during the process of mapping LIDO records to EDM, was in our attempt to represent an actor's role participating in an event. Unless the actor is a creator of the work at hand, publisher or contributor, his/her role cannot be specified. In addition, for any personal information concerning the actor, for instance the date and place of birth, the occupation etc, EDM suggests the creation of an explicit link between the work under study and a carefully curated resource. This resource shall stand for the actor as a person and provide all necessary information about him/her, for instance a specific VIAF authority record.
A general observation about EDM is that it provides a strong structure to represent the required knowledge about the cultural heritage object and its correlation with the events that the object participated in, the actors participating in these events, the places that happened at, the time spans that occurred at and the representing resource of the object. However, it does not support directly the representation of the information provided for these resources (actor, place, time, digital representation, etc.) related to the cultural heritage object (CHO). In fact, as in the case of actor's personal information, EDM urges the ontology engineer to link these resources with the appropriate external resources.

In order to avoid losing any data during the RDFization process of the LIDO XML records, the creation of a LIDO ontology that contains the further specializations missing from CRM or EDM is currently under examination.

To achieve this we can introduce an appropriate set of LIDO concepts and properties. For instance, as the expressiveness of EDM does not suffice to represent the full range of types of relation between two objects/works we can define the set of possible sub-properties, such as "lido: is ReplicaOf", "lido: is Edition Of" of the EDM property "ens:is Related To", as they are defined from the LIDO profiles.

Similarly, a set of properties defining the type of relation between two events could also be introduced. The LIDO elements "roleActor", "attributionQualifierActor" and "extentActor" can again be represented by introducing an appropriate set of subproperties, such as "lido: executed", "lido: designed", of the EDM property "ens: was Present At". Concerning the personal information of the participating actor as it is provided by LIDO, a possible solution would be to create the respective properties or classes. For instance, the element "nationalityActor" could be represented with the property "lido:hasNationality", with domain the class "ens:Agent" and range a literal value or a resource. Concerning the modeling of the gender of the actor, we could introduce the classes lido:Male and lido:Female.Finally, the LIDO element "extentSubject" could be represented by introducing the property "lido:hasExtentSubject" as subproperty of dc:subject with domain the "ens:Proxy" class and range a literal value.

The above constitute only some suggestions for the development of the LIDO ontology. To this end further study has to be conducted.

5 CONCLUSIONS

This deliverable describes the procedure followed for the mapping of a valid LIDO file to a valid EDM file according to the EDM schema Europeana provides. The mapping from the one model to the other is quite complicated because of the different expressive capabilities of the two models and the different structure they follow - LIDO is a tree structure data model while EDM is an ontology and thus is a graph based / RDF structured model.

Two different ways of mapping a LIDO instance to an EDM instance have been explored. The one currently used relies on an XSLT for the transformation of a valid LIDO instance to a valid EDM instance -according to the EDM schema that Europeana provides. This approach can sufficiently map LIDO to EDM according to the requirements of the project and Europeana but it does not fully exploit the expressiveness of LIDO model. It's main limitations lie on the fact that EDM at the moment does not support events – despite the fact that they are included in its specification – and therefore the event related information that LIDO can express is either downgraded or lost. In addition at some points LIDO is more expressive and some of its elements cannot be directly mapped to EDM (e.g. roleActor).

Therefore for fully exploiting the semantic expressiveness of both models a most flexible / expressive way is currently under investigation that includes the development of an EDM – LIDO profile (i.e. an EDM-LIDO ontology).

6 APPENDIX 1: REFERENCES

[1] Coburn, E. & Light, R. & McKenna, G. & Stein, R. & Vitzthum, A. (2010) **“LIDO – Lightweight Information Describing Objects Version 1.0”**. Available: http://www.lido-schema.org/schema/v1.0/lido-v1.0-specification.pdf

[2] Crofts, N. & Doerr, M. & Gill, T. & Stead, S. & Stiff, M. (2011) **“Definition of the CIDOC Conceptual Reference Model”**. Available: http://www.cidoc-crm.org/docs/cidoc_crm_version_5.0.4.pdf

[3] Isaac, A. (2010) “Europeana Data Model Primer”. Availabe: http://pro.europeana.eu/documents/900548/770bdb58-c60e-4beb-a687-874639312ba5

[4] Isaac, A. (2010) “Europeana Data Model Primer”. Availabe: http://pro.europeana.eu/documents/900548/770bdb58-c60e-4beb-a687-874639312ba5

[5] LIDO specification http://www.lido-schema.org/schema/v1.0/lido-v1.0-specification.pdf

[6] EDM Mapping guidelines for data providers: http://pro.europeana.eu/edm-documentation

[7] EDM definition http://pro.europeana.eu/documents/900548/0d0f6ec3-1905-4c4f-96c8-1d817c03123c

[8] EDM object templates and XML schema http://europeanalabs.eu/wiki/EDMMXMLSchema

[9] Recommendations for the representation of hierarchical objects in Europeana: http://pro.europeana.eu/documents/468623/4a6eb2ec-4cc6-48b1-8824-92a1e564a279

7 APPENDIX 2: DEFINITION OF TERMS AND ABBREVIATIONS

- EDM: Europeana Data Model
- LIDO: Lightweight Information Describing Objects
- ESE: Europeana Semantic Elements
- DC: Dublin Core

8 APPENDIX 3: LIDO to EDM XSLT

```
<?xml version="1.0" encoding="UTF-8" ?>
<xsl:stylesheet exclude-result-prefixes="lido xml" version="2.0"
  xmlns:crm="http://www.cidoc-crm.org/rdfs/cidoc_crm_v5.0.2_english_label.rdfs#"
  xmlns:dc="http://purl.org/dc/elements/1.1/"
  xmlns:dcterms="http://purl.org/dc/terms/"
  xmlns:edm="http://www.europeana.eu/schemas/edm/"
  xmlns:foaf="http://xmlns.com/foaf/0.1/"
  xmlns:lido="http://www.lido-schema.org"
  xmlns:ore="http://www.openarchives.org/ore/terms/"
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  xmlns:rdaGr2="http://rdvocab.info/ElementsGr2/"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:skos="http://www.w3.org/2004/02/skos/core#"
  xmlns:wgs84="http://www.w3.org/2003/01/geo/wgs84_pos#"
  xmlns:xalan="http://xml.apache.org/xalan"
  xmlns:xml="http://www.w3.org/XML/1998/namespace"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:output method="xml" encoding="UTF-8" standalone="yes" indent="yes"/>

<!-- version1_FINAL-PP-2013-10-22: lido-v1.0-to-edm-v5.2.4-transform-v1.1-FINAL-PP-2013-10-22.xsl
-->
<!--

xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxx
    xx    XSL Transform to convert LIDO XML data, according to http://www.lido-
schema.org/schema/v1.0/lido-v1.0.xsd,
    xx    into EDM RDF/XML, according to http://www.europeana.eu/schemas/edm/EDM.xsd (EDM
v5.2.4, 2013-07-14)
    xx
    xx    Prepared for Linked Heritage (http://www.linkedheritage.org/) / Athena Plus
(http://www.athenaplus.eu/) by
        xx  Nikolaos Simou, National Technical University of Athens - NTUA
(nsimou@image.ntua.gr)
        xx  Regine Stein, Bildarchiv Foto Marburg, Philipps-Universitaet Marburg - UNIMAR
(r.stein@fotomarburg.de)
        xx
        xx  Notes:
        xx  • If a resource in LIDO is identified by an http URI this URI is referenced by the EDM
property and a respective EDM resource is created
        xx  • If, additionally or solely, a resource in LIDO is identified by a name for the resource an
EDM property with the literal value is created
        xx  • If both index and display names for a LIDO resource are given the display variant is
preferred as literal value of the EDM property
        xx
        xx  • If more than a lido record exist in the LIDO file the result is a file with a single root
(<rdf:RDF>) and many providedCHOs and Aggregations
        xx  • lido:recordID and the dataProvider's name are used for the creation of dummy identifiers
for providedCHOs and Aggregations
        xx
        xx  • The following parameters need to be set: edm_provider - baseURL edm_providedCHO -
baseURL ore_Aggregation
        xx
        xx  *** version1_DRAFT-2013-09-30 ***
        xx  •  The order of properties is fixed according to EDM schema
        xx  •  Repository Location maps to dcterms:provenance instead of dcterms:spatial
        xx • Filter names from actors if the actorID is given (dc:contributor, dc:creator, dc:publisher,
dc:subject (Actor))
        xx • Filter names from places if placeID is given (dcterms:spatial, dc:subject (Place))
        xx • Filter labels from concepts if conceptID is given (dc:format, dc:subject (Concept),
dc:type, dcterms:medium)
        xx
        xx • edm:Agent, skos:Concept resources are created with prefLabel from Partage
vocabulary: xml:lang according to descLang, default is english
        xx   if URI does not dereference in Partage vocab resources are created without any literals
        xx • edm:Place resources are created without any literals (eg prefLabel)
        xx
        xx *** version1_FINAL-Core-2013-10-16 ***
        xx • Names from actors given by the providers are not filtered (dc:contributor, dc:creator,
dc:publisher, dc:subject (Actor))
        xx   if a partage plus vocab uri is given then two elements are created a resource and a literal
with prefLabel from Partage vocabulary
        xx    xml:lang according to descLang, default is english. If the term value is the sameAs the
vocab value then only one literal is created.
        xx • Names from places given by the providers are not filtered (dcterms:spatial, dc:subject
(Place))
        xx • Names from concepts given by the providers are not filtered (dc:format, dc:subject
(Concept), dc:type, dcterms:medium)
        xx   if a partage plus vocab uri is given then two elements are created a resource and a literal
with prefLabel from Partage vocabulary
        xx    xml:lang according to descLang, default is english. If the term value is the sameAs the
vocab value then only one literal is created.
        xx
        xx • edm:Agent, skos:Concept resources are created with prefLabel and altLabels broader
and narrower concepts from Partage vocabulary
        xx     if URI does not dereference in Partage vocab resources are created without any literals
        xx • edm:Place resources are created without any literals
        xx • edm:hasView and and edm:isShownBy corrections
        xx • lido:administrativeMetadata/lido:recordWrap/lido:recordID -> dc:identifier
        xx • dcterms:isPartOf is added
        xx          lido:relatedWorkRelType/lido:conceptID = 'http://purl.org/dc/terms/isPartOf and
        xx                              lido:relatedWork/lido:object/lido:objectNote              then
<dcterms:isPartOf>objectNote</dcterms:isPartOf>
        xx • dcterms:hasPart is added
        xx          lido:relatedWorkRelType/lido:conceptID = 'http://purl.org/dc/terms/hasPart and
        xx                              lido:relatedWork/lido:object/lido:objectNote              then
<dcterms:isPartOf>objectNote</dcterms:hasPart>
        xx
        xx *** version1.1_FINAL-Core-2013-10-16 ***
        xx • Mapping of lido:resourceSet -> resourceSource and resourceDescription to
edm:providedCHO -> dc: description deleted: it doesn't make sense
        xx    with the new feature of mulitple views (edm:hasView).(It would be nice if Europeana
implemented at least the edm:WebResource->dc:description
        xx   property and displayed this information with each of the WebResource.)
        xx • dc:contributor, dc:creator, dc:publisher and dc:subject (Actor) and to create the literal
from the metadata only if no dereferencing of the URI is possible.

        xx
        xx

    xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxx
-->

 <xsl:variable name="var0">
   <item>TEXT</item>
```

```
    <item>VIDEO</item>
    <item>IMAGE</item>
    <item>SOUND</item>
    <item>3D</item>
  </xsl:variable>


  <!-- Specify edm:provider -->
  <xsl:param name="edm_provider" select="'Athena_Plus'" />
  <!-- Specify edm:providedCHO baseURL -->
  <xsl:param name="edm_providedCHO" select="'http://mint-
projects.image.ntua.gr/Athena_Plus/ProvidedCHO/'" />
  <!-- Specify ore:Aggregation baseURL -->
  <xsl:param name="ore_Aggregation" select="'http://mint-
projects.image.ntua.gr/Athena_Plus/Aggregation/'" />

  <xsl:function name="lido:langSelect">
    <xsl:param name="metadataLang"/>
    <xsl:param name="localLang"/>
        <xsl:if test="( string-length($metadataLang) &lt; 4 and string-length($metadataLang) &gt; 0) or
                                (string-length($localLang) &lt; 4 and string-length($localLang) &gt; 0)">
                        <xsl:choose>
                        <xsl:when test="(string-length($localLang) &lt; 4 and string-length($localLang) &gt; 0)">
                                <xsl:value-of select="$localLang"/>
                            </xsl:when>
                            <xsl:when test="( string-length($metadataLang) &lt; 4 and string-
length($metadataLang) &gt; 0)">
                                <xsl:value-of select="$metadataLang"/>
                            </xsl:when>
                            <xsl:otherwise></xsl:otherwise>
                        </xsl:choose>
        </xsl:if>
  </xsl:function>

  <xsl:template match="/">
     <rdf:RDF>

  <xsl:for-each select="/lido:lidoWrap/lido:lido | lido:lido">

    <xsl:variable name="descLang">
        <xsl:for-each select="lido:descriptiveMetadata/@xml:lang">
                        <xsl:value-of select="."/>
                </xsl:for-each>
    </xsl:variable>

     <xsl:variable name="adminLang">
        <xsl:for-each select="lido:administrativeMetadata/@xml:lang">
                        <xsl:value-of select="."/>
                </xsl:for-each>
    </xsl:variable>

    <xsl:variable name="dataProvider">
        <xsl:choose>
                        <xsl:when
test="lido:administrativeMetadata/lido:recordWrap/lido:recordSource[lido:type='europeana:dataProvider'
]">
                            <xsl:for-each
select="lido:administrativeMetadata/lido:recordWrap/lido:recordSource[lido:type='europeana:dataProvid
er']/lido:legalBodyName/lido:appellationValue">
                              <xsl:if test="position() = 1">
```

```
                                   <edm:dataProvider>
                                     <xsl:value-of select="."/>
                                   </edm:dataProvider>
                              </xsl:if>
                            </xsl:for-each>
                    </xsl:when>
                    <xsl:otherwise>
                            <xsl:for-each
select="lido:administrativeMetadata/lido:recordWrap/lido:recordSource/lido:legalBodyName/lido:appellat
ionValue">
                              <xsl:if test="position() = 1">
                                   <edm:dataProvider>
                                     <xsl:value-of select="."/>
                                   </edm:dataProvider>
                              </xsl:if>
                            </xsl:for-each>
                    </xsl:otherwise>
            </xsl:choose>
    </xsl:variable>

      <edm:ProvidedCHO>
        <xsl:attribute name="rdf:about">
          <xsl:for-each select="lido:administrativeMetadata/lido:recordWrap/lido:recordID">
            <xsl:if test="position() = 1">
              <xsl:value-of select="concat($edm_providedCHO, $dataProvider,'/',.)"/>
            </xsl:if>
          </xsl:for-each>
        </xsl:attribute>

        <!-- dc:contributor : lido:eventActor with lido:eventType NOT production or creation or designing or
publication -->
        <!-- dc:contributor Resource -->
        <xsl:for-each select="lido:descriptiveMetadata/lido:eventWrap/lido:eventSet/lido:event">
        <xsl:if test="not((lido:eventType/lido:conceptID = 'http://terminology.lido-schema.org/lido00007') or
(lido:eventType/lido:conceptID = 'http://terminology.lido-schema.org/lido00012') or
(lido:eventType/lido:conceptID = 'http://terminology.lido-schema.org/lido00224') or
(lido:eventType/lido:conceptID = 'http://terminology.lido-schema.org/lido00228'))">
            <xsl:for-each select="lido:eventActor">
            <xsl:if test="lido:actorInRole/lido:actor/lido:actorID[starts-with(., 'http://') or starts-with(.,
'https://')]">
                <dc:contributor>
                    <xsl:attribute name="rdf:resource">
                    <xsl:for-each select="lido:actorInRole/lido:actor/lido:actorID[starts-with(., 'http://') or starts-
with(., 'https://')]">
                            <xsl:if test="position() = 1">
                                        <xsl:value-of select="."/>
                                   </xsl:if>
                                   </xsl:for-each>
                                   </xsl:attribute>
                            </dc:contributor>
                    </xsl:if>
            </xsl:for-each>
        </xsl:if>
        </xsl:for-each>

             <!-- dc:contributor : lido:eventActor with lido:eventType NOT production or creation or
designing or publication -->
        <!-- dc:contributor Resource dereferenced to prefLabel-->
        <xsl:for-each select="lido:descriptiveMetadata/lido:eventWrap/lido:eventSet/lido:event">
        <xsl:if test="not((lido:eventType/lido:conceptID = 'http://terminology.lido-schema.org/lido00007') or
(lido:eventType/lido:conceptID           =           'http://terminology.lido-schema.org/lido00012')           or
(lido:eventType/lido:conceptID           =           'http://terminology.lido-schema.org/lido00224')           or
(lido:eventType/lido:conceptID = 'http://terminology.lido-schema.org/lido00228'))">
            <xsl:for-each select="lido:eventActor">
                <xsl:if test="lido:actorInRole/lido:actor/lido:actorID[starts-with(., 'http://partage.vocnet.org/')]">
                                <xsl:variable  name="partID"  select="lido:actorInRole/lido:actor/lido:actorID[1]/text()"
/>

                                <xsl:variable name="labelLang">
                                        <xsl:choose>
                                                <xsl:when
test="$PPActors/skos:Concept[@rdf:about=$partID]/skos:prefLabel[@xml:lang=$descLang]"><xsl:value
-of select="$descLang" /></xsl:when>
                                                <xsl:otherwise><xsl:value-of select="'en'" /></xsl:otherwise>
                                        </xsl:choose>
                                </xsl:variable>
                                <xsl:variable                                                    name="prefLabel"
select="$PPActors/skos:Concept[@rdf:about=$partID]/skos:prefLabel[@xml:lang=$labelLang]" />
                                <!--                              <xsl:if                   test                  =
"not(lido:actorInRole/lido:actor/lido:nameActorSet/lido:appellationValue[@lido:pref='preferred'           or
not(@lido:pref)])                                                                                             or
not(lido:actorInRole/lido:actor/lido:nameActorSet/lido:appellationValue[@lido:pref='preferred'           or
not(@lido:pref)]/text() = $prefLabel)">-->
                                        <dc:contributor>
                                                <xsl:attribute name="xml:lang" select="$labelLang" />
                                                <xsl:value-of select="$prefLabel" />
                                        </dc:contributor>
                                <!--</xsl:if>-->

                        </xsl:if>
                        </xsl:for-each>
                </xsl:if>
                </xsl:for-each>
        <!-- dc:contributor Resource dereferenced to prefLabel-->

        <!-- dc:contributor Literal-->
        <xsl:for-each select="lido:descriptiveMetadata/lido:eventWrap/lido:eventSet/lido:event">
        <xsl:if test="not((lido:eventType/lido:conceptID = 'http://terminology.lido-schema.org/lido00007') or
(lido:eventType/lido:conceptID           =           'http://terminology.lido-schema.org/lido00012')           or
(lido:eventType/lido:conceptID           =           'http://terminology.lido-schema.org/lido00224')           or
(lido:eventType/lido:conceptID = 'http://terminology.lido-schema.org/lido00228'))">
        <xsl:if   test="not(lido:eventActor/lido:actorInRole/lido:actor/lido:actorID)">   <!--   FILTER   ACTOR
NAME -->
                <xsl:for-each
select="lido:eventActor/lido:actorInRole/lido:actor/lido:nameActorSet/lido:appellationValue[@lido:pref='p
referred' or not(@lido:pref)]">
                                <dc:contributor>
                                        <xsl:if test="string-length( lido:langSelect($descLang,@xml:lang)) &gt; 0">
                                        <xsl:attribute name="xml:lang">
                                                <xsl:value-of select="lido:langSelect($descLang,@xml:lang)"/>
                                        </xsl:attribute>
                                </xsl:if>
                        <xsl:value-of select="."/>
                          </dc:contributor>
                </xsl:for-each>
        </xsl:if>
        </xsl:if>
        </xsl:for-each>
        <!-- dc:contributor Literal-->
```

```
	<!-- dc:contributor : lido:culture from any lido:eventSet -->
		<!-- dc:contributor Resource-->
	<xsl:for-each
select="lido:descriptiveMetadata/lido:eventWrap/lido:eventSet/lido:event/lido:culture/lido:conceptID[start
s-with(., 'http://') or starts-with(., 'https://')]">
	  <dc:contributor>
			<xsl:attribute name="rdf:resource">
				<xsl:value-of select="."/>
			</xsl:attribute>
	  </dc:contributor>
		</xsl:for-each>
		<!-- dc:contributor Resource-->

		<!-- dc:contributor : lido:culture from any lido:eventSet -->
	<!-- dc:contributor Resource dereferenced to prefLabel-->
	<xsl:for-each
select="lido:descriptiveMetadata/lido:eventWrap/lido:eventSet/lido:event/lido:culture/lido:conceptID[start
s-with(., 'http://partage.vocnet.org/')]">
			<xsl:variable name="partID" select="./text()" />

			<xsl:variable name="labelLang">
				<xsl:choose>
					<xsl:when
test="$PPActors/skos:Concept[@rdf:about=$partID]/skos:prefLabel[@xml:lang=$descLang]"><xsl:value
-of select="$descLang" /></xsl:when>
					<xsl:otherwise><xsl:value-of select="'en'" /></xsl:otherwise>
				</xsl:choose>
			</xsl:variable>
			<xsl:variable                                                    name="prefLabel"
select="$PPActors/skos:Concept[@rdf:about=$partID]/skos:prefLabel[@xml:lang=$labelLang]" />
			<!--<xsl:if test = "not(../lido:term) or not(../lido:term/text() = $prefLabel)">-->
				<dc:contributor>
					<xsl:attribute name="xml:lang" select="$labelLang" />
					<xsl:value-of select="$prefLabel" />
				</dc:contributor>
			<!--</xsl:if>-->

		</xsl:for-each>
	<!-- dc:contributor Resource dereferenced to prefLabel-->

	<!-- dc:contributor Literal-->
	<xsl:for-each
select="lido:descriptiveMetadata/lido:eventWrap/lido:eventSet/lido:event/lido:culture">
	<xsl:if test="not(lido:conceptID)"> <!-- FILTER ACTOR NAME -->
		  <xsl:for-each   select="lido:term[@lido:pref='preferred'   or   (not(@lido:pref)   and
not(@lido:addedSearchTerm='yes'))]">
			<dc:contributor>
		<xsl:if test="string-length( lido:langSelect($descLang,@xml:lang)) &gt; 0">
				<xsl:attribute name="xml:lang">
					<xsl:value-of select="lido:langSelect($descLang,@xml:lang)"/>
				</xsl:attribute>
			</xsl:if>
				<xsl:value-of select="."/>
			</dc:contributor>
		  </xsl:for-each>
		  </xsl:if>
	</xsl:for-each>
	<!-- dc:contributor Literal-->
```

```
        <!-- dc:coverage -->
        <!-- No LIDO property is mapped to dc:coverage of EDM -->

        <!-- dc:creator : lido:eventActor with lido:eventType = production or creation or designing-->
        <!-- dc:creator Resource-->
        <xsl:for-each select="lido:descriptiveMetadata/lido:eventWrap/lido:eventSet/lido:event">
        <xsl:if test="(lido:eventType/lido:conceptID = 'http://terminology.lido-schema.org/lido00007') or
(lido:eventType/lido:conceptID = 'http://terminology.lido-schema.org/lido00012') or
(lido:eventType/lido:conceptID = 'http://terminology.lido-schema.org/lido00224')">
            <xsl:for-each select="lido:eventActor">
                    <xsl:if test="lido:actorInRole/lido:actor/lido:actorID[starts-with(., 'http://') or starts-with(.,
'https://')]">
                <dc:creator>
                            <xsl:attribute name="rdf:resource">
                            <xsl:for-each select="lido:actorInRole/lido:actor/lido:actorID[starts-with(.,
'http://') or starts-with(., 'https://')]">
                            <xsl:if test="position() = 1">
                                <xsl:value-of select="."/>
                            </xsl:if>
                            </xsl:for-each>
                            </xsl:attribute>
                </dc:creator>
                    </xsl:if>
            </xsl:for-each>
        </xsl:if>
       </xsl:for-each>
        <!-- dc:creator Resource-->

        <!-- dc:creator : lido:eventActor with lido:eventType = production or creation or designing-->
        <!-- dc:creator Resource dereferenced to prefLabel-->
        <xsl:for-each select="lido:descriptiveMetadata/lido:eventWrap/lido:eventSet/lido:event">
        <xsl:if test="(lido:eventType/lido:conceptID = 'http://terminology.lido-schema.org/lido00007') or
(lido:eventType/lido:conceptID = 'http://terminology.lido-schema.org/lido00012') or
(lido:eventType/lido:conceptID = 'http://terminology.lido-schema.org/lido00224')">
            <xsl:for-each select="lido:eventActor">
              <xsl:if test="lido:actorInRole/lido:actor/lido:actorID[starts-with(., 'http://partage.vocnet.org/')]">
                        <xsl:variable name="partID" select="lido:actorInRole/lido:actor/lido:actorID[1]/text()"
/>

                        <xsl:variable name="labelLang">
                            <xsl:choose>
                                <xsl:when
test="$PPActors/skos:Concept[@rdf:about=$partID]/skos:prefLabel[@xml:lang=$descLang]"><xsl:value
-of select="$descLang" /></xsl:when>
                                <xsl:otherwise><xsl:value-of select="'en'" /></xsl:otherwise>
                            </xsl:choose>
                        </xsl:variable>
                        <xsl:variable name="prefLabel"
select="$PPActors/skos:Concept[@rdf:about=$partID]/skos:prefLabel[@xml:lang=$labelLang]" />
                        <!-- <xsl:if test =
"not(lido:actorInRole/lido:actor/lido:nameActorSet/lido:appellationValue[@lido:pref='preferred' or
not(@lido:pref)]) or
not(lido:actorInRole/lido:actor/lido:nameActorSet/lido:appellationValue[@lido:pref='preferred' or
not(@lido:pref)]/text() = $prefLabel)">-->
                            <dc:creator>
                                <xsl:attribute name="xml:lang" select="$labelLang" />
                                <xsl:value-of select="$prefLabel" />
                            </dc:creator>
                        <!--</xsl:if>-->
                    </xsl:if>
                    </xsl:for-each>
        </xsl:if>
       </xsl:for-each>
        <!-- dc:creator Resource dereferenced to prefLabel-->

            <!-- dc:creator Literal-->
        <xsl:for-each select="lido:descriptiveMetadata/lido:eventWrap/lido:eventSet/lido:event">
        <xsl:if test="(lido:eventType/lido:conceptID = 'http://terminology.lido-schema.org/lido00007') or
(lido:eventType/lido:conceptID = 'http://terminology.lido-schema.org/lido00012') or
(lido:eventType/lido:conceptID = 'http://terminology.lido-schema.org/lido00224')">
        <xsl:if test="not(lido:eventActor/lido:actorInRole/lido:actor/lido:actorID)"> <!-- FILTER ACTOR
NAME -->
        <xsl:for-each select="lido:eventActor[lido:displayActorInRole or
lido:actorInRole/lido:actor/lido:nameActorSet/lido:appellationValue[@lido:pref='preferred' or
not(@lido:pref)]]">
            <dc:creator>
                        <xsl:if test="string-length( lido:langSelect($descLang,@xml:lang)) &gt; 0">
                        <xsl:attribute name="xml:lang">
                              <xsl:value-of select="lido:langSelect($descLang,@xml:lang)"/>
                        </xsl:attribute>
                  </xsl:if>
                  <xsl:choose>
                              <xsl:when test="lido:displayActorInRole"><xsl:value-of
select="lido:displayActorInRole[1]/text()" /></xsl:when>
                              <xsl:otherwise><xsl:value-of
select="lido:actorInRole/lido:actor/lido:nameActorSet/lido:appellationValue[@lido:pref='preferred' or
not(@lido:pref)][1]/text()" /></xsl:otherwise>
                        </xsl:choose>
            </dc:creator>
          </xsl:for-each>
        </xsl:if>
        </xsl:if>
        </xsl:for-each>
          <!-- dc:creator Literal-->

          <!-- dc:date -->
          <!-- dc:date : lido:eventDate with lido:eventType NOT production or creation or designing -->
        <xsl:for-each select="lido:descriptiveMetadata/lido:eventWrap/lido:eventSet/lido:event">
        <xsl:if test="not((lido:eventType/lido:conceptID = 'http://terminology.lido-schema.org/lido00007') or
(lido:eventType/lido:conceptID = 'http://terminology.lido-schema.org/lido00012') or
(lido:eventType/lido:conceptID = 'http://terminology.lido-schema.org/lido00224'))">
        <xsl:for-each select="lido:eventDate">
        <xsl:if test="lido:date/lido:earliestDate | lido:displayDate">
            <dc:date>
                  <xsl:choose>
                        <xsl:when test="lido:date/lido:earliestDate = lido:date/lido:latestDate">
                              <xsl:value-of select="lido:date/lido:earliestDate"/>
                        </xsl:when>
                        <xsl:when test="lido:date/lido:earliestDate">
                              <xsl:value-of select="concat(lido:date/lido:earliestDate, '/',
lido:date/lido:latestDate)"/>
                        </xsl:when>
                        <xsl:otherwise>
                              <xsl:value-of select="lido:displayDate" />
                        </xsl:otherwise>
                  </xsl:choose>
            </dc:date>
            </xsl:if>
        </xsl:for-each>
        </xsl:if>
        </xsl:for-each>
        <!-- dc:date -->

                <!-- dc:description : lido:objectDescriptionSet -->
        <xsl:for-each
select="lido:descriptiveMetadata/lido:objectIdentificationWrap/lido:objectDescriptionWrap/lido:objectDes
criptionSet[lido:descriptiveNoteValue/string-length(.)&gt;0]">
                        <dc:description>
                                <xsl:if test="string-length( lido:langSelect($descLang,@xml:lang)) &gt; 0">
                                <xsl:attribute name="xml:lang">
                                        <xsl:value-of select="lido:langSelect($descLang,@xml:lang)"/>
                                </xsl:attribute>
                        </xsl:if>
                                <xsl:if test="@lido:type">
                                        <xsl:value-of select="concat(@lido:type, ': ')"/>
                                </xsl:if>
                                <xsl:for-each select="lido:descriptiveNoteValue">
                                        <xsl:value-of select="concat(., ' ')"/>
                                </xsl:for-each>
                                <xsl:if test="string-length(lido:sourceDescriptiveNote[1])&gt;0">
                                        <xsl:value-of select="concat(' (', lido:sourceDescriptiveNote[1], ')')" />
                                </xsl:if>
                                <xsl:if test="string-length(lido:descriptiveNoteID[1])&gt;0">
                                        <xsl:value-of select="concat(' (', lido:descriptiveNoteID[1], ')')" />
                                </xsl:if>
                        </dc:description>
                </xsl:for-each>
                <!-- dc:description -->

        <!-- dc:format : lido:eventMaterialsTech//lido:termMaterials NOT @lido:type=material -->
        <!-- dc:format Resource -->
        <xsl:for-each
select="lido:descriptiveMetadata/lido:eventWrap/lido:eventSet/lido:event/lido:eventMaterialsTech/lido:m
aterialsTech/lido:termMaterialsTech[not(lower-case(@lido:type)='material')]">
                        <xsl:for-each select="lido:conceptID[starts-with(., 'http://') or starts-with(., 'https://')]">
                                <dc:format>
                                        <xsl:attribute name="rdf:resource">
                                                <xsl:value-of select="."/>
                                        </xsl:attribute>
                 </dc:format>
                        </xsl:for-each>
        </xsl:for-each>
        <!-- dc:format Resource -->

        <!-- dc:format : lido:eventMaterialsTech//lido:termMaterials NOT @lido:type=material -->
        <!-- dc:format Resource dereferenced to prefLabel-->
        <xsl:for-each
select="lido:descriptiveMetadata/lido:eventWrap/lido:eventSet/lido:event/lido:eventMaterialsTech/lido:m
aterialsTech/lido:termMaterialsTech[not(lower-case(@lido:type)='material')]">
                        <xsl:for-each select="lido:conceptID[starts-with(., 'http://partage.vocnet.org/')]">
                                <xsl:variable name="partID" select="./text()" />
                                <xsl:variable name="labelLang">
                                        <xsl:choose>
                                                <xsl:when
test="$PPTerminology/skos:Concept[@rdf:about=$partID]/skos:prefLabel[@xml:lang=$descLang]"><xsl
:value-of select="$descLang" /></xsl:when>
                                                <xsl:otherwise><xsl:value-of select="'en'" /></xsl:otherwise>
                                        </xsl:choose>
                                </xsl:variable>
                        <xsl:variable                                                    name="prefLabel"
select="$PPTerminology/skos:Concept[@rdf:about=$partID]/skos:prefLabel[@xml:lang=$labelLang]" />
                        <xsl:if test = "not(../lido:term) or not(../lido:term/text() = $prefLabel)">
                        <dc:format>
                              <xsl:value-of select="$prefLabel" />
                        </dc:format>
                        </xsl:if>
                  </xsl:for-each>
      </xsl:for-each>
      <!-- dc:format Resource dereferenced to prefLabel -->

      <!-- dc:format Literal-->
      <xsl:for-each
select="lido:descriptiveMetadata/lido:eventWrap/lido:eventSet/lido:event/lido:eventMaterialsTech[lido:di
splayMaterialsTech                    or                    lido:materialsTech/lido:termMaterialsTech[not(lower-
case(@lido:type)='material')]]">
                  <xsl:choose>
                        <xsl:when test="lido:displayMaterialsTech">
                <dc:format>
                   <xsl:value-of select="lido:displayMaterialsTech[1]/text()" />
                </dc:format>
             </xsl:when>
                        <xsl:otherwise>
                              <xsl:if          test          ="lido:materialsTech/lido:termMaterialsTech[not(lower-
case(@lido:type)='material')]/lido:term[@lido:pref='preferred'          or          (not(@lido:pref)          and
not(@lido:addedSearchTerm='yes'))]">
                <dc:format>
                   <xsl:value-of                    select="lido:materialsTech/lido:termMaterialsTech[not(lower-
case(@lido:type)='material')]/lido:term[@lido:pref='preferred'          or          (not(@lido:pref)          and
not(@lido:addedSearchTerm='yes'))]" />
                </dc:format>
                </xsl:if>
             </xsl:otherwise>
                  </xsl:choose>
      </xsl:for-each>
      <!-- dc:format Literal-->

      <!-- dc:identifier : lido:objectPublishedID -->
      <xsl:for-each select="lido:objectPublishedID">
        <dc:identifier>
          <xsl:value-of select="."/>
        </dc:identifier>
      </xsl:for-each>
      <!-- dc:identifier -->

     <!-- dc:identifier : lido:workID -->
      <xsl:for-each
select="lido:descriptiveMetadata/lido:objectIdentificationWrap/lido:repositoryWrap/lido:repositorySet[not(
@lido:type='former')]/lido:workID">
        <dc:identifier>
          <xsl:value-of select="."/>
        </dc:identifier>
      </xsl:for-each>
      <!-- dc:identifier -->

      <!-- dc:identifier -->
      <xsl:for-each select="lido:administrativeMetadata/lido:recordWrap/lido:recordID">
            <dc:identifier>
            <xsl:value-of select="."/>
            </dc:identifier>
        </xsl:for-each>
        <!-- dc:identifier -->

            <!-- dc:language : lido:classification / @lido:type=language (MANDATORY with
edm:type=TEXT) -->
        <xsl:for-each
select="lido:descriptiveMetadata/lido:objectClassificationWrap/lido:classificationWrap/lido:classification[
@lido:type='language']/lido:term">
          <dc:language>
            <xsl:value-of select="."/>
          </dc:language>
        </xsl:for-each>
            <!-- dc:language -->

            <!-- dc:publisher : lido:eventActor with lido:eventType = publication -->
            <!-- dc:publisher Resource-->
            <xsl:for-each select="lido:descriptiveMetadata/lido:eventWrap/lido:eventSet/lido:event">
         <xsl:if test="(lido:eventType/lido:conceptID = 'http://terminology.lido-schema.org/lido00228')">
           <xsl:for-each select="lido:eventActor">
                    <xsl:if test="lido:actorInRole/lido:actor/lido:actorID[starts-with(., 'http://') or starts-with(.,
'https://')]">
                <dc:publisher>
                                    <xsl:attribute name="rdf:resource">
                                    <xsl:for-each select="lido:actorInRole/lido:actor/lido:actorID[starts-with(.,
'http://') or starts-with(., 'https://')]">
                                    <xsl:if test="position() = 1">
                                           <xsl:value-of select="."/>
                                    </xsl:if>
                                    </xsl:for-each>
                                    </xsl:attribute>
                </dc:publisher>
                    </xsl:if>
           </xsl:for-each>
        </xsl:if>
        </xsl:for-each>
            <!-- dc:publisher Resource-->

            <!-- dc:publisher : lido:eventActor with lido:eventType = publication -->
        <!-- dc:publisher Resource dereferenced to prefLabel-->
        <xsl:for-each select="lido:descriptiveMetadata/lido:eventWrap/lido:eventSet/lido:event">
         <xsl:if test="(lido:eventType/lido:conceptID = 'http://terminology.lido-schema.org/lido00228')">
         <xsl:for-each select="lido:eventActor">
            <xsl:if test="lido:actorInRole/lido:actor/lido:actorID[starts-with(., 'http://partage.vocnet.org/')]">
                        <xsl:variable name="partID" select="lido:actorInRole/lido:actor/lido:actorID[1]/text()"
/>

                        <xsl:variable name="labelLang">
                              <xsl:choose>
                                     <xsl:when
test="$PPActors/skos:Concept[@rdf:about=$partID]/skos:prefLabel[@xml:lang=$descLang]"><xsl:value
-of select="$descLang" /></xsl:when>
                                     <xsl:otherwise><xsl:value-of select="'en'" /></xsl:otherwise>
                              </xsl:choose>
                        </xsl:variable>
                        <xsl:variable name="prefLabel"
select="$PPActors/skos:Concept[@rdf:about=$partID]/skos:prefLabel[@xml:lang=$labelLang]" />
                        <!-- <xsl:if test =
"not(lido:actorInRole/lido:actor/lido:nameActorSet/lido:appellationValue[@lido:pref='preferred' or
not(@lido:pref)]) or
not(lido:actorInRole/lido:actor/lido:nameActorSet/lido:appellationValue[@lido:pref='preferred' or
not(@lido:pref)]/text() = $prefLabel)">-->
                                <dc:publisher>
                                    <xsl:attribute name="xml:lang" select="$labelLang" />
                                    <xsl:value-of select="$prefLabel" />
                                </dc:publisher>
                            <!--  </xsl:if>-->

                        </xsl:if>
                        </xsl:for-each>
    </xsl:if>
    </xsl:for-each>
    <!-- dc:publisher Resource dereferenced to prefLabel-->

        <!-- dc:publisher Literal-->
    <xsl:for-each select="lido:descriptiveMetadata/lido:eventWrap/lido:eventSet/lido:event">
    <xsl:if test="(lido:eventType/lido:conceptID = 'http://terminology.lido-schema.org/lido00228')">
    <xsl:if test="not(lido:eventActor/lido:actorInRole/lido:actor/lido:actorID)"> <!-- FILTER ACTOR
NAME -->
      <xsl:for-each select="lido:eventActor[lido:displayActorInRole or
lido:actorInRole/lido:actor/lido:nameActorSet/lido:appellationValue[@lido:pref='preferred' or
not(@lido:pref)]]">
        <dc:publisher>
                                <xsl:if test="string-length( lido:langSelect($descLang,@xml:lang)) &gt; 0">
                                <xsl:attribute name="xml:lang">
                                    <xsl:value-of select="lido:langSelect($descLang,@xml:lang)"/>
                                </xsl:attribute>
                            </xsl:if>
                            <xsl:choose>
                                    <xsl:when test="lido:displayActorInRole"><xsl:value-of
select="lido:displayActorInRole[1]/text()" /></xsl:when>
                                    <xsl:otherwise><xsl:value-of
select="lido:actorInRole/lido:actor/lido:nameActorSet/lido:appellationValue[@lido:pref='preferred' or
not(@lido:pref)][1]/text()" /></xsl:otherwise>
                                </xsl:choose>
        </dc:publisher>
      </xsl:for-each>
    </xsl:if>
    </xsl:if>
    </xsl:for-each>
        <!-- dc:publisher Literal-->

    <!-- dc:relation -->
    <!-- No LIDO property is mapped to dc:coverage of EDM -->

    <!-- dc:rights : lido:rightsWorkSet (rights for the CHO) -->
    <xsl:for-each select="lido:administrativeMetadata/lido:rightsWorkWrap/lido:rightsWorkSet">
          <xsl:choose>
                            <xsl:when test="lido:creditLine">
           <dc:rights>
              <xsl:value-of select="lido:creditLine"/>
           </dc:rights>
        </xsl:when>
                            <xsl:when
test="lido:rightsHolder/lido:legalBodyName/lido:appellationValue[@lido:pref='preferred' or
not(@lido:pref)]">
                            <xsl:for-each
select="lido:rightsHolder/lido:legalBodyName/lido:appellationValue[@lido:pref='preferred' or
not(@lido:pref)]">
                                <dc:rights>
```

```
                                        <xsl:value-of select="."/>
                                <dc:rights>
                        </xsl:for-each>
                        </xsl:when>
                </xsl:choose>
        </xsl:for-each>

        <!-- dc:source -->
        <!-- No LIDO property is mapped to dc:source of EDM -->

        <!-- dc:subject : lido:subjectConcept -->
                <!-- dc:subject Resource-->
        <xsl:for-each
select="lido:descriptiveMetadata/lido:objectRelationWrap/lido:subjectWrap/lido:subjectSet/lido:subject/li
do:subjectConcept/lido:conceptID[starts-with(., 'http://') or starts-with(., 'https://')]">
            <dc:subject>
                        <xsl:attribute name="rdf:resource">
                                <xsl:value-of select="."/>
                        </xsl:attribute>
            </dc:subject>
                </xsl:for-each>
        <!-- dc:subject Resource-->

        <!-- dc:subject : lido:subjectConcept -->
                <!-- dc:subject Resource dereferenced to prefLabel-->
        <xsl:for-each
select="lido:descriptiveMetadata/lido:objectRelationWrap/lido:subjectWrap/lido:subjectSet/lido:subject/li
do:subjectConcept/lido:conceptID[starts-with(., 'http://partage.vocnet.org/')]">
                        <xsl:variable name="partID" select="./text()" />

                        <xsl:variable name="labelLang">
                                <xsl:choose>
                                        <xsl:when
test="$PPTerminology/skos:Concept[@rdf:about=$partID]/skos:prefLabel[@xml:lang=$descLang]"><xsl
:value-of select="$descLang" /></xsl:when>
                                        <xsl:otherwise><xsl:value-of select="'en'" /></xsl:otherwise>
                                </xsl:choose>
                        </xsl:variable>
                        <xsl:variable                                          name="prefLabel"
select="$PPTerminology/skos:Concept[@rdf:about=$partID]/skos:prefLabel[@xml:lang=$labelLang]" />
                        <xsl:if test = "not(../lido:term) or not(../lido:term/text() = $prefLabel)">
                                <dc:subject>
                                        <xsl:attribute name="xml:lang" select="$labelLang" />
                                        <xsl:value-of select="$prefLabel" />
                                </dc:subject>
                        </xsl:if>
        </xsl:for-each>
                <!-- dc:subject Resource dereferenced to prefLabel-->

        <!-- dc:subject Literal-->
        <xsl:for-each
select="lido:descriptiveMetadata/lido:objectRelationWrap/lido:subjectWrap/lido:subjectSet/lido:subject/li
do:subjectConcept">
                <xsl:for-each    select="lido:term[@lido:pref='preferred'    or    (not(@lido:pref)    and
not(@lido:addedSearchTerm='yes'))]">
                        <dc:subject>
                                <xsl:if test="string-length( lido:langSelect($descLang,@xml:lang)) &gt; 0">
                                <xsl:attribute name="xml:lang">
                                        <xsl:value-of select="lido:langSelect($descLang,@xml:lang)"/>
                                </xsl:attribute>
                        </xsl:if>
                              <xsl:value-of select="."/>
                              <xsl:if test="not(position()=last())"><xsl:value-of select="' '" /></xsl:if>
                        </dc:subject>
                  </xsl:for-each>
      </xsl:for-each>
      <!-- dc:subject Literal-->

      <!-- dc:subject : lido:subjectActor -->
      <!-- dc:subject Resource-->
      <xsl:for-each
select="lido:descriptiveMetadata/lido:objectRelationWrap/lido:subjectWrap/lido:subjectSet/lido:subject/li
do:subjectActor/lido:actor/lido:actorID[starts-with(., 'http://') or starts-with(., 'https://')]">
         <dc:subject>
                        <xsl:attribute name="rdf:resource">
                              <xsl:value-of select="."/>
                        </xsl:attribute>
         </dc:subject>
            </xsl:for-each>
      <!-- dc:subject Resource-->

      <!-- dc:subject : lido:subjectActor -->
      <!-- dc:subject Resource dereferenced to prefLabel-->
      <xsl:for-each
select="lido:descriptiveMetadata/lido:objectRelationWrap/lido:subjectWrap/lido:subjectSet/lido:subject/li
do:subjectActor/lido:actor/lido:actorID[starts-with(., 'http://partage.vocnet.org/')]">
                        <xsl:variable name="partID" select="./text()" />

                        <xsl:variable name="labelLang">
                              <xsl:choose>
                                    <xsl:when
test="$PPActors/skos:Concept[@rdf:about=$partID]/skos:prefLabel[@xml:lang=$descLang]"><xsl:value
-of select="$descLang" /></xsl:when>
                                    <xsl:otherwise><xsl:value-of select="'en'" /></xsl:otherwise>
                              </xsl:choose>
                        </xsl:variable>
                        <xsl:variable                                          name="prefLabel"
select="$PPActors/skos:Concept[@rdf:about=$partID]/skos:prefLabel[@xml:lang=$labelLang]" />
                        <!--                          <xsl:if           test              =
"not(../lido:nameActorSet/lido:appellationValue[@lido:pref='preferred'    or    not(@lido:pref)])    or
not(../lido:nameActorSet/lido:appellationValue[@lido:pref='preferred'    or    not(@lido:pref)]/text()    =
$prefLabel)">-->
                              <dc:subject>
                                    <xsl:attribute name="xml:lang" select="$labelLang" />
                                    <xsl:value-of select="$prefLabel" />
                              </dc:subject>
                        <!-- </xsl:if>-->

            </xsl:for-each>
      <!-- dc:subject Resource dereferenced to prefLabel-->


      <!-- dc:subject Literal-->
       <xsl:for-each
select="lido:descriptiveMetadata/lido:objectRelationWrap/lido:subjectWrap/lido:subjectSet/lido:subject/li
do:subjectActor[lido:displayActor or lido:actor]">
            <xsl:choose>
                        <xsl:when test="lido:displayActor">
                              <dc:subject>
                                    <xsl:if test="string-length( lido:langSelect($descLang,@xml:lang)) &gt; 0">
```

```
                                        <xsl:attribute name="xml:lang">
                                            <xsl:value-of select="lido:langSelect($descLang,@xml:lang)"/>
                                        </xsl:attribute>
                                    </xsl:if>
                                        <xsl:value-of select="lido:displayActor[1]/text()" />
                                    </dc:subject>
                                </xsl:when>
                                <xsl:otherwise>
                                    <xsl:if test="not(lido:actor/lido:actorID)"> <!-- FILTER CONCEPT LABEL -->
                                        <dc:subject>
                                            <xsl:if    test="string-length(    lido:langSelect($descLang,@xml:lang))
&gt; 0">
                                            <xsl:attribute name="xml:lang">
                                                <xsl:value-of select="lido:langSelect($descLang,@xml:lang)"/>
                                            </xsl:attribute>
                                        </xsl:if>
                                            <xsl:value-of                                                select=
"lido:actor/lido:nameActorSet/lido:appellationValue[@lido:pref='preferred' or not(@lido:pref)][1]/text()" />
                                        </dc:subject>
                                    </xsl:if>
                                </xsl:otherwise>
                            </xsl:choose>
        </xsl:for-each>
        <!-- dc:subject Literal-->

        <!-- dc:subject : lido:subjectPlace -->
        <!-- dc:subject Resource-->
        <xsl:for-each
select="lido:descriptiveMetadata/lido:objectRelationWrap/lido:subjectWrap/lido:subjectSet/lido:subject/li
do:subjectPlace/lido:place/lido:placeID[starts-with(., 'http://') or starts-with(., 'https://')]">
          <dc:subject>
                                <xsl:attribute name="rdf:resource">
                                    <xsl:value-of select="."/>
                                </xsl:attribute>
          </dc:subject>
            </xsl:for-each>
        <!-- dc:subject Resource-->
        <!-- dc:subject Literal-->
        <xsl:for-each
select="lido:descriptiveMetadata/lido:objectRelationWrap/lido:subjectWrap/lido:subjectSet/lido:subject/li
do:subjectPlace[lido:displayPlace                                                                    or
lido:place/lido:namePlaceSet/lido:appellationValue[@lido:pref='preferred' or not(@lido:pref)]]">
                            <xsl:choose>
                                <xsl:when test="lido:displayPlace">
                                    <dc:subject>
                                        <xsl:if test="string-length( lido:langSelect($descLang,@xml:lang)) &gt; 0">
                                        <xsl:attribute name="xml:lang">
                                            <xsl:value-of select="lido:langSelect($descLang,@xml:lang)"/>
                                        </xsl:attribute>
                                    </xsl:if>
                                        <xsl:value-of select="lido:displayPlace[1]/text()" />
                                    </dc:subject>
                                </xsl:when>
                                <xsl:otherwise>
                                        <dc:subject>
                                            <xsl:if    test="string-length(    lido:langSelect($descLang,@xml:lang))
&gt; 0">
                                            <xsl:attribute name="xml:lang">
                                                <xsl:value-of select="lido:langSelect($descLang,@xml:lang)"/>
                                            </xsl:attribute>
```

```
                                  </xsl:if>
                                        <xsl:value-of                                                    select=
"lido:place/lido:namePlaceSet//lido:appellationValue[@lido:pref='preferred'  or  not(@lido:pref)][1]/text()"
/>
                                  </dc:subject>
                        </xsl:otherwise>
                    </xsl:choose>
    </xsl:for-each>
    <!-- dc:subject Literal-->

    <!-- dc:title : lido:titleSet / @lido:pref=preferred or empty -->
    <xsl:for-each
select="lido:descriptiveMetadata/lido:objectIdentificationWrap/lido:titleWrap/lido:titleSet/lido:appellationV
alue[@lido:pref='preferred' or not(@lido:pref)]">
      <dc:title>
          <xsl:if test="string-length( lido:langSelect($descLang,@xml:lang)) &gt; 0">
                              <xsl:attribute name="xml:lang">
                                  <xsl:value-of select="lido:langSelect($descLang,@xml:lang)"/>
                              </xsl:attribute>
                        </xsl:if>
        <xsl:value-of select="."/>
      </dc:title>
    </xsl:for-each>
    <!-- dc:title -->

    <!-- dc:type : lido:objectWorkType -->
          <!-- dc:type Resource-->
    <xsl:for-each
select="lido:descriptiveMetadata/lido:objectClassificationWrap/lido:objectWorkTypeWrap/lido:objectWor
kType/lido:conceptID[starts-with(., 'http://') or starts-with(., 'https://')]">
        <dc:type>
                        <xsl:attribute name="rdf:resource">
                              <xsl:value-of select="."/>
                        </xsl:attribute>
        </dc:type>
          </xsl:for-each>
          <!-- dc:type Resource-->

          <!-- dc:type Resource dereferenced to prefLabel-->
          <xsl:for-each
select="lido:descriptiveMetadata/lido:objectClassificationWrap/lido:objectWorkTypeWrap/lido:objectWor
kType/lido:conceptID[starts-with(., 'http://partage.vocnet.org/')]">
                        <xsl:variable name="partID" select="./text()" />
                        <xsl:variable name="labelLang">
                              <xsl:choose>
                                  <xsl:when
test="$PPTerminology/skos:Concept[@rdf:about=$partID]/skos:prefLabel[@xml:lang=$descLang]"><xsl
:value-of select="$descLang" /></xsl:when>
                                  <xsl:otherwise><xsl:value-of select="'en'" /></xsl:otherwise>
                              </xsl:choose>
                        </xsl:variable>
                        <xsl:variable                                                    name="prefLabel"
select="$PPTerminology/skos:Concept[@rdf:about=$partID]/skos:prefLabel[@xml:lang=$labelLang]" />
                        <xsl:if test = "not(../lido:term) or not(../lido:term/text() = $prefLabel)">
                              <dc:type>
                                  <xsl:attribute name="xml:lang" select="$labelLang" />
                                  <xsl:value-of select="$prefLabel" />
                              </dc:type>
                        </xsl:if>
    </xsl:for-each>
```

```
        <!-- dc:type Resource dereferenced to prefLabel-->

    <!-- dc:type Literal-->
    <xsl:for-each
select="lido:descriptiveMetadata/lido:objectClassificationWrap/lido:objectWorkTypeWrap/lido:objectWor
kType">
            <xsl:for-each   select="lido:term[@lido:pref='preferred'   or   (not(@lido:pref)   and
not(@lido:addedSearchTerm='yes'))]">
        <dc:type>
        <xsl:if test="string-length( lido:langSelect($descLang,@xml:lang)) &gt; 0">
                    <xsl:attribute name="xml:lang">
                        <xsl:value-of select="lido:langSelect($descLang,@xml:lang)"/>
                    </xsl:attribute>
                </xsl:if>
                    <xsl:value-of select="."/>
                </dc:type>
            </xsl:for-each>
    </xsl:for-each>
    <!-- dc:type Literal-->

        <!-- dc:type : lido:classification -->
    <!-- dc:type Resource-->
    <xsl:for-each
select="lido:descriptiveMetadata/lido:objectClassificationWrap/lido:classificationWrap/lido:classification/l
ido:conceptID[starts-with(., 'http://') or starts-with(., 'https://')]">
      <dc:type>
                <xsl:attribute name="rdf:resource">
                    <xsl:value-of select="."/>
                </xsl:attribute>
      </dc:type>
        </xsl:for-each>
        <!-- dc:type Resource-->

        <!-- dc:type Resource dereferenced to prefLabel-->
        <xsl:for-each
select="lido:descriptiveMetadata/lido:objectClassificationWrap/lido:classificationWrap/lido:classification/l
ido:conceptID[starts-with(., 'http://partage.vocnet.org/')]">
                <xsl:variable name="partID" select="./text()" />
                <xsl:variable name="labelLang">
                    <xsl:choose>
                        <xsl:when
test="$PPTerminology/skos:Concept[@rdf:about=$partID]/skos:prefLabel[@xml:lang=$descLang]"><xsl
:value-of select="$descLang" /></xsl:when>
                        <xsl:otherwise><xsl:value-of select="'en'" /></xsl:otherwise>
                    </xsl:choose>
                </xsl:variable>
                <xsl:variable                                              name="prefLabel"
select="$PPTerminology/skos:Concept[@rdf:about=$partID]/skos:prefLabel[@xml:lang=$labelLang]" />
                <xsl:if test = "not(../lido:term) or not(../lido:term/text() = $prefLabel)">
                    <dc:type>
                        <xsl:attribute name="xml:lang" select="$labelLang" />
                        <xsl:value-of select="$prefLabel" />
                    </dc:type>
                </xsl:if>
    </xsl:for-each>
        <!-- dc:type Resource dereferenced to prefLabel-->

    <!-- dc:type Literal-->
    <xsl:for-each
select="lido:descriptiveMetadata/lido:objectClassificationWrap/lido:classificationWrap/lido:classification[
not(@lido:type='language') and not (@lido:type='europeana:project') and not(lido:term[(. = 'IMAGE') or (.
= 'VIDEO') or (. = 'TEXT') or (. = '3D') or (. = 'SOUND')])]">

            <xsl:for-each select="lido:term[@lido:pref='preferred' or (not(@lido:pref) and
not(@lido:addedSearchTerm='yes'))] ">
              <dc:type>
                <xsl:if test="string-length( lido:langSelect($descLang,@xml:lang)) &gt; 0">
                    <xsl:attribute name="xml:lang">
                        <xsl:value-of select="lido:langSelect($descLang,@xml:lang)"/>
                    </xsl:attribute>
                </xsl:if>
          <xsl:value-of select="."/>
        </dc:type>
            </xsl:for-each>

    </xsl:for-each>
    <!-- dc:type Literal-->

        <!-- dcterms:alternative : lido:titleSet / @lido:pref=alternative  -->
        <xsl:for-each
select="lido:descriptiveMetadata/lido:objectIdentificationWrap/lido:titleWrap/lido:titleSet/lido:appellationV
alue[starts-with(@lido:pref, 'alternat')]">
     <dcterms:alternative>
        <xsl:if test="string-length( lido:langSelect($descLang,@xml:lang)) &gt; 0">
                    <xsl:attribute name="xml:lang">
                        <xsl:value-of select="lido:langSelect($descLang,@xml:lang)"/>
                    </xsl:attribute>
                </xsl:if>
      <xsl:value-of select="."/>
     </dcterms:alternative>
    </xsl:for-each>
    <!-- dcterms:alternative -->

        <!-- dct:conformsTo -->
        <!-- No LIDO property is mapped to dct:conformsTo of EDM -->

        <!-- dcterms:created : lido:eventDate with lido:eventType = production or creation or designing
-->
        <!-- dcterms:created -->
    <xsl:for-each select="lido:descriptiveMetadata/lido:eventWrap/lido:eventSet/lido:event">
    <xsl:if test="(lido:eventType/lido:conceptID = 'http://terminology.lido-schema.org/lido00007') or
(lido:eventType/lido:conceptID = 'http://terminology.lido-schema.org/lido00012') or
(lido:eventType/lido:conceptID = 'http://terminology.lido-schema.org/lido00224')">
    <xsl:for-each select="lido:eventDate">
      <xsl:if test="lido:date/lido:earliestDate | lido:displayDate">
      <dcterms:created>
                <xsl:choose>
                    <xsl:when test="lido:date/lido:earliestDate = lido:date/lido:latestDate">
                        <xsl:value-of select="lido:date/lido:earliestDate"/>
                    </xsl:when>
                    <xsl:when test="lido:date/lido:earliestDate">
                        <xsl:value-of select="concat(lido:date/lido:earliestDate, '/',
lido:date/lido:latestDate)"/>
                    </xsl:when>
                    <xsl:otherwise>
                        <xsl:value-of select="lido:displayDate" />
                    </xsl:otherwise>
                </xsl:choose>
      </dcterms:created>
      </xsl:if>
    </xsl:for-each>
    </xsl:if>
    </xsl:for-each>
    <!-- dcterms:created -->

    <!-- dcterms:extent : lido:objectMeasurementsSet -->
    <xsl:for-each
select="lido:descriptiveMetadata/lido:objectIdentificationWrap/lido:objectMeasurementsWrap/lido:object
MeasurementsSet[lido:displayObjectMeasurements or
lido:objectMeasurements/lido:measurementsSet]">
     <dcterms:extent>
          <xsl:choose>
               <xsl:when test="lido:displayObjectMeasurements"><xsl:value-of
select="lido:displayObjectMeasurements[1]/text()"/></xsl:when>
               <xsl:otherwise>
                    <xsl:for-each select="lido:objectMeasurements/lido:measurementsSet[string-
length(lido:measurementValue)&gt;0]">
                         <xsl:value-of select="concat(lido:measurementType, ': ',
lido:measurementValue, ' ', lido:measurementUnit)"/>
                         <xsl:for-each select="../lido:extentMeasurements[string-
length(.)&gt;0]"><xsl:value-of select="concat(' (', ., ')')" /></xsl:for-each>
                    </xsl:for-each>
               </xsl:otherwise>
          </xsl:choose>
     </dcterms:extent>
    </xsl:for-each>
    <!-- dcterms:extent -->

    <!-- No LIDO property is mapped to the following properties of EDM -->
    <!-- dcterms:hasFormat -->
    <!-- dcterms:hasPart -->
    <!-- dcterms:hasPart -->
    <xsl:for-each
select="lido:descriptiveMetadata/lido:objectRelationWrap/lido:relatedWorksWrap/lido:relatedWorkSet">
       <xsl:if test = "lido:relatedWorkRelType/lido:conceptID = 'http://purl.org/dc/terms/hasPart' and
                              (lido:relatedWork/lido:object/lido:objectWebResource or
lido:relatedWork/lido:object/lido:objectID or
                    lido:relatedWork/lido:object/lido:objectNote)">
      <xsl:for-each select = "lido:relatedWork/lido:object/lido:objectWebResource">
      <xsl:if test = "lido:relatedWork/lido:object/lido:objectWebResource[starts-with(.,'http')]">
          <dcterms:hasPart>
       <xsl:attribute name="rdf:resource">
               <xsl:value-of select =   "."/>
       </xsl:attribute>
          </dcterms:hasPart>
       </xsl:if>
       </xsl:for-each>
       <xsl:for-each select = "lido:relatedWork/lido:object/lido:objectID">
       <dcterms:hasPart>
               <xsl:value-of select =   "."/>
          </dcterms:hasPart>
       </xsl:for-each>
       <xsl:for-each select = "lido:relatedWork/lido:object/lido:objectNote">
       <dcterms:hasPart>
               <xsl:value-of select =   "."/>
          </dcterms:hasPart>
       </xsl:for-each>
       </xsl:if>
    </xsl:for-each>
    <!-- dcterms:hasVersion -->
```

```
        <!-- dcterms:isFormatOf -->
        <!-- dcterms:isPartOf -->
        <xsl:for-each
select="lido:descriptiveMetadata/lido:objectRelationWrap/lido:relatedWorksWrap/lido:relatedWorkSet">
                <xsl:if test = "lido:relatedWorkRelType/lido:conceptID = 'http://purl.org/dc/terms/isPartOf' and
                                        (lido:relatedWork/lido:object/lido:objectWebResource            or
lido:relatedWork/lido:object/lido:objectID or
                                lido:relatedWork/lido:object/lido:objectNote)">
            <xsl:for-each select = "lido:relatedWork/lido:object/lido:objectWebResource">
            <xsl:if test = "lido:relatedWork/lido:object/lido:objectWebResource[starts-with(.,'http')]">
                        <dcterms:isPartOf>
                <xsl:attribute name="rdf:resource">
                                <xsl:value-of select =     "."/>
                </xsl:attribute>
                        </dcterms:isPartOf>
                </xsl:if>
                </xsl:for-each>
                <xsl:for-each select = "lido:relatedWork/lido:object/lido:objectID">
                <dcterms:isPartOf>
                                <xsl:value-of select =     "."/>
                        </dcterms:isPartOf>
                </xsl:for-each>
                <xsl:for-each select = "lido:relatedWork/lido:object/lido:objectNote">
                <dcterms:isPartOf>
                                <xsl:value-of select =     "."/>
                        </dcterms:isPartOf>
                </xsl:for-each>
                </xsl:if>
        </xsl:for-each>
        <!-- dcterms:isReferencedBy -->
        <!-- dcterms:isReplacedBy -->
                <!-- dcterms:isReplacedBy -->
                <!-- dcterms:RequiredBy -->
                <!-- dcterms:issued -->
                <!-- dcterms:isVersionOf -->

        <!-- dcterms:medium : lido:eventMaterialsTech//lido:termMaterials / @lido:type=material -->
        <!-- dcterms:medium Resource-->
        <xsl:for-each
select="lido:descriptiveMetadata/lido:eventWrap/lido:eventSet/lido:event/lido:eventMaterialsTech/lido:m
aterialsTech/lido:termMaterialsTech[lower-case(@lido:type)='material']/lido:conceptID[starts-with(.,
'http://') or starts-with(., 'https://')]">
                <dcterms:medium>
                                <xsl:attribute name="rdf:resource">
                                        <xsl:value-of select="."/>
                                </xsl:attribute>
                </dcterms:medium>
        </xsl:for-each>
        <!-- dcterms:medium Resource-->

        <!-- dcterms:medium Resource dereferenced to prefLabel-->
                <xsl:for-each
select="lido:descriptiveMetadata/lido:eventWrap/lido:eventSet/lido:event/lido:eventMaterialsTech/lido:m
aterialsTech/lido:termMaterialsTech[lower-case(@lido:type)='material']/lido:conceptID[starts-with(.,
'http://partage.vocnet.org/')]">

                                <xsl:variable name="partID" select="./text()" />

                                <xsl:variable name="labelLang">
                                        <xsl:choose>
```

```
                              <xsl:when
test="$PPTerminology/skos:Concept[@rdf:about=$partID]/skos:prefLabel[@xml:lang=$descLang]"><xsl
:value-of select="$descLang" /></xsl:when>
                              <xsl:otherwise><xsl:value-of select="'en'" /></xsl:otherwise>
                         </xsl:choose>
                    </xsl:variable>
                    <xsl:variable                                              name="prefLabel"
select="$PPTerminology/skos:Concept[@rdf:about=$partID]/skos:prefLabel[@xml:lang=$labelLang]" />
                    <xsl:if test = "not(../lido:term) or not(../lido:term/text() = $prefLabel)">
                         <dcterms:medium>
                              <xsl:attribute name="xml:lang" select="$labelLang" />
                              <xsl:value-of select="$prefLabel" />
                         </dcterms:medium>
                    </xsl:if>

     </xsl:for-each>
          <!-- dcterms:medium Resource dereferenced to prefLabel-->

     <!-- dcterms:medium Literal (display -> dc:format, see above) -->
     <xsl:for-each
select="lido:descriptiveMetadata/lido:eventWrap/lido:eventSet/lido:event/lido:eventMaterialsTech/lido:m
aterialsTech/lido:termMaterialsTech[lower-case(@lido:type)='material']">

               <xsl:for-each    select="lido:term[@lido:pref='preferred'    or    (not(@lido:pref)    and
not(@lido:addedSearchTerm='yes'))]">
               <dcterms:medium>
                    <xsl:if test="string-length( lido:langSelect($descLang,@xml:lang)) &gt; 0">
                         <xsl:attribute name="xml:lang">
                              <xsl:value-of select="lido:langSelect($descLang,@xml:langg)"/>
                         </xsl:attribute>
                    </xsl:if>
                         <xsl:value-of select="."/>
                    </dcterms:medium>
               </xsl:for-each>

     </xsl:for-each>
     <!-- dcterms:medium Literal-->

     <!-- dcterms:provenance : lido:repositorySet -> lido:repositoryName and/or lido:repositoryLocation -
->
     <!-- First approach was:
                    IF        NOT        lido:repositoryName        THEN        dcterms:spatial        :
lido:repositorySet/lido:repositoryLocation
                    changed by request of Europeana
          -->
     <xsl:for-each
select="lido:descriptiveMetadata/lido:objectIdentificationWrap/lido:repositoryWrap/lido:repositorySet[@li
do:type='current']">
               <xsl:if
test="lido:repositoryName/lido:legalBodyName/lido:appellationValue[@lido:pref='preferred'            or
not(@lido:pref)]                                                                                     or
lido:repositoryLocation/lido:namePlaceSet/lido:appellationValue[@lido:pref='preferred'               or
not(@lido:pref)]">
                    <xsl:choose>
                         <xsl:when
test="lido:repositoryName/lido:legalBodyName/lido:appellationValue[@lido:pref='preferred'            or
not(@lido:pref)]                                                                                    and
lido:repositoryLocation/lido:namePlaceSet/lido:appellationValue[@lido:pref='preferred'               or
not(@lido:pref)]">
                              <dcterms:provenance>
```

```
                                        <xsl:value-of
select="concat(lido:repositoryName[1]/lido:legalBodyName[1]/lido:appellationValue[@lido:pref='preferre
d'                          or                          not(@lido:pref)][1],                           ',                           ',
lido:repositoryLocation[1]/lido:namePlaceSet[1]/lido:appellationValue[@lido:pref='preferred'               or
not(@lido:pref)][1])" />
                                    </dcterms:provenance>
                                    </xsl:when>
                                    <xsl:when
test="lido:repositoryName/lido:legalBodyName/lido:appellationValue[@lido:pref='preferred'               or
not(@lido:pref)]">
                                    <dcterms:provenance>
                                        <xsl:value-of
select="lido:repositoryName/lido:legalBodyName/lido:appellationValue[@lido:pref='preferred'             or
not(@lido:pref)][1]" />
                                    </dcterms:provenance>
                                    </xsl:when>
                                    <xsl:when
test="lido:repositoryLocation/lido:namePlaceSet/lido:appellationValue[@lido:pref='preferred'              or
not(@lido:pref)]">
                                    <dcterms:provenance>
                                        <xsl:value-of
select="lido:repositoryLocation/lido:namePlaceSet/lido:appellationValue[@lido:pref='preferred'           or
not(@lido:pref)][1]"/>
                                    </dcterms:provenance>
                                    </xsl:when>
                                </xsl:choose>
                            </xsl:if>
        </xsl:for-each>
        <!-- dcterms:provenance -->

        <!-- dct:references -->
        <!-- No LIDO property is mapped to dct:references of EDM -->

        <!-- dcterms:spatial : lido:eventPlace from any lido:eventSet -->
        <!-- dcterms:spatial Resource-->
        <xsl:for-each
select="lido:descriptiveMetadata/lido:eventWrap/lido:eventSet/lido:event/lido:eventPlace/lido:place/lido:
placeID[starts-with(., 'http://') or starts-with(., 'https://')]">
            <dcterms:spatial>
                                <xsl:attribute name="rdf:resource">
                                    <xsl:value-of select="."/>
                                </xsl:attribute>
                <dcterms:spatial>
        </xsl:for-each>
        <!-- dcterms:spatial Resource-->
        <!-- dcterms:spatial Literal-->
        <xsl:for-each
select="lido:descriptiveMetadata/lido:eventWrap/lido:eventSet/lido:event/lido:eventPlace[lido:displayPla
ce or lido:place/lido:namePlaceSet/lido:appellationValue]">

                <dcterms:spatial>
                                <xsl:choose>
                                    <xsl:when                               test="lido:displayPlace"><xsl:value-of
select="lido:displayPlace"/></xsl:when>
                                    <xsl:otherwise><xsl:value-of
select="lido:place/lido:namePlaceSet/lido:appellationValue[1]/text()" /></xsl:otherwise>
                                </xsl:choose>
                            </dcterms:spatial>

        </xsl:for-each>
```

```
    <!-- dcterms:spatial Literal-->

    <!-- No LIDO property is mapped to the following properties of EDM -->
    <!-- dct:tableOfContents -->
    <!-- dct:temporal -->
        <!-- edm:currentLocation -->
        <!-- edm:hasMet -->
        <!-- edm:hasType -->
        <!-- edm:incorporates -->
        <!-- edm:isDerivativeOf -->
        <!-- edm:isNextInSequence -->
        <!-- edm:isRelatedTo -->
        <!-- edm:isRepresentationOf -->
        <!-- edm:isSimilarTo -->
        <!-- edm:isSuccessorOf -->
        <!-- edm:realizes -->

    <!-- edm:type : lido:classification / @lido:type=europeana:type -->
    <xsl:if
test="(lido:descriptiveMetadata/lido:objectClassificationWrap/lido:classificationWrap/lido:classification/lid
o:term = 'IMAGE') or
(lido:descriptiveMetadata/lido:objectClassificationWrap/lido:classificationWrap/lido:classification/lido:ter
m = 'VIDEO') or
(lido:descriptiveMetadata/lido:objectClassificationWrap/lido:classificationWrap/lido:classification/lido:ter
m = 'TEXT') or
(lido:descriptiveMetadata/lido:objectClassificationWrap/lido:classificationWrap/lido:classification/lido:ter
m = '3D') or
(lido:descriptiveMetadata/lido:objectClassificationWrap/lido:classificationWrap/lido:classification/lido:ter
m = 'SOUND')">
      <xsl:for-each
select="lido:descriptiveMetadata/lido:objectClassificationWrap/lido:classificationWrap/lido:classification/l
ido:term[(. = 'IMAGE') or (. = 'VIDEO') or (. = 'TEXT') or (. = '3D') or (. = 'SOUND')]">
        <xsl:if test="position() = 1">
          <xsl:if test="index-of($var0/item, normalize-space()) > 0">
            <edm:type>
              <xsl:value-of select="."/>
            </edm:type>
          </xsl:if>
        </xsl:if>
      </xsl:for-each>
    </xsl:if>

    <!-- owl:sameAs -->
    <!-- No LIDO property is mapped to owl:sameAs of EDM -->
  </edm:ProvidedCHO>

  <!-- edm:WebResource : lido:resourceSet -->
  <xsl:for-each
select="lido:administrativeMetadata/lido:resourceWrap/lido:resourceSet[lido:resourceRepresentation/lid
o:linkResource]">
    <xsl:if test="lido:resourceRepresentation/lido:linkResource[starts-with(., 'http://') or starts-with(.,
'https://')]">
    <edm:WebResource>
        <xsl:attribute name="rdf:about">
        <xsl:for-each select="lido:resourceRepresentation/lido:linkResource[starts-with(., 'http://') or
starts-with(., 'https://')]">
            <xsl:if test="position() = 1">
              <xsl:value-of select="."/>
            </xsl:if>
        </xsl:for-each>
        </xsl:attribute>
                        <xsl:for-each select="lido:rightsResource">
                          <xsl:choose>
                                    <xsl:when test="lido:creditLine">
                                          <dc:rights>
                                                <xsl:value-of select="lido:creditLine"/>
                                          </dc:rights>
                                    </xsl:when>
                                    <xsl:when
test="lido:rightsHolder/lido:legalBodyName/lido:appellationValue[@lido:pref='preferred' or
not(@lido:pref)]">
                                    <xsl:for-each
select="lido:rightsHolder/lido:legalBodyName/lido:appellationValue[@lido:pref='preferred' or
not(@lido:pref)]">
                                          <dc:rights>
                                                <xsl:value-of select="."/>
                                           </dc:rights>
                                    </xsl:for-each>
                                    </xsl:when>
                              </xsl:choose>
                        </xsl:for-each>
      </edm:WebResource>
      </xsl:if>
    </xsl:for-each>
    <!-- edm:WebResource -->

        <!-- edm:WebResource : lido:recordInfoLink -->
    <xsl:for-each
select="lido:administrativeMetadata/lido:recordWrap/lido:recordInfoSet/lido:recordInfoLink[starts-with(.,
'http://') or starts-with(., 'https://')]">
      <edm:WebResource>
        <xsl:attribute name="rdf:about">
          <xsl:if test="position() = 1">
            <xsl:value-of select="."/>
          </xsl:if>
        </xsl:attribute>
      </edm:WebResource>
    </xsl:for-each>
    <!-- edm:WebResource -->

    <!-- edm:Agent : lido:eventActor or lido:subjectActor -->
    <xsl:for-each
select="lido:descriptiveMetadata/lido:eventWrap/lido:eventSet/lido:event/lido:eventActor/lido:actorInRole
|
lido:descriptiveMetadata/lido:objectRelationWrap/lido:subjectWrap/lido:subjectSet/lido:subject/lido:subje
ctActor">
     <xsl:if test="lido:actor/lido:actorID[starts-with(., 'http://') or starts-with(., 'https://')]">
      <edm:Agent>
        <xsl:attribute name="rdf:about">
          <xsl:for-each select="lido:actor/lido:actorID[starts-with(., 'http://') or starts-with(., 'https://')]">
            <xsl:if test="position() = 1">
              <xsl:value-of select="."/>
            </xsl:if>
          </xsl:for-each>
        </xsl:attribute>
        <!-- include prefLabel from Partage creator authority / xml:lang="en" -->
        <xsl:for-each select="lido:actor/lido:actorID[starts-with(., 'http://partage.vocnet.org')]">
                        <xsl:variable name="partID" select=".[1]/text()" />
          <xsl:if test="position() = 1">
          <xsl:for-each select="$PPActors/skos:Concept[@rdf:about=$partID]/skos:prefLabel">
```

```
                              <!-- prefLabel -->
                              <skos:prefLabel>
                    <xsl:attribute name="xml:lang" select="@xml:lang" />
                                      <xsl:value-of select="."/>
                              </skos:prefLabel>
                </xsl:for-each>
                <xsl:for-each select="$PPActors/skos:Concept[@rdf:about=$partID]/skos:altLabel">
                <!-- altLabel -->
                              <skos:altLabel>
                    <xsl:attribute name="xml:lang" select="@xml:lang" />
                                      <xsl:value-of select="."/>
                              </skos:altLabel>
                </xsl:for-each>
                              <!-- broader -->
                <xsl:for-each
select="$PPActors/skos:Concept[@rdf:about=$partID]/skos:broader/@rdf:resource">
                       <skos:broader>
                    <xsl:attribute name="rdf:resource" select="." />
                </skos:broader>
                              <!-- narrower -->
                </xsl:for-each>
                <xsl:for-each
select="$PPActors/skos:Concept[@rdf:about=$partID]/skos:narrower/@rdf:resource">
                       <skos:narrower>
                    <xsl:attribute name="rdf:resource" select="." />
                              </skos:narrower>
                              </xsl:for-each>
                              </xsl:if>
            </xsl:for-each>
        </edm:Agent>
        </xsl:if>
      </xsl:for-each>
      <!-- edm:Agent -->

      <!-- edm:Place : lido:eventPlace or lido:subjectPlace -->
      <xsl:for-each
select="lido:descriptiveMetadata/lido:eventWrap/lido:eventSet/lido:event/lido:eventPlace |
lido:descriptiveMetadata/lido:objectRelationWrap/lido:subjectWrap/lido:subjectSet/lido:subject/lido:subje
ctPlace">
      <xsl:if test="lido:place/lido:placeID[starts-with(., 'http://') or starts-with(., 'https://')]">
       <edm:Place>
            <xsl:attribute name="rdf:about">
              <xsl:for-each select="lido:place/lido:placeID[starts-with(., 'http://') or starts-with(., 'https://')]">
                <xsl:if test="position() = 1">
                  <xsl:value-of select="."/>
                </xsl:if>
              </xsl:for-each>
            </xsl:attribute>
        </edm:Place>
        </xsl:if>
      </xsl:for-each>
      <!-- edm:Place -->

      <!-- edm:TimeSpan -->
      <!-- No LIDO property is mapped to this EDM class -->

        <!-- skos:Concept : lido:objectWorkType or lido:classification or lido:termMaterialsTech or
lido:culture or lido:subjectConcept -->
      <xsl:for-each select="
```

```
lido:descriptiveMetadata/lido:objectClassificationWrap/lido:objectWorkTypeWrap/lido:objectWorkType |

lido:descriptiveMetadata/lido:objectClassificationWrap/lido:classificationWrap/lido:classification[not(start
s-with(@lido:type, 'europeana'))] |

lido:descriptiveMetadata/lido:eventWrap/lido:eventSet/lido:event/lido:eventMaterialsTech/lido:materialsT
ech/lido:termMaterialsTech |
                lido:descriptiveMetadata/lido:eventWrap/lido:eventSet/lido:event/lido:culture |

lido:descriptiveMetadata/lido:objectRelationWrap/lido:subjectWrap/lido:subjectSet/lido:subject/lido:subje
ctConcept
              ">
       <xsl:if test="lido:conceptID[starts-with(., 'http://') or starts-with(., 'https://')]">
       <skos:Concept>
          <xsl:attribute name="rdf:about">
            <xsl:for-each select="lido:conceptID[starts-with(., 'http://') or starts-with(., 'https://')]">
              <xsl:if test="position() = 1">
                <xsl:value-of select="."/>
              </xsl:if>
            </xsl:for-each>
          </xsl:attribute>
          <!-- include prefLabel from Partage vocabulary -->
          <xsl:for-each select="lido:conceptID[starts-with(., 'http://partage.vocnet.org')]">
                          <xsl:variable name="partID" select=".[1]/text()" />
              <xsl:if test="position() = 1">
              <!-- prefLabel -->
                        <xsl:for-each                                                              select
="$PPTerminology/skos:Concept[@rdf:about=$partID]/skos:prefLabel">
                          <skos:prefLabel>
                                 <xsl:attribute name="xml:lang" select="@xml:lang" />
                                 <xsl:value-of select="." />
                          </skos:prefLabel>
              </xsl:for-each>
              <!-- altLabel -->
              <xsl:for-each select="$PPTerminology/skos:Concept[@rdf:about=$partID]/skos:altLabel">
                     <skos:altLabel>
                 <xsl:attribute name="xml:lang" select="@xml:lang" />
                                 <xsl:value-of select="."/>
                          </skos:altLabel>
                          <!-- broader -->
              </xsl:for-each>
              <xsl:for-each
select="$PPTerminology/skos:Concept[@rdf:about=$partID]/skos:broader/@rdf:resource">
                     <skos:broader>
                 <xsl:attribute name="rdf:resource" select="." />
              </skos:broader>
                          <!-- narrower -->
              </xsl:for-each>
              <xsl:for-each
select="$PPTerminology/skos:Concept[@rdf:about=$partID]/skos:narrower/@rdf:resource">
                     <skos:narrower>
                 <xsl:attribute name="rdf:resource" select="." />
                          </skos:narrower>
              </xsl:for-each>
                          </xsl:if>
          </xsl:for-each>
       </skos:Concept>
       </xsl:if>
     </xsl:for-each>
```

```
   <!-- skos:Concept -->

   <!-- ore:Aggregation -->
   <ore:Aggregation>
     <xsl:attribute name="rdf:about">
       <xsl:for-each select="lido:administrativeMetadata/lido:recordWrap/lido:recordID">
         <xsl:if test="position() = 1">
           <xsl:value-of select="concat($ore_Aggregation, $dataProvider,'/',.)"/>
         </xsl:if>
       </xsl:for-each>
     </xsl:attribute>

     <!-- edm:aggregatedCHO -->
      <edm:aggregatedCHO>
       <xsl:attribute name="rdf:resource">
       <xsl:for-each select="lido:administrativeMetadata/lido:recordWrap/lido:recordID">
           <xsl:if test="position() = 1">
           <xsl:value-of select="concat($edm_providedCHO, $dataProvider,'/',.)"/>
           </xsl:if>
         </xsl:for-each>
       </xsl:attribute>
     </edm:aggregatedCHO>
     <!-- edm:aggregatedCHO -->

     <!-- edm:dataProvider : lido:recordSource -->
     <xsl:choose>
                <xsl:when
test="lido:administrativeMetadata/lido:recordWrap/lido:recordSource[lido:type='europeana:dataProvider'
]">
                    <xsl:for-each
select="lido:administrativeMetadata/lido:recordWrap/lido:recordSource[lido:type='europeana:dataProvid
er']/lido:legalBodyName/lido:appellationValue">
                     <xsl:if test="position() = 1">
                        <edm:dataProvider>
                          <xsl:value-of select="."/>
                        </edm:dataProvider>
                     </xsl:if>
                    </xsl:for-each>
                </xsl:when>
                <xsl:otherwise>
                    <xsl:for-each
select="lido:administrativeMetadata/lido:recordWrap/lido:recordSource/lido:legalBodyName/lido:appellat
ionValue">
                     <xsl:if test="position() = 1">
                        <edm:dataProvider>
                          <xsl:value-of select="."/>
                        </edm:dataProvider>
                     </xsl:if>
                    </xsl:for-each>
                </xsl:otherwise>
          </xsl:choose>
          <!-- edm:dataProvider -->

          <!-- edm:hasView : lido:resourceRepresentation / @lido:type=image_master or empty -->
     <xsl:for-each
select="lido:administrativeMetadata/lido:resourceWrap/lido:resourceSet/lido:resourceRepresentation[@l
ido:type='image_master' or not(@lido:type)]">
     <xsl:if test="position() > 1">
          <xsl:for-each select="lido:linkResource[starts-with(., 'http://') or starts-with(., 'https://')]">
                <edm:hasView>
```

```
                              <xsl:attribute name="rdf:resource">
                 <xsl:value-of select="."/>
                              </xsl:attribute>
                         </edm:hasView>
       </xsl:for-each>
       </xsl:if>
       </xsl:for-each>
       <!-- edm:hasView -->

               <!-- edm:isShownAt : lido:recordInfoLink -->
        <xsl:if
test="lido:administrativeMetadata/lido:recordWrap/lido:recordInfoSet/lido:recordInfoLink[starts-with(.,
'http://') or starts-with(., 'https://')]">
              <xsl:for-each
select="lido:administrativeMetadata/lido:recordWrap/lido:recordInfoSet/lido:recordInfoLink">
               <edm:isShownAt>
                   <xsl:attribute name="rdf:resource">
                      <xsl:if test="position() = 1">
                              <xsl:value-of select="."/>
                      </xsl:if>
                   </xsl:attribute>
               </edm:isShownAt>
              </xsl:for-each>
          </xsl:if>
               <!-- edm:isShownAt -->

               <!-- edm:isShownBy : lido:resourceRepresentation / @lido:type=image_master or empty -->
        <xsl:for-each
select="lido:administrativeMetadata/lido:resourceWrap/lido:resourceSet/lido:resourceRepresentation[@l
ido:type='image_master' or not(@lido:type)]">
        <xsl:if test="position() = 1">
               <xsl:for-each select="lido:linkResource[starts-with(., 'http://') or starts-with(., 'https://')]">
                         <edm:isShownBy>
                              <xsl:attribute name="rdf:resource">
                 <xsl:value-of select="."/>
                              </xsl:attribute>
                         </edm:isShownBy>
       </xsl:for-each>
          </xsl:if>
       </xsl:for-each>
       <!-- edm:isShownBy -->

       <!-- edm:object -->
        <xsl:for-each
select="lido:administrativeMetadata/lido:resourceWrap/lido:resourceSet/lido:resourceRepresentation[@l
ido:type='image_thumb']">
        <xsl:for-each select="lido:linkResource[starts-with(., 'http://') or starts-with(., 'https://') ]">
          <xsl:if test="position() = 1">
                         <edm:object>
                              <xsl:attribute name="rdf:resource">
                 <xsl:value-of select="."/>
                              </xsl:attribute>
                         </edm:object>
          </xsl:if>
       </xsl:for-each>
       </xsl:for-each>
       <!-- edm:object -->

       <!-- edm:provider -->
       <edm:provider>
```

```
        <xsl:value-of select="$edm_provider" />
      </edm:provider>
      <!-- edm:provider -->

      <!-- edm:rights : lido:rightsResource (MANDATORY, URI taken from a  set of URIs defined for use
in Europeana) -->
       <xsl:for-each
select="lido:administrativeMetadata/lido:resourceWrap/lido:resourceSet/lido:rightsResource/lido:rightsT
ype[lido:conceptID[starts-with(., 'http://creativecommons.org/') or starts-with(.,
'http://www.europeana.eu/rights')] or lido:term[starts-with(., 'http://creativecommons.org/') or starts-with(.,
'http://www.europeana.eu/rights')]]">
        <xsl:if test="position() = 1">
          <edm:rights>
             <xsl:attribute name="rdf:resource">
                                <xsl:choose>
                                        <xsl:when  test="lido:conceptID[starts-with(.,  'http://creativecommons.org/')
or starts-with(., 'http://www.europeana.eu/rights')]">
                                                <xsl:value-of select="lido:conceptID[starts-with(.,
'http://creativecommons.org/') or starts-with(., 'http://www.europeana.eu/rights')][1]/text()"/>
                                        </xsl:when>
                                        <xsl:when test="lido:term[starts-with(., 'http://creativecommons.org/') or
starts-with(., 'http://www.europeana.eu/rights')]">
                                                <xsl:value-of select="lido:term[starts-with(.,
'http://creativecommons.org/') or starts-with(., 'http://www.europeana.eu/rights')][1]/text()"/>
                                        </xsl:when>
                                </xsl:choose>
                     </xsl:attribute>
          </edm:rights>
        </xsl:if>
       </xsl:for-each>
       <!-- edm:rights -->
     </ore:Aggregation>
        <!-- ore:Aggregation -->
   </xsl:for-each>
  </rdf:RDF>
 </xsl:template>

<xsl:variable name="PPTerminology">
<skos:Concept rdf:about="http://partage.vocnet.org/part00036">
       <skos:prefLabel xml:lang="en">works on paper</skos:prefLabel>
       <skos:altLabel xml:lang="de">Kunst auf Papier</skos:altLabel>
       <skos:prefLabel xml:lang="sv">konst på papper</skos:prefLabel>
       <skos:prefLabel xml:lang="fi">paperityöt</skos:prefLabel>
       <skos:altLabel xml:lang="es">obras en papel</skos:altLabel>
       <skos:prefLabel xml:lang="es">obras sobre papel</skos:prefLabel>
       <skos:prefLabel xml:lang="sl">dela na papirju</skos:prefLabel>
       <skos:prefLabel xml:lang="pt">trabalho sobre papel</skos:prefLabel>
       <skos:prefLabel xml:lang="pl">prace na papierze</skos:prefLabel>
       <skos:prefLabel xml:lang="nl">werken op papier</skos:prefLabel>
       <skos:prefLabel xml:lang="it">oggetti di carta</skos:prefLabel>
       <skos:prefLabel xml:lang="cs">práce na papíře</skos:prefLabel>
       <skos:prefLabel xml:lang="ca">obra sobre paper</skos:prefLabel>
       <skos:prefLabel xml:lang="fr">œuvre sur papier</skos:prefLabel>
       <skos:prefLabel xml:lang="hu">papír alapú tárgy</skos:prefLabel>
       <skos:prefLabel xml:lang="no">arbeid på papir</skos:prefLabel>
       <skos:prefLabel xml:lang="hr">radovi na papiru</skos:prefLabel>
       <skos:prefLabel xml:lang="de">Werk auf Papier</skos:prefLabel>
       <skos:altLabel xml:lang="de">Arbeiten auf Papier (visuelle Werke)</skos:altLabel>
       <skos:altLabel xml:lang="de">Werke auf Papier</skos:altLabel>
       <skos:altLabel xml:lang="de">Graphik (visuelles Werk)</skos:altLabel>
```

```
        <skos:altLabel xml:lang="de">Grafik (visuelles Werk)</skos:altLabel>
        <skos:inScheme rdf:resource="http://partage.vocnet.org"/>
        <skos:broader rdf:resource="http://partage.vocnet.org/part01377"/>
        <skos:exactMatch rdf:resource="http://vocab.getty.edu/aat/300189621"/>
        <skos:closeMatch rdf:resource="http://d-nb.info/gnd/4395805-9"/>
        <skos:editorialNote xml:lang="en">PPObjectType</skos:editorialNote>
</skos:Concept>
</xsl:variable>

 <xsl:variable name="PPActors">
        <skos:ConceptScheme rdf:about="http://partage.vocnet.org/Actor">
                <rdf:type rdf:resource="http://www.w3.org/2004/02/skos/core#ConceptScheme"/>
        </skos:ConceptScheme>
</xsl:variable>
</xsl:stylesheet>
<!--end -->
```

9 APPENDIX 4: A LIDO Sample created using MINT mapping tool

```
<?xml version="1.0" encoding="UTF-8"?>
<lido:lidoWrap                        xmlns:lido="http://www.lido-schema.org"
xmlns:xalan="http://xml.apache.org/xalan">
  <lido:lido>
    <lido:lidoRecID
lido:type="AthenaPlus">/AthenaPlus:000000</lido:lidoRecID>
    <lido:descriptiveMetadata xml:lang="en">
      <lido:objectClassificationWrap>
        <lido:objectWorkTypeWrap>
          <lido:objectWorkType>
            <lido:term lido:addedSearchTerm="no">Photography</lido:term>
          </lido:objectWorkType>
        </lido:objectWorkTypeWrap>
        <lido:classificationWrap>
          <lido:classification lido:type="europeana:project">
            <lido:term lido:addedSearchTerm="no">Athena Plus</lido:term>
          </lido:classification>
          <lido:classification lido:type="europeana:type">
            <lido:term lido:addedSearchTerm="no">IMAGE</lido:term>
          </lido:classification>
        </lido:classificationWrap>
      </lido:objectClassificationWrap>
      <lido:objectIdentificationWrap>
        <lido:titleWrap>
          <lido:titleSet>
            <lido:appellationValue>The Parthenon</lido:appellationValue>
          </lido:titleSet>
        </lido:titleWrap>
        <lido:repositoryWrap>
          <lido:repositorySet lido:type="current">
            <lido:repositoryName>
              <lido:legalBodyName>
                <lido:appellationValue>IVML</lido:appellationValue>
              </lido:legalBodyName>
            </lido:repositoryName>
            <lido:repositoryLocation>
              <lido:namePlaceSet>
                <lido:appellationValue>Athens,
Greece</lido:appellationValue>
              </lido:namePlaceSet>
            </lido:repositoryLocation>
          </lido:repositorySet>
        </lido:repositoryWrap>
        <lido:objectMeasurementsWrap>
          <lido:objectMeasurementsSet>
            <lido:displayObjectMeasurements>400x323</lido:displayObjectMeasu
rements>
          </lido:objectMeasurementsSet>
        </lido:objectMeasurementsWrap>
      </lido:objectIdentificationWrap>
      <lido:eventWrap>
        <lido:eventSet>
          <lido:event>
            <lido:eventType>
              <lido:conceptID       lido:type="URI">http://terminology.lido-
schema.org/lido00007</lido:conceptID>
            </lido:eventType>
            <lido:eventActor>
```

```
              <lido:actorInRole>
                <lido:actor>
                  <lido:nameActorSet>
                    <lido:appellationValue>Petros Katsaros,
</lido:appellationValue>
                  </lido:nameActorSet>
                </lido:actor>
                <lido:roleActor>
                  <lido:term
lido:addedSearchTerm="no">Photographer</lido:term>
                </lido:roleActor>
              </lido:actorInRole>
            </lido:eventActor>
            <lido:eventActor>
              <lido:actorInRole>
                <lido:actor>
                  <lido:nameActorSet>
                    <lido:appellationValue>Gavril Papadopoulos,
</lido:appellationValue>
                  </lido:nameActorSet>
                </lido:actor>
                <lido:roleActor>
                  <lido:term               lido:addedSearchTerm="no">Lighting
Technician</lido:term>
                </lido:roleActor>
              </lido:actorInRole>
            </lido:eventActor>
            <lido:eventDate>
              <lido:date>
                <lido:earliestDate>2013-09-14</lido:earliestDate>
              </lido:date>
            </lido:eventDate>
            <lido:eventMaterialsTech>
              <lido:materialsTech>
                <lido:termMaterialsTech lido:type="technique">
                  <lido:term                lido:addedSearchTerm="no">Digital
Camera</lido:term>
                </lido:termMaterialsTech>
                <lido:termMaterialsTech lido:type="material">
                  <lido:conceptID
lido:type="URI">http://partage.vocnet.org/part00575</lido:conceptID>
                </lido:termMaterialsTech>
              </lido:materialsTech>
            </lido:eventMaterialsTech>
          </lido:event>
        </lido:eventSet>
      </lido:eventWrap>
      <lido:objectRelationWrap>
        <lido:subjectWrap>
          <lido:subjectSet>
            <lido:subject>
              <lido:subjectConcept>
                <lido:term                  lido:addedSearchTerm="no">Ancient
Greece</lido:term>
                <lido:term                      lido:addedSearchTerm="no">The
Parthenon</lido:term>
              </lido:subjectConcept>
            </lido:subject>
          </lido:subjectSet>
        </lido:subjectWrap>
      </lido:objectRelationWrap>
```

```
    </lido:descriptiveMetadata>
    <lido:administrativeMetadata xml:lang="en">
      <lido:rightsWorkWrap>
        <lido:rightsWorkSet>
          <lido:rightsHolder>
            <lido:legalBodyName>
              <lido:appellationValue>Ancient-
Greece.org</lido:appellationValue>
            </lido:legalBodyName>
          </lido:rightsHolder>
        </lido:rightsWorkSet>
      </lido:rightsWorkWrap>
      <lido:recordWrap>
        <lido:recordID lido:type="URI">0851b</lido:recordID>
        <lido:recordType>
          <lido:term lido:addedSearchTerm="no">Photography</lido:term>
        </lido:recordType>
        <lido:recordSource lido:type="europeana:dataProvider">
          <lido:legalBodyName>
            <lido:appellationValue>IVML</lido:appellationValue>
          </lido:legalBodyName>
        </lido:recordSource>
        <lido:recordInfoSet>
          <lido:recordInfoLink>http://www.image.ntua.gr/~nsimou/EuPhoto/Data
/108_0851b.xml</lido:recordInfoLink>
        </lido:recordInfoSet>
      </lido:recordWrap>
      <lido:resourceWrap>
        <lido:resourceSet>
          <lido:resourceRepresentation lido:type="image_thumb">
            <lido:linkResource>http://www.image.ntua.gr/~nsimou/EuPhoto/Imag
e/108_0851b.jpeg</lido:linkResource>
          </lido:resourceRepresentation>
          <lido:resourceRepresentation lido:type="image_master">
            <lido:linkResource>http://www.image.ntua.gr/~nsimou/EuPhoto/Imag
e/108_0851b.jpeg</lido:linkResource>
          </lido:resourceRepresentation>
          <lido:rightsResource>
            <lido:rightsType>
              <lido:term                            lido:addedSearchTerm="no"
lido:pref="preferred">http://www.europeana.eu/rights/rr-f/</lido:term>
            </lido:rightsType>
          </lido:rightsResource>
        </lido:resourceSet>
      </lido:resourceWrap>
    </lido:administrativeMetadata>
  </lido:lido>
</lido:lidoWrap>
```

10 APPENDIX 5: A EDM Sample created using MINT mapping tool

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<rdf:RDF xmlns:ore="http://www.openarchives.org/ore/terms/"
         xmlns:owl="http://www.w3.org/2002/07/owl#"
         xmlns:rdaGr2="http://rdvocab.info/ElementsGr2/"
         xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
         xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
         xmlns:crm="http://www.cidoc-
crm.org/rdfs/cidoc_crm_v5.0.2_english_label.rdfs#"
         xmlns:skos="http://www.w3.org/2004/02/skos/core#"
         xmlns:dc="http://purl.org/dc/elements/1.1/"
         xmlns:wgs84="http://www.w3.org/2003/01/geo/wgs84_pos#"
         xmlns:dcterms="http://purl.org/dc/terms/"
         xmlns:xalan="http://xml.apache.org/xalan"
         xmlns:edm="http://www.europeana.eu/schemas/edm/"
         xmlns:foaf="http://xmlns.com/foaf/0.1/">
   <edm:ProvidedCHO                                   rdf:about="http://mint-
projects.image.ntua.gr/Athena_Plus/ProvidedCHO/IVML/0851b">
      <dc:creator xml:lang="en">Petros Katsaros, </dc:creator>
      <dc:creator xml:lang="en">Gavril Papadopoulos, </dc:creator>
      <dc:format>Digital Camera</dc:format>
      <dc:rights>Ancient-Greece.org</dc:rights>
      <dc:subject xml:lang="en">Ancient Greece </dc:subject>
      <dc:subject xml:lang="en">The Parthenon</dc:subject>
      <dc:title xml:lang="en">The Parthenon</dc:title>
      <dc:type xml:lang="en">Photography</dc:type>
      <dc:type xml:lang="en">Athena Plus</dc:type>
      <dcterms:created>2013-09-14/</dcterms:created>
      <dcterms:extent>400x323</dcterms:extent>
      <dcterms:medium rdf:resource="http://partage.vocnet.org/part00575"/>
      <dcterms:provenance>IVML, Athens, Greece</dcterms:provenance>
      <edm:type>IMAGE</edm:type>
   </edm:ProvidedCHO>
   <edm:WebResource
rdf:about="http://www.image.ntua.gr/~nsimou/EuPhoto/Image/108_0851b.jpeg"/>
   <edm:WebResource
rdf:about="http://www.image.ntua.gr/~nsimou/EuPhoto/Data/108_0851b.xml"/>
   <skos:Concept rdf:about="http://partage.vocnet.org/part00575">
      <skos:prefLabel xml:lang="en">paper (fiber product)</skos:prefLabel>
   </skos:Concept>
   <ore:Aggregation                                   rdf:about="http://mint-
projects.image.ntua.gr/Athena_Plus/Aggregation/IVML/0851b">
      <edm:aggregatedCHO                           rdf:resource="http://mint-
projects.image.ntua.gr/Athena_Plus/ProvidedCHO/IVML/0851b"/>
      <edm:dataProvider>IVML</edm:dataProvider>
      <edm:isShownAt
rdf:resource="http://www.image.ntua.gr/~nsimou/EuPhoto/Data/108_0851b.xml"/>
      <edm:isShownBy
rdf:resource="http://www.image.ntua.gr/~nsimou/EuPhoto/Image/108_0851b.jpeg"
/>
      <edm:object
rdf:resource="http://www.image.ntua.gr/~nsimou/EuPhoto/Image/108_0851b.jpeg"
/>
      <edm:provider>Athena_Plus</edm:provider>
      <edm:rights rdf:resource="http://www.europeana.eu/rights/rr-f/"/>
   </ore:Aggregation>
</rdf:RDF>
```