Int J Digit Libr (2015) 15:87–101
DOI 10.1007/s00799-014-0136-9

PREMIS OWL

A semantic long-term preservation model

Sam Coppens · Ruben Verborgh · Sébastien Peyrard · Kevin Ford · Tom Creighton · Rebecca Guenther · Erik Mannens · Rik Van de Walle

Received: 6 December 2013 / Revised: 5 December 2014 / Accepted: 12 December 2014 / Published online: 11 January 2015

Abstract In this article, we present PREMIS OWL. This is a semantic formalisation of the PREMIS 2.2 data dictionary of the Library of Congress. PREMIS 2.2 are metadata implementation guidelines for digitally archiving information for the long term. Nowadays, the need for digital preservation is growing. A lot of the digital information produced merely a decade ago is in danger of getting lost as technologies are changing and getting obsolete. This also threatens a lot of information from heritage institutions. PREMIS OWL is a semantic long-term preservation schema. Preservation metadata are actually a mixture of provenance information, technical information on the digital objects to be preserved and rights information. PREMIS OWL is an OWL schema that can be used as data model supporting digital archives. It can be used for dissemination of the preservation metadata as Linked Open Data on the Web and, at the same time, for supporting semantic web technologies in the preservation processes. The model incorporates 24 preservation vocabularies, published by the LOC as SKOS vocabularies. Via these vocabularies, PREMIS descriptions from different institutions become highly interoperable. The schema is approved and now managed by the Library of Congress. The PREMIS OWL schema is published at http://www.loc.gov/premis/rdf/v1.

Keywords Linked open data · PREMIS OWL · Preservation · Metadata · Ontology · Semantic

Sam Coppens: This work was carried out while working at Ghent University-iMinds-Multimedia Lab.

S. Coppens (✉)
IBM Research, Smarter Cities Technology Centre,
Mulhuddart, Dublin 15, Ireland
e-mail: samcoppe@ie.ibm.com

R. Verborgh · E. Mannens · R. Van de Walle
iMinds, Multimedia Lab, Ghent University, Gaston Crommenlaan 8 bus 201, Ledeberg, 9050 Ghent, Belgium
e-mail: ruben.verborgh@ugent.be

E. Mannens
e-mail: erik.mannens@ugent.be

R. Van de Walle
e-mail: rik.walle@ugent.be

S. Peyrard
Bibliothque nationale de France, Paris, France
e-mail: sebastien.peyrard@bnf.fr

K. Ford · R. Guenther
Library of Congress, Washington, DC, USA
e-mail: kefo@loc.gov

R. Guenther
e-mail: rgue@loc.gov

T. Crieghton
FamilySearch, Salt Lake City, UT, USA
e-mail: CreightonNT@familysearch.org

1 Introduction

The need for digital long-term preservation is growing. A lot of material is still stored on analogue carriers which are degrading rapidly. Not only the material stored on analogue carriers is at danger, but also a lot of digital born material. In the digital world, the life cycle of file formats is very short and many file formats from one or two decades ago are not supported anymore by today's operating systems and browsers. A digital long-term preservation archive will have all the necessary processes in place to make sure that the information remains intact and interpretable.

The heart of such a digital archive is its data model. This data model needs to reflect and support all the preservation processes. Such a data model is called preservation meta-

data. Preservation metadata exist of provenance information, supplemented with technical information on the digital bitstreams, file formats and representations, rights information and structural information.

In this article, PREMIS OWL is introduced. PREMIS OWL is a semantic formalisation of the PREMIS 2.2 data dictionary [13]. The PREMIS 2.2 data dictionary defines a conceptual model (i.e., a list of terms) for modeling the preservation information of a digital archive. PREMIS is maintained by the Library of Congress. In fact, this model implements some of the best practices, stipulated by various preservation bodies, e.g., OAIS, the Open Archival Information System [3], and TDR, Trusted Digital Repositories: Attributes and Responsibilities [4] by the OCLC.[1] Until now, the formalisations of PREMIS were not very interoperable. Every archive has its own preservation policies and processes. To support this, there were a lot of free text fields in the PREMIS formalisations. PREMIS OWL solves this by integrating 24 preservation vocabularies of the LOC. These preservation vocabularies are modeled as SKOS [12] vocabularies.

First, in Sect. 2, we will give an overview of preservation information. Next, in Sect. 3, we introduce more specifically the PREMIS 2.2 Data Dictionary of the Library of Congress. This is a long-term preservation standard, for which a semantic binding is designed. Then, in Sect. 4, the ontology itself is discussed and explained in detail. In Sect. 5, we introduce shortly the preservation infrastructure, Archipel, for which the ontology originally was designed, followed with some related work on this in Sect. 7. The article ends with a conclusion in Sect. 8.

2 Preservation information

An Open Archival Information System has defined the responsibilities of a digital archive as follows:

> 'An Open Archival Information System (or OAIS) is an archive, consisting of an organization of people and systems, that has accepted the responsibility to preserve information and make it available for a designated community.'

This means that a digital archive is responsible for (1) storing the information for the long term and keeping it intact and (2) keeping the information interpretable for its designated community, i.e., the primary end-users of the digital archive. These two objectives are subjected to many risks, in which a digital archive needs to cover. A first step towards covering these risks is defining a preservation model that will support tackling these risks. Later on (Chapter 3), the infrastructure can be built to install the necessary processes to preserve the information for the long term. All the risks inherent to digital long-term preservation can be grouped into five categories spanning a long period of time. In chronological order, these risks are: interpretation of the file format, bit errors, file format changes, changing technologies, and, finally, organizational changes.

It is the responsibility of the archive not only to keep the archived information intact, but also interpretable over time. Today, there is a big discrepancy between the *short life-span of file formats* and the need for long-term preservation. File formats, and their different flavours, e.g., TIFF, GeoTIFF, and pyramid TIFF, emerge rapidly. A first risk, the archive has to deal with, is the right *interpretation of the file format*. A little further in time, file formats can become obsolete. The file formats supported by browsers and operating system can change (*file format changes*). The archive has then two solutions to present the stored information to the end user: migration or emulation. Metadata are needed to support each of these actions.

The archive also has to cope with *bit errors, bit rot, and bugs*. Bit rot is the gradual and natural decay of digital information and storage media over time, resulting in eventual unreadability. Bit rot affects different storage formats at different rates depending on the format's durability. Magnetic storage and optical discs are especially subject to varying forms of digital decay. For the time being, masked ROM cartridges appear to be fairly durable while EPROMs are at greater risk. This is why the risk of bit rot comes into play before the risks of file format obsolescence. The archive will need to have processes, and hence also the metadata, in place to correct these errors and to guarantee authenticity of the data. Examples of these are binary metadata, e.g., file format information, fixity information, e.g., MD5 checksums, and digital signatures.

As a next threat, the digital preservation platform has to deal with *technology changes*. Technology changes are less more frequently than file format obsolescence. Examples of these are for instance *Commodore 64* games, operating system incompatibilities, or even changing technologies regarding information storage, like relational databases, graph databases, etc. This puts specific demands on the architecture of the archive: it has to organise its data in a platform-independent manner. The OAIS reference model provides a high level architecture to deal with this issue.

In the long run, *institution structures*, terminologies, and the intended audience for your information might change, referred to as *organizational changes*. In practice, this means that your descriptive metadata can change, and the metadata format used for it. Other issues at the same level are the rights of an archived object or institution, which can change over time too. To keep the information interpretable, the archive needs: descriptive metadata, for a general description of the object, e.g., MARC; rights metadata, for describing copyright

[1] http://www.oclc.org/.

Fig. 1 Packages for a digital archive

statements, licenses, and possible grants that are given; and context metadata, for describing the relations of the content information to information from external data sources.

When developing a long-term preservation archive, all these described risks have to be taken into account. They have an impact on the architecture used for the archive, the data model used for the archive, and the processes it must have in place for keeping the information meaningful for the end-user. The OAIS reference model is a basis model for long-term archives. It assures platform-independent access to all archived information. In general, it describes the digital archive using three types of packages. Each package aggregates the needed metadata, according to its function, and the digital files accompanying the metadata. As shown in Fig. 1, these packages are:

- *Submission information package (SIP)* This is the package the archives accept for ingest. It consists of descriptive metadata, digital multimedia files, referenced by the metadata and some additional metadata, e.g., structural metadata or rights metadata.
- *Archival information package (AIP)* This is the package the archives use for preservation. It is in fact the SIP supplemented with preservation information.
- *Dissemination information package (DIP)* This is the package the archives offer for dissemination, e.g., if the rights do not permit to publish the archived digital files accompanying the metadata, this package will only contain metadata.

The data model for the archive will extend the SIP with preservation information to become an AIP and support the preservation processes of the archive. In general, this preservation information can be seen as an aggregation of four types of metadata:

- *Provenance information* this information will allow to describe the whole history of an object to be stored for the long term.
- *Technical information* the technical metadata will support processes to guarantee the data remain intact and interpretable. One of the processes will be the transcoding of file formats becoming obsolete. Technical information will support these actions.
- *Rights metadata* rights information is needed for the archive to know who can do which action on the stored information.
- *Structural information* objects to be stored can be supplied to the archive as an aggregation of different entities (for instance an HTML page or a book consisting of an ordered number of scanned TIFF images). When preserving these aggregated objects, its structure also needs to be preserved.

3 PREMIS 2.2

PREMIS [13] is a preservation standard based on the OAIS reference model. This standard was called the *Data Dictionary for Preservation Metadata: Final Report of the PREMIS working group*.[2] Next to the data dictionary, an XML schema that implements the data dictionary for digital preservation is also published. This preservation standard is described by a data model, which consists of five semantic units or classes important for digital preservation purposes:

- *Intellectual entity* a part of the content that can be considered as an intellectual unit for the management and the description of the content. This can be for example a book, a photo, or a database.
- *Object* a discrete unit of information in digital form, typically multimedia objects related to the intellectual entity.
- *Event* an action that has an impact on an object or an agent.
- *Agent* a person, institution, or software application that is related to an event of an object or is associated with the rights of an object.
- *Rights* description of one or more rights, permissions of an object or an agent.

Intellectual entities, events, and rights are directly related to an object, whereas an agent can only be related to an object through an event or through rights, as can be seen in Fig. 3. This way, not only the changes to an object are stored, but the event involved in this change is also described. These relationships offer the necessary tools to properly store the provenance of an archived object. The rights metadata needed for preservation are covered by the rights entity. Technical metadata and structural metadata are encapsulated in the PREMIS data dictionary via the description of the object entity.

If we relate this back to the reference model of Fig. 2, we notice both figures are very similar. The only difference is that the Entities to be preserved in Fig. 2 are split in PREMIS into Intellectual Entities and Objects. This is because a digital archive stores records or aggregations (collections) of

[2] http://www.loc.gov/standards/premis/v2/premis-2-2.pdf.

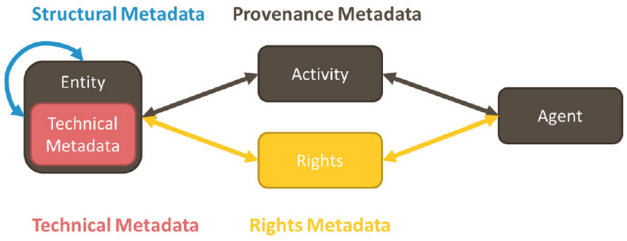

Fig. 2 Conceptual model of preservation metadata

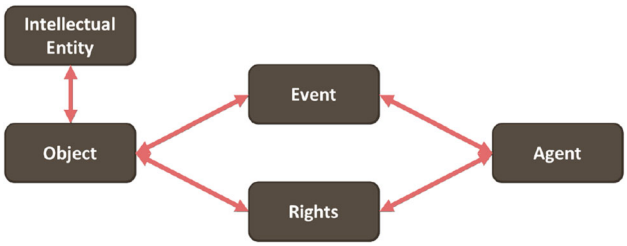

Fig. 3 Data model of the PREMIS 2.2 data dictionary

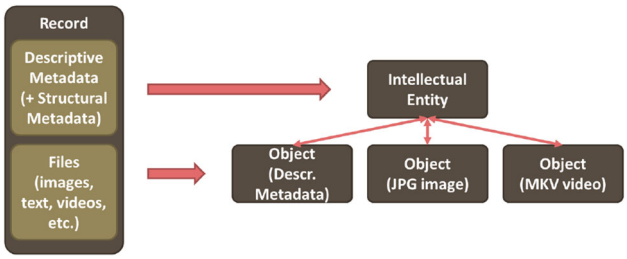

Fig. 4 Data model of the PREMIS 2.2 data dictionary

records. Such a record can be seen as a package of descriptive metadata and files that are referenced in the descriptive metadata, e.g., images or videos, as shown in Fig. 4. The Intellectual Entity of PREMIS is actually the descriptive metadata of the record and the Objects of PREMIS are the referenced files. Now, because the descriptive metadata (Intellectual Enties) can change also over time, they have also an Object equivalent, submitted to Rights and Events. Modeling the Intellectual Entity is not the purpose of PREMIS. PREMIS is concerned in modeling preservation metadata. The Intellectual Entity is descriptive metadata, for which every archive can choose its own appropriate model, e.g., Dublin Core, MARC, EAD, CDWA, etc.

4 PREMIS OWL

The PREMIS OWL ontology is published at http://id.loc.gov/ontologies/premis.rdf and its documentation can be found at http://id.loc.gov/ontologies/premis.html. The namespace for PREMIS OWL is http://www.loc.gov/premis/rdf/v1 and is often shortened to 'premisowl'. For the remainder of this chapter, this namespace will be considered the base URI. It needs to be stressed that PREMIS is a data model for the management of the archive, though on top of that, PREMIS can be used to disseminate the preservation information for instance as Linked Open Data (LOD, [1]), although it is not its primary concern. When designing the OWL ontology of the PREMIS 2.2, the choice was hence made to stick as closely as possible to the data dictionary of PREMIS 2.2, although it was not always possible or appropriate to do so in OWL. The reason to stick to the data dictionary is that the data dictionary of PREMIS 2.2 was developed by experts in the domain of long-term preservation, and every element has its own clearly defined semantics.

Why OWL? Looking at the data model, one can notice that it is dynamically relating the five entities to each other. Until now, an XML schema[3] was available that implemented the PREMIS 2.2 data dictionary. Implementing the data dictionary using the Web Ontology Language (OWL, [10]) allows us to relate the entities to each other in a more harmonious way, because RDF is resource based (and, as a consequence, every subject is identified by its URI). Another advantage of using OWL to implement the PREMIS 2.2 data dictionary, is that the relations can be made bidirectional using inverse properties. Using this semantic model of the data dictionary helps to keep the whole archive more consistent and to reuse information as much as possible. At the same time, this OWL model allows to easily integrate external information, a feature of data described in RDF, which allows to link to, e.g., technical file format information from an external registry. A last benefit of providing a semantic formalisation of PREMIS is that it can easily be extended to suit your archive's infrastructure and preservation processes. RDF descriptions can be easily extended with extra information, using the vocabulary that suits the institution's platform.

Thus, the main changes of the ontology to the data dictionary are first of all the formalisation (OWL). This formalisation has consequences on how the relationships between the PREMIS entities are constructed. Another consequence of using OWL for the PREMIS are the identifiers. The PREMIS Data Dictionary had a model to describe identifiers, because it is XML based and document based. In OWL and RDF, all the resources are identified by URIs. The links are directly made between the URIs of the objects. Despite this feature, PREMIS OWL remains to have a class for describing non-URI identifiers. The PREMIS Data Dictionary also had a special construct for describing extensions. In OWL, this class can be omitted, because OWL descriptions are by nature extensible. The fact that OWL (in fact RDF) uses URIs as identifiers and is extensible by default allows to refer to resources located outside the archive's metadata repository. In PREMIS OWL, this has consequences on the description

[3] http://www.loc.gov/standards/premis/premis.xsd.

of the formats of the bitstreams and files. In PREMIS OWL, one can directly reference URIs of format description from external format registries, such as UDFR, for the Object's format descriptions.[4]

Another big change in PREMIS OWL is the introduction of 24 preservation vocabularies that the Library of Congress published at http://id.loc.gov/preservationdescriptions/. For instance, there is a vocabulary for enumerating the event types. Event types are actions performed on digital objects within a preservation repository, e.g., migration (transcoding a one file format to another), virus check, or compression. Another example of one of the preservation vocabularies is a vocabulary for listing the storage medium of the preserved objects, e.g., hard disk, or magnetic tape. These vocabularies are formalised as SKOS vocabularies. This way, the ontology remains interoperable, while offering sufficient extensibility options to reflect the institutions preservation policies. For the remainder of this chapter, the namespace of the preservation vocabularies is shortened to 'idlc'.

In this section, the PREMIS OWL ontology is explained in detail and some design decisions made are discussed. For describing the PREMIS OWL ontology, we give first an overview of core PREMIS OWL classes and its structural information. Later on, we will expand on each of the five PREMIS entities, focusing on the ontology features. PREMIS OWL is a huge ontology, so we will not be able to describe all the formalisation details. For this, we refer to the ontology itself. There is also a workspace online[5] for people interested in discussing PREMIS OWL, contributing to PREMIS OWL, or just want some more practical guidelines for implementing PREMIS OWL in their institution's framework. In this workspace, one can also find some examples instances of PREMIS OWL. Next to the workspace, there was also an implementation fair given on PREMIS OWL during the iPres2013 conference.[6] The slides[7] of the workshop are publicly available and provide also some examples on the PREMIS OWL instances.

4.1 PREMIS OWL: core

For each of the five PREMIS entities, an OWL class was introduced, forming the core of PREMIS OWL. Thus, we have the following classes: *IntellectualEntity*, *Object*, *Event*, *RightsStatement*, and *Agent*. These classes are related to each other using the object properties *hasObject*, *hasEvent*, *hasAgent*, and *hasRightsStatement*. In Fig. 5, this model is depicted.

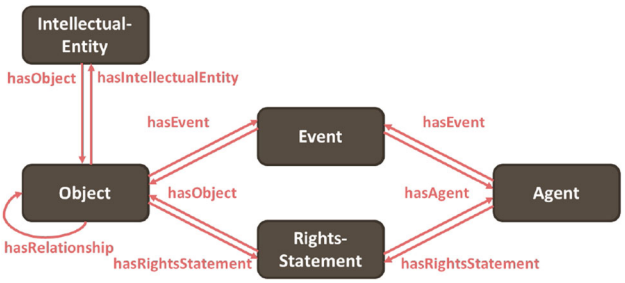

Fig. 5 Core classes and properties of PREMIS OWL

The relating properties are very general. Sometimes, one needs more specialised relationships to relate the entities, for instance to denote a certain role of the relating entity. For this reason, specific subproperties were created. These subproperties are then further detailed via idlc preservation vocabularies, which are formalised as separate SKOS vocabularies. The choice for introducing subproperties in the ontology for refining the PREMIS entities' relationships is that this way, they can be refined by SKOS vocabularies listing subproperties which also indicate the role of the linked entity in the relationship. OWL is not able to describe these ternary relationships. Adding further information on the relationship should be done using ternary relationships. These are relationships which give information on another relationship. OWL does not support ternary relationships. There are several ways to overcome this issue. One way is using subproperties which also encapsulate the further information, such as the role of the related object. Another way to do this is using reification. In reification, a relationship is modelled as a resource having at least three properties: one for denoting the subject, one for the property and one for the object. Extra information can then be added to the resource for refining the relationship. The choice was here made to use subproperties, first of all, because reification yields difficult querying and has performance impacts. A second reason for choosing subproperties is that one can create a separate SKOS vocabulary for each subproperty, which keeps things manageable. Later, we will describe how to introduce your own vocabularies into the ontology, which is in first place linked to the idlc vocabularies.

An *Object* can have several roles when linked to an event. For this reason, the *hasEventRelatedObject* property is introduced. The idlc:eventRelatedObjectRole vocabulary contains several subproperties of the *hasEventRelatedObject* property. Possible values are *source* and *outcome* (Fig. 6).

The same holds true for an *Agent* related to an *Event*. The property *hasEventRelatedAgent* has been defined as a subproperty of the *hasAgent* property. The idlc:eventRelatedAgentRole vocabulary contains four subproperties to *hasEventRelatedAgent* for denoting an agent's role in an *Event*, i.e., *authorizer*, *executing program*, *implementer*, and *validator* (Fig. 7).

[4] http://udfr.org/.

[5] http://premisontologypublic.pbworks.com.

[6] http://ipres2013.ist.utl.pt/.

[7] http://www.loc.gov/standards/premis/pif-presentations-2013/04PREMIS-Peyrard-Ontology-idlc.pdf.

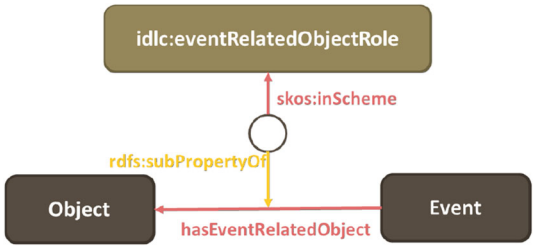

Fig. 6 Specialised subproperties for *hasEventRelatedObject*

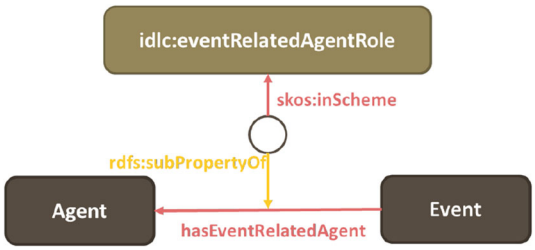

Fig. 7 Specialised subproperties for *hasEventRelatedAgent*

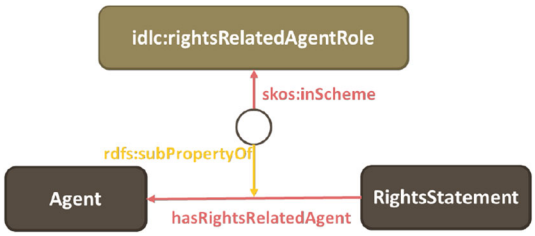

Fig. 8 Specialised subproperties for *hasRightsRelatedAgent*

Finally, an *Agent* can also have different roles when linked to a *RightsStatement*. The property *hasRightsRelatedAgent* is introduced as a subproperty to the *hasAgent* property. The idlc:rightsRelatedAgentRole vocabulary contains three subproperties to *hasRightsRelatedAgent* denoting the *Agent*'s role, i.e., *contact*, *grantor*, and *rightsholder* (Fig. 8).

4.2 PREMIS OWL: structural information

The *Object* instances can have simple to complex relationships between them. For this, PREMIS OWL has the *hasRelationship* property. This is of course too general. The *hasRelationship* has also some subproperties for a finer-grained notion of the relation between two *Objects*. There is a vocabulary for this: idlc:relationshipSubtype. This vocabulary has eight members, e.g., *hasPart*, *hasSource*, or *includes* (Fig. 9).

Sometimes, relationships are more complex and include a certain sequence. For these cases, the class *RelatedObjectIdentification* is introduced. It is related through the same property, i.e., *hasRelationship*, to an object, but at the same time, the class *RelatedObjectIdentification* allows to describe its sequence via the *hasRelatedObjectSequence* property, as depicted in Fig. 10.

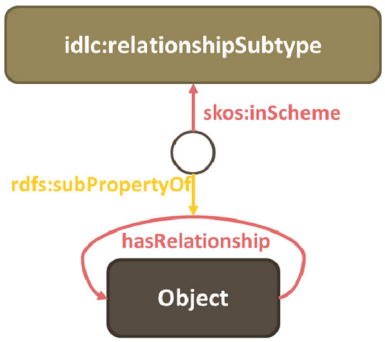

Fig. 9 Specialised subproperties for *hasRelationship*

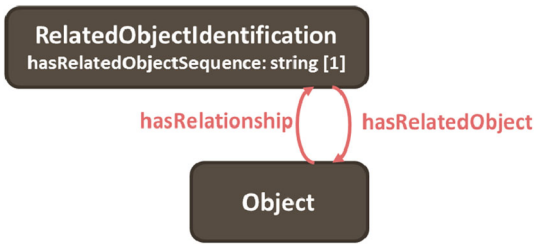

Fig. 10 Describing sequences via the *RelatedObjectIdentification* Class

All the instances of these core classes of PREMIS OWL can have, next to a URI, other identifiers. Many archives tend to use local identifiers, e.g., for preserved objects. Another example of non-URI identifiers are ISBN numbers for books. For these cases, PREMIS OWL foresees an *Identifier* class for describing these identifiers. This class is defined by two mandatory properties, i.e., *hasIdentiferType* and *hasIdentifierValue*. Both properties have a string as range. To link a PREMIS OWL core class instance to the *Identifier* instance, the object property *hasIdentifier* is used.

4.3 Object

As explained in Fig. 2, the *Object* class will contain a lot of technical metadata. This is because of the responsibilities of the digital archive, i.e., keep the information intact, and keep the information interpretable. To fulfill these requirements, the archived objects are described carefully with information such that it can check and take the necessary precautions to keep the archived object intact and interpretable. Intact mainly means assuring the integrity of the stored digital information. Interpretable means assuring the stored digital information remains renderable.

The *Object* class has three subclasses: *Bitstream*, *File* and *Representation*, as shown in Fig. 11. Thus, the objects related to a record to be preserved can be described at different levels. This is needed, because at every level there are preservation risks involved. On the lowest level, the data inside a file can be described using *Bitstream*. At a higher level, the file itself,

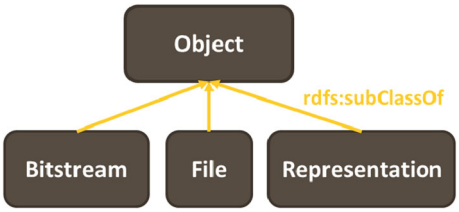

Fig. 11 Subclasses of the *Object* class

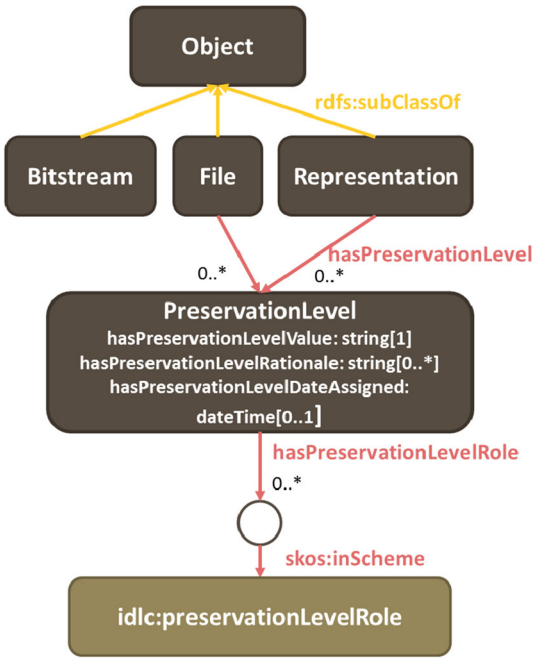

Fig. 12 Overview of the *PreservationLevel* class

of course, can be described using the *File* class. Finally, an object can also be described at representation level. For this, we have the *Representation* class. Typically, a representation is a set of files with some structural metadata relating the files, e.g., a book as a set of ordered TIFF images. Actually the *Object* class has another subclass: *IntellectualEntity*. The *IntellectualEntity* is a class holding the descriptive metadata of a record to be preserved. The descriptive metadata of a record can change over time too, and even the descriptive metadata rights can also be declared. By making the *IntellectualEntity* a subclass of *Object*, one can describe the events related to the descriptive metadata, e.g., reconciliation, and the rights related to the descriptive metadata.

Every archive can have different preservation policies. Even within an archive, there can exist several preservation policies. Via the *PreservationLevel* class, as depicted in Fig. 12, one can assign a certain level of preservation, e.g., some objects may need dissemination online, others do not. This can impact certain transcoding strategies for these objects. The preservation level value indicates the set of preservation functions expected to be applied to the object.

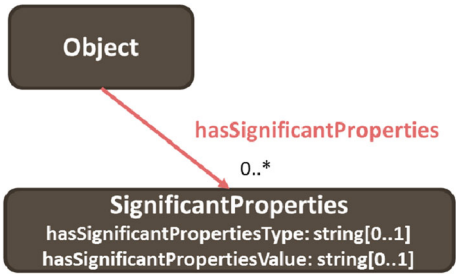

Fig. 13 Overview of the *SignificantProperties* class

This value is dependent on the archive and its policies and thus is denoted with a string. Examples of preservation level values are “bit-level preservation”, “full”, or “online dissemination”. Next to the actual preservation level value, one can also describe the rationale behind the preservation level and the role of the preservation level via a member of the preservation vocabulary idlc:preservationLevelRole. Example values of this vocabulary are ‘requirement’ for denoting what the archive is required to preserve, or ‘intention’ for what is intended to be preserved. Thus, an instance of this class can, e.g., indicate the a certain file is required to be ‘disseminated online’. The same file can have a second preservation level instance linked to it, indicating that it is intended to be bitlevel preserved. Thus, an object can have multiple preservation levels linked to it. The preservation level class also allows describing the rationale behind the preservation level and the date the preservation level was assigned to the object. The preservation level can only be described for *Representations* and *Files*, because *Bitstreams* are always contained in a file, which has the preservation level attached.

Sometimes, an object has certain characteristics that need to be preserved. This can be done by the class *SignificantProperties*. Via this class, these characteristics that need to be preserved can be described via the *hasSignificantPropertiesValue* and *hasSignificantPropertiesType* properties, as depicted in Fig. 13. Such a significant property of an representation can be: ‘all textual content and images’, which is of type ‘content’, or it can indicate that an object needs to be ‘editable’, which has type ‘behavior’. Other repositories may choose to describe significant properties at a more granular level, for example, the significant property value can be ‘7’ for the type ‘page count’ for a file. For a bitstream, a significant property can describe the color space that needs to be preserved. Objects can have thus several significant properties that need to be preserved when migrating. This is why a separate class is created for this.

Once preserved, instances of *Objects* can be digitally signed to guarantee their authenticity. These digital signatures can be described using the *Signature* class, as shown in Fig. 14. This class lets you describe the signature value, the signature’s method and encoding, the val-

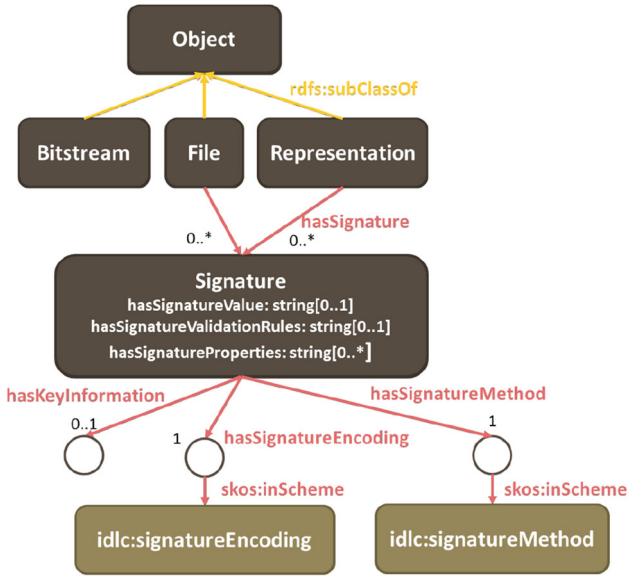

Fig. 14 Overview of the *Signature* class

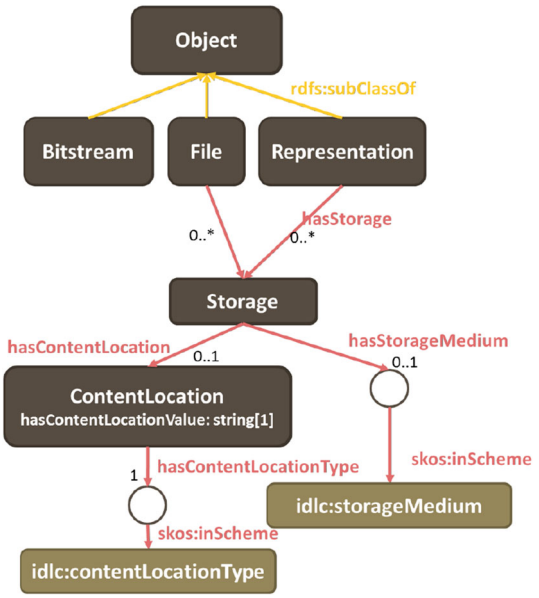

Fig. 15 Overview of the *Storage* class

idation rules, key information and some of its properties. For the encoding and the method, the ontology relies on the resp. preservation vocabularies idlc:signatureMethod and idlc:signatureEncoding. An example of a signature method from the vocabulary can be 'DSA-SHA1' and the signature encoding from the resp. vocabulary can be 'base64'. The actual signature value is stored in the *hasSignatureValue* property. Of course, this class is only foreseen for the *Bitstream* and the *File* class, as a *Representation* consists of *Files* and may be even *Bitstreams*. Files and bitstreams can have multiple signatures and thus a separate class was created for describing the different signatures.

All the digital objects that need to be preserved need to be stored, of course. Some archiving policies will prescribe storage on tape, while others may prescribe storage on hard disks. The storage of a preserved object can be described via the *Storage* class, as depicted in Fig. 15. The *Storage* class makes it possible to describe the storage medium, for which the values are taken from the preservation vocabulary idlc:storageMedium. and the content location. A content location is characterised by a value and a type, taken from the preservation vocabulary idlc:contentLocationType, e.g., 'Handle' or 'URI'. For the same reason, signatures can only be described for bitstreams and files; the storage can also only be described for bitstreams and files.

Of course, the file format information also needs to be described in an *Object* class. This is done through the *ObjectCharacteristics* class. This class aggregates some bit-level information, such as the file format using the *Format* class, the creating application via the *CreatingApplication* class, fixity information through the *Fixity* class, and inhibitor information when encryption is used.

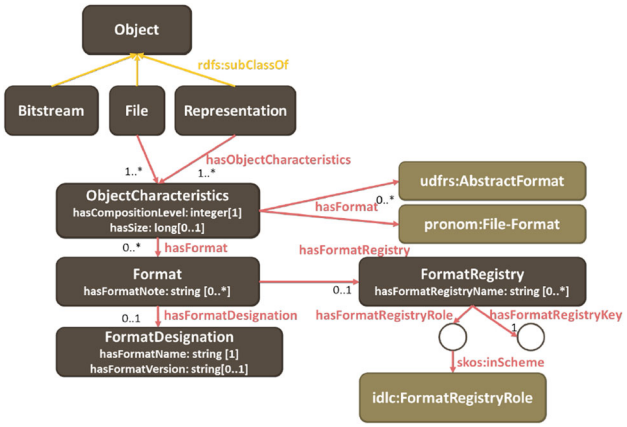

Fig. 16 Overview of the *Format* class

The *Format* class, as shown in Fig. 16, is equivalent to the *udfrs:AbstractFormat* class and the *pronom:File-Format* class. Thus, these classes are interchangeable. The *Format* class is described either by a format designation, which allows describing the format name and format version, or by an identifier from a format registry. Next to the registry key, the role of the registry can also be described using the preservation vocabulary idlc:formatRegistryRole, e.g., 'specification' or 'validation profile'.

A creating application is further detailed by a name of the creating application and a version. Fixity information is described further by its message digest, its message digest originator, and a message digest algorithm, for which the preservation vocabulary idlc:cryptographicHashFunctions is used. The inhibitors, when encryption is used on an object, are characterised by a key, a target and a type. For the last

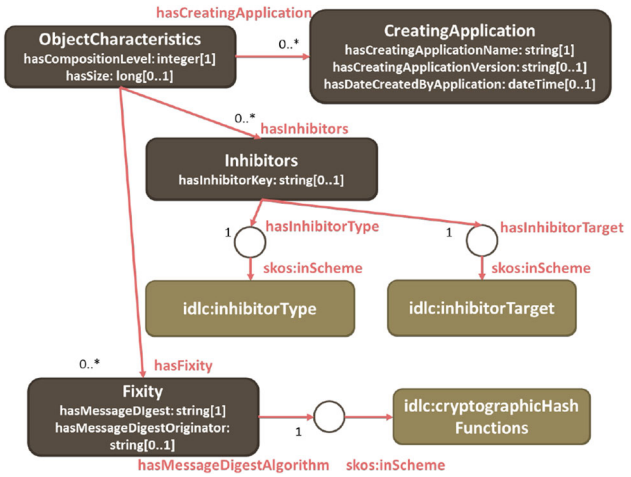

Fig. 17 Overview of the *CreatingApplication*, *Fixity*, and *Inhibitor* classes

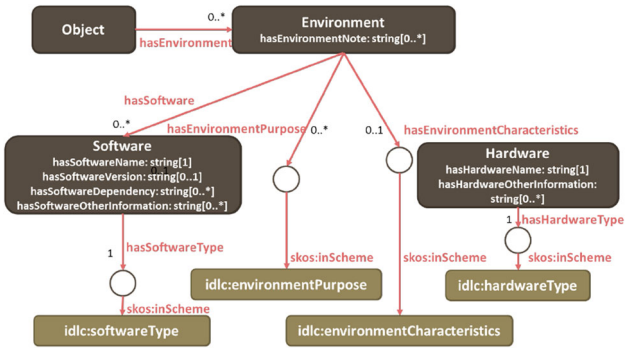

Fig. 18 Overview of the *Environment* classes

two, the resp. preservation vocabularies idlc:inhibitorTarget and idlc:inhibitorType are used. An example value from the idlc:inhibitorTarget vocabulary is 'print function', and from the idlc:inhibitorType vocabulary is 'PGP'. This information is depicted in Fig. 17.

Finally, next to the file format information, information on the rendering environment needs to be stored. To keep information interpretable, there are basically two options: transcoding or emulation. The file format information will support transcoding, and the rendering environment information will support the emulation. For this, the *Environment* class exists, as shown in Fig. 18. This class basically allows to detail the rendering hardware and the rendering software, via the resp. classes *Hardware* and *Software*. Next to this, the *Environment* class also allows describing some environment characteristics, using the preservation vocabulary idlc:environmentCharacteristic, and the environment purpose, via values from the preservation vocabulary idlc:environmentPurpose.

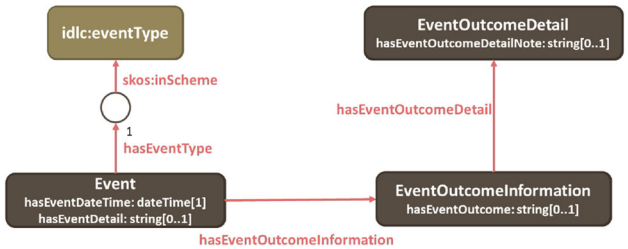

Fig. 19 Schematic description of the *Event* class

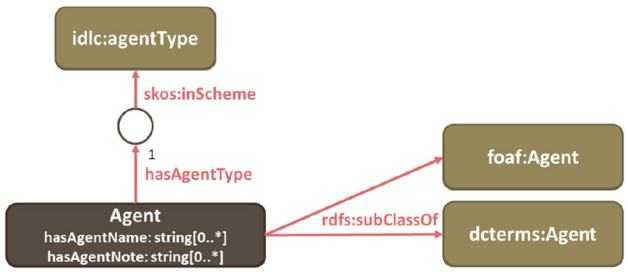

Fig. 20 Schematic description of the *Agent* class

4.4 Event

An event aggregates all the information about an action that involves one or more objects. These metadata are stored separately from the object metadata. Actions that modify objects should always be recorded as events.

The *Event* class, shown in Fig. 19, is described at least by an *eventType* and an *eventDateTime*. The *eventType* values are taken from the preservation vocabulary idlc:eventType. This information can be extended using the *eventDetail* property, which gives a more detailed description of the event, and the *eventOutcomeInformation*, which describes the outcome of the event, in terms of success, failure, partial success, etc.

4.5 Agent

This class aggregates information about attributes or characteristics of agents. Agents can be persons, organisations or software. This class provides the necessary tools to identify unambiguously an agent. The minimum property needed to describe the *Agent* class is *hasAgentType*. The values of this property are taken from the preservation vocabulary idlc:agentType. Optionally, an agent can also be described using the *hasAgentName* and *hasAgentNote*. The *Agent* class is a subclass of *foaf:Agent* and *dcterms:Agent*. In Fig. 20, a schematic overview of the *Agent* class is depicted.

4.6 Rights

PREMIS also foresees to describe rights. The minimum core rights information that a preservation repository must know, however, is what rights or permissions a repository has to

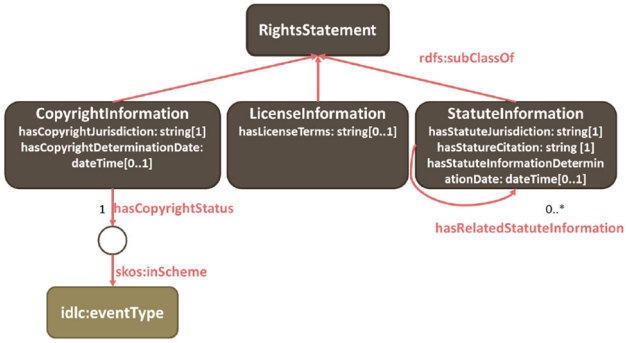

Fig. 21 Subclasses of the *RightsStatement* class

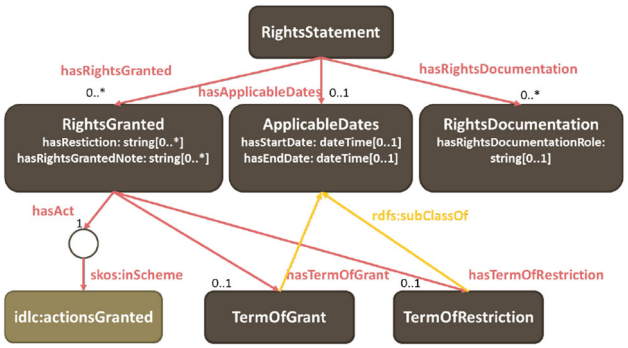

Fig. 22 Details of the *RightsStatement* class

carry out actions related to objects within the repository. The rights or permissions of certain agents may generally be granted by copyright law, by statute, or by a license agreement with the rightsholder. For this, the *RightsStatement* class knows three subclasses, denoting the rights basis for the rights statement: *CopyrightInformation*, *LicenseInformation*, and *StatuteInformation*, as depicted in Fig. 21. In some situations, the basis for the rights is for other reasons, for instance institutional policy. If the basis for the rights is different, one can introduce its own subclass to *RightsStatement*. The *RightsStatement* class on itself is a subclass of *dcterms:RightsStatement*.

Documentation of the rights can be attached to a *RightsStatement* instance using the *RightsDocumentation* class. A *RightsStatement* instance can be further characterised by the dates it is applicable using the *ApplicableDates* class. And, it allows describing the granted rights using the *RightsGranted* class. This *RightsGranted* class is able to denote the actions that are granted to the archive using the *hasAct* property and the term these actions are permissioned using the *hasTermOfGrant* property. The values for this property are taken from the preservation vocabulary idlc:actionsGranted. In Fig. 22, these details on the *RightsStatement* class are shown.

4.7 SKOS vocabularies

As explained earlier, the PREMIS ontology is linked to 24 preservation vocabularies formalised as SKOS. The main reason for this is interoperability. In the PREMIS Data Dictionary, there were a lot of free text fields, which were recommended to be covered by terms of a controlled vocabulary. The PREMIS Data Dictionary did not impose any controlled vocabulary, because these are often very specific to the archive. As a consequence, PREMIS descriptions coming from different institutions were not interoperable, despite the fact they used the same metadata schema.

In PREMIS OWL, SKOS vocabularies were introduced to overcome this interoperability issue. One can think that this comes with the cost of flexibility to adapt the preservation descriptions to the archive's policies, but this is not true. Every archiving institution can still introduce its own SKOS vocabularies. The only condition is that the vocabularies of the institution are linked to the idlc vocabularies. The idlc SKOS vocabularies act as a sort of glue between the PREMIS ontology and the institution-specific vocabularies. This way, interoperability of the preservation metadata between different institutions is maintained. To link the terms of the institution's vocabulary to the idlc vocabulary, one can use the following properties: rdfs:subClassOf, rdfs:subPropertyOf, skos:broader, skos:narrower or skos:related.

In general, the 24 preservation vocabularies integrated in PREMIS OWL cover already the most common preservation policies. They give already a good entry point for institutions wanting to adopt a preservation strategy and using PREMIS OWL as a data model for supporting their preservation tasks. The offered preservation vocabularies can be divided into several categories: refining the *relationships* between the PREMIS entities (Event-Related Agent Role, Event-Related Object Role, Rights-Related Agent Role, and Relationship Subtype). Next, we have some preservation vocabularies used in the object descriptions. They can be split up into really *preservation-specific* vocabularies (Preservation Level Role), *storage*-related vocabularies (Content Location Type, Storage Medium), technical vocabularies supporting the *migrations and rendering* or emulation (Environment Characteristics, Environment Purpose, Format Registry Role, Hardware Type, Object Category, and Software Type), and some *security and integrity* vocabularies (Cryptographic Hash Functions, Inhibitor Target, Inhibitor Type, Signature Encoding, and Signature Method). Finally, there are some vocabularies for describing *agents* (Agent Type), *events* (Event Type), and the *rights* (Actions Granted, Copyright Status, and Rights Basis).

Some vocabularies are also useful outside the preservation context. In the first place the vocabularies for describing security and integrity, the more technical vocabularies supporting emulation and migration, and the vocabularies for describing the rights of an object. The vocabularies are quite general and not too specific. The generality is an advantage for linking your more specialized vocabularies to it, but can also be an disadvantage if you directly want to use one of

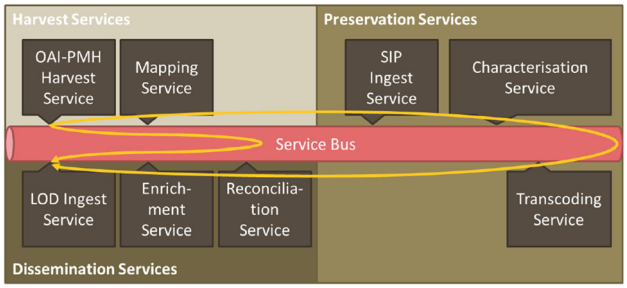

Fig. 23 Schematic overview of the service bus and its connected services

them, because they might lack some specific terms. Overall, they offer a good starting point and in the end they mainly serve to overcome interoperability issues.

5 Preservation infrastructure

In this section, our architecture of the digital long-term preservation archive is briefly described, as it shows how PREMIS OWL support semantic, distributed preservation infrastructures. The presented architecture was the result of the Archipel project.[8] For the project, we developed a demonstrator of a networked, digital infrastructure for archiving and dissemination of multimedia content from the cultural and heritage sectors. During the project, a number of diverse organizations (archives, performing arts organizations, heritage institutions, libraries, museums) could link their archives to our system. The presentation of this system gives insight into the different objects/versions that are created and their preservation information.

Our platform has a service-oriented architecture[9] (SOA). This SOA will make use of a central service hub, which will offer the needed services for the platform. The objectives of our platform are twofold:

- Disseminate the content and provenance information as LOD.
- Enable long-term preservation.

To support these two functionalities, we have designed two workflows, i.e., a dissemination workflow and a preservation workflow, as depicted in Fig. 23. The whole preservation/dissemination cycle starts with a *harvesting process*, which will harvest the metadata and the referenced files. This will typically form the first version of the record to be preserved, namely its original content. The metadata harvested are described using several descriptive metadata formats, e.g., MARC, DC, or CDWA. For management and dissemination purposes, these metadata need to be mapped to DC RDF. For this, we rely on a *mapping service*, which will map the incoming metadata to DC descriptions. This mapping service will create our second version of the record.

If the content also needs to be preserved, the original metadata record, the mapped DC RDF record and the referenced files get packed into a Submission Information Package (SIP), according to the OAIS specifications by the *SIP creator service*. For this SIP, the *BagIt* [2] package format is used. This SIP package is then ingested into the archive, using the *SIP ingest service*.

When ingesting this *BagIt* package into the archive, it has to be supplemented with the preservation information to form an Archival Information Package (AIP) in the OAIS terminology. This package holds all the different versions of the metadata and the multimedia files, referenced by the metadata files. For this preservation information, we will use our described PREMIS OWL ontology. During this ingest process, all files in the package get a PREMIS *Object* description, related to the mapped DC RDF description, thus becoming the PREMIS *intellectual entity*. For this, we rely on a *characterisation service*, which will identify the file format of the files and model the files as PREMIS *Objects*. Every action performed on such a PREMIS *Object*, as shown in the workflow description in Fig. 23, will get related to that *Object* and will be modeled as a PREMIS *Event*. This way, the platform is able to store and track the provenance of the descriptive metadata and the referenced multimedia files.

For this characterisation service, we employed DROID.[10] This tool is designed to automatically identify and validate file formats and to output preservation-related metadata from these files. This output is transformed into a PREMIS *Object* description. This information gets enriched with information from the *UDFR* format registry.[11] This registry publishes the *PRONOM* database,[12] together with *GDFR* registry[13] as LOD, enriched with information from *DBpedia*. This enrichment gives extra information on the needed environment to render or create such file formats. This object description is then also ingested into the SIPs, extending them to AIPs.

The next thing within the workflow is the migration of the stored, related multimedia files. These files get migrated to a certain file format, defined by the archives' preservation plans. Such a preservation plan can stipulate, e.g., that all image files must be migrated to the TIFF file format to keep the image information accessible for long-term preservation purposes, or, e.g., that all image files must be migrated to the

[8] http://events.iminds.be/en/final-event-archipel-network-centric-app-roach-sustainable-digital-archives.

[9] http://opengroup.org/subjectareas/soa.

[10] http://droid.sourceforge.net/.

[11] http://www.udfr.org/.

[12] http://www.nationalarchives.gov.uk/PRONOM/Default.aspx.

[13] http://hul.harvard.edu/gdfr/documents.html.

JPEG file format to keep the image information accessible for dissemination purposes. For this, we need *transcoding services*, which can then migrate various incoming file formats to the appropriate file format according to the preservation plans. This migration will extend the AIP package with the extra migrated data stream. This data stream is then passed to the characterisation service to get a PREMIS *Object* description of the generated data stream and the preservation information is also extended with a description of the migration service as a PREMIS *Event* relating the source object to the migrated object. This transcoding service will create new versions of the referenced multimedia files. These newly created multimedia files will be referenced by the descriptive metadata. Thus, this service creates actually a new version of the record to be preserved, and this for each referenced multimedia file, until all multimedia files are transcoded to the file formats, defined by the preservation plan.

During the last phase, the archived information is moved to the LOD server for dissemination of the information. For this, the descriptive DC RDF metadata will get reconciled by the *reconciliation service*, and then enriched by the *enrichment service* before it gets ingested into the LOD server's triple store by the *LOD ingest service*. Both services create new versions of the descriptive metadata. The reconciliation is different for each data provider and is defined by a rule file. This will, for instance, give all names the same format, e.g., first name last name, to support the enrichment. For the enrichment service, the platform relies on data sources like the *OpenCalais* infrastructure[14] for extracting these named entities, *GeoNames*[15] for enriching the locations, *DBPedia*[16] for enriching the persons, organisations and events, *BibNet*[17] for authors, singers and music bands enrichment, and *Toerisme Vlaanderen*[18] for tourism-related information enrichment on locations. This way, our approach provides (1) unique identifiers for the resource and (2) formalised knowledge about this resource. We will not only disseminate the intellectual entity, i.e., the descriptive metadata, but also the preservation information, so the end-user has access to all the information available about that object.

If the harvested content does not need to be preserved, it is directly routed to our *enrichment service*, which will interlink the data with external data sources after harvesting and mapping the metadata. This enriched DC description then gets ingested into the triple store of the LOD server, which automatically publishes the enriched DC records as LOD.

[14] http://www.opencalais.com/.

[15] http://www.geonames.org.

[16] http://dbpedia.org.

[17] http://www.bibnet.be/.

[18] http://www.toerismevlaanderen.be.

6 Lessons learned

From the implementation of the Archipel infrastructure and the use of PREMIS OWL in this infrastructure, some lessons learned could be extracted. These are presented in this section. The lessons bring up some limitations in the PREMIS OWL ontology, some future work to improve PREMIS OWL and some best practices for the architecture for the preservation platform and how PREMIS OWL supports these best practices.

It must be stressed that long-term preservation structures often lead to large-scale, distributed infrastructures. There are several reasons for this. First of all, such an infrastructure needs processing power. Processing power is needed for the transcodings of file formats, and with the newly developed video file formats, this processing power will only grow in the future. A second reason for the distribution of a preservation platform is the storage requirements. These two aspects lead to a huge storage infrastructure: storage of the files and all their transcodings, and backup of all the files and their transcodings. These two factors lead to a storage solution supporting a multiple of the incoming data size. For the Archipel project, all the content providers offered a total of about 2.5 TB of data. In total, we needed a storage solution of 20 TB, taking into account the transcodings and the backups of all the archived files. A solution is to make your infrastructure distributed, such that the processing power can be spread of multiple machines, and elastic, such that the processing power can grow with the increasing demand of processing power. At the same time, the storage infrastructure needs to distributed as well: you do not want to pull huge files over the wire, just to transcode them. Thus, the solution is to have the storage at the servers which will do the heavy processing to minimise the transport of the files.

Another lesson learned is that not only the storage requirements grow over time rapidly, but also the amount of metadata grows rapidly. Every file and all it versions need an Object description in PREMIS OWL. The objects are all connected with each other via preservation events, such as a transcoding. Thus, the metadata also grow rapidly over time. A good practice here is to separate the descriptive metadata (Intellectual Entities in PREMIS OWL), with the preservation metadata (Objects, Agents, and Events in PREMIS OWL). In many cases, the descriptive metadata are published as open data, and, thus, these metadata will be accessed more than the preservation metadata, which stay often private to the preservation infrastructure. Splitting up the descriptive metadata from the preservation metadata is good, because the growth of the preservation metadata can slow down the access if stored centrally. Another benefit of this split is that the database storing the descriptive metadata can be deployed on a public network, while the database for the preservation metadata will be deployed on a private network. Even the

preservation metadata can be sharded over multiple databases. PREMIS OWL covers this naturally, because it relies on RDF for its metadata descriptions and RDF uses URIs to identify its resources. This feature supports the metadata sharding over multiple databases.

PREMIS OWL can also help to minimise the amount of metadata that needs to be stored. A big part of the preservation metadata consists of the technical metadata, describing, e.g., the file formats. PREMIS OWL foresees to link to external technical registries, such as PRONOM. This way, the preservation platform does not need to store the technical metadata, but can reference it. This is not a very rigid solution, because your preservation platform becomes dependent on the technical registry. A better solution is to come up with unique identifiers of all the file formats. Then, the preservation infrastructure only needs to store these identifiers, instead of a reference to a technical registry. PREMIS OWL already foresees that in its schema.

A last lesson learned is an object of future work in which PREMIS OWL falls short: the support of versioning the descriptive metadata. These metadata will also change over time, as there are always new metadata schemes being standardised and used to describe objects. Not only the metadata schemes change, but also the enrichments of the descriptive metadata can change, certainly nowadays, where these metadata are being published as Linked Open Data. The solution for this, is to treat the descriptive metadata (Intellectual Entities in PREMIS OWL) as PREMIS OWL Objects, which can be versioned. To fully support the versioning of the descriptive metadata, object descriptions in PREMIS OWL also must take into account database objects and not always envision them as files, a bitstream or an aggregation. Thus, a fourth subclass of the Object class, specifically for describing database objects, would cover this shortcoming. In the Archipel infrastructure, this was solved by serialising the descriptive metadata also in files and describe the metadata files as PREMIS OWL Objects. Of course, next to the serialisation in files, the metadata were also stored in the triplestore as database.

7 Related work

Interest in digital preservation can be seen by the multitude of projects in this area. Planets (Preservation and Long-term Access through Networked Services)[19] was especially aimed at defining guidelines for preservation planning. However, it did not tackle the integration of different existing metadata formats, or the dissemination of the metadata as LOD. Likewise, the Prestospace (Preservation towards storage and access) project's objective was to provide technical solutions and integrated systems for a complete digital preservation of all kinds of audio-visual collections.[20] The project was especially focused on the underlying technologies, e.g., automated generation of metadata or detection of errors in content. The project was not focused on using a standardised, semantic preservation model to support the archiving, nor does the project tackle the problem of publishing the generated provenance information to the Web. The CRiB system [5] delivers a set of services that client applications will be able to invoke to perform complex format migrations, evaluate the outcome of those migrations according to multiple criteria (e.g., data loss and performance), and obtain detailed migration reports for documenting the preservation intervention. It is aimed at supporting the migrations of the archived content, but it does not tackle problems like the LOD publication the archived content and their provenance information. PANIC [9], An Integrated Approach to the Preservation of Composite Digital Objects using Semantic Web Services, is an integrated, flexible system, which leverages existing tools and services and assists organizations to dynamically discover the optimum preservation strategy for compound objects. The system has a service-oriented architecture and the Web services are semantically described, allowing to discover the most appropriate preservation strategy for the compound object or its aggregated objects. But, as the CRiB system, it does not focus on the publication of the content or on the publication of the provenance information.

The CASPAR project (Cultural Artistic and Scientific knowledge for Preservation, Access, and Retrieval) presented technologies for digital preservation.[21] The OAIS Reference Model was chosen as the base platform, and the project was focused on implementing the different steps in the preservation workflow. They focus more on preservation services than on describing the preservation information. BOM Vlaanderen,[22] a national research project, aimed at preservation and disclosure of audio-visual content in Flanders. Additionally, it looked at ways to unify different metadata standards currently used for describing audio-visual content. Current trends are on integrating different media archives. Presto-PRIME has investigated and developed practical solutions for the long-term preservation of digital media objects, programmes and collections, and finds ways to increase access by integrating the media archives with European on-line digital libraries in a digital preservation framework.[23]

The previous discussed related works were focusing on the digital long-term preservation, not on the more general problem of enabling their provenance information on the Web. For the work done in this area, the work of the W3C Prove-

[19] http://www.planets-project.eu/.

[20] http://prestospace.org/project/index.nl.html.

[21] http://www.casparpreserves.eu/.

[22] http://events.iminds.be/category/event-tags/bom-vlaanderen.

[23] http://www.prestoprime.org/.

nance Incubator Group[24] is the major reference. This incubator group produced working definitions for provenance information, provided a state-of-the-art understanding and developed a roadmap for development and possible standardisation of provenance on the Web. This work included defining key dimensions for provenance, collecting use cases, designing three flagship scenarios from the use cases, creating mappings between existing provenance vocabularies, looking how provenance could fit in the Web architecture and providing a state-of-the-art report on the current provenance activities. Their work is summarised in a final report [7]. The first flagship scenario describes a news aggregator site that assembles news items from a variety of data sources, e.g., news sites, blogs and tweets. The provenance records of these data providers can help with verification, credit and licensing. This flagship scenario could be covered by publishing the provenance information using our framework. What still forms a problem is the lack of a standardised metadata model for publishing provenance on the Web. In our framework, we publish the provenance information as LOD using PREMIS OWL. This information is only interoperable in the long-term preservation context, where PREMIS is well known, not in a Web context. This standardised provenance model for the Web is still a major research area. The work of the W3C Provenance Incubator Group [7] was a first step into that direction. The W3C Provenance Working Group [6] finalised a specification for provenance on the Web.

Another interesting work done in the area of publishing provenance for linked data is the paper of Olaf Hartig and Jun Zhao published at IPAW [8]. In that paper, they describe the Provenance Vocabulary[25] used for describing the provenance information as Linked Open Data. Next to this, they also offer ways of publishing this provenance information for Linked Data. They discuss how provenance can be added to Linked Data objects, how provenance can be included into RDF dumps and how the provenance information can be queried using SPARQL endpoints. This work enables provenance for Linked Data, but it does not offer solutions for automatic discovery of the provenance information or ways for publishing provenance on the Web beyond using semantic web technologies. Future work could involve publishing the provenance information using this vocabulary, which is more suited for publication on the Web than PREMIS OWL, which is intended to be a data model for digital long-term archives. The mapping table, relating various provenance vocabularies, produced by the W3C Incubator Group[26] will be the reference for this work.

[24] http://www.w3.org/2005/Incubator/prov/wiki/W3C_Provenance_Incubator_Group_Wiki.

[25] http://purl.org/net/provenance/.

[26] http://www.w3.org/2005/Incubator/prov/wiki/Provenance_Vocabulary_Mappings.

8 Conclusions

In this article, we have presented PREMIS OWL, an ontology based on the PREMIS Data Dictionary for Preservation Metadata version 2.2, a digital preservation standard based on the OAIS reference model. The ontology was first designed within the project Archipel, initiating the long-term preservation archivation in Flanders. Later on, this ontology was picked up by the Library of Congress. After several years of cooperation with the Library of Congress, the Bibliothque nationale de France (BNF), and Family Search in refining the ontology, it is now supported by the Library of Congress. At the moment, the ontology is published, and maintained by the Library of Congress, as their official semantic binding of the PREMIS.

The ontology tries to stick as closely as possible to the PREMIS Data Dictionary semantic unit definitions. The ontology is ideal for creating, validating and storing the preservation metadata of a particular digital asset. Before, the PREMIS Data Dictionary was only implemented as an XML schema. This semantic binding of PREMIS does not replace the XML schema; it is complementary to it.

The main difference of the ontology to the data dictionary is the introduction of 24 preservation vocabularies, formalised as SKOS vocabularies, also published by the Library of Congress. In the XML version of the data dictionary, every institution could use their own controlled vocabularies, reflecting its policies. This feature prohibited interoperability between different archiving institutions. In the ontology, the use of proprietary vocabularies is still possible, but they must be linked to the preservation vocabularies to guarantee interoperability across institution boundaries. This feature is becoming more and more important as on the long term, the designated community of an institution can change and an archive can thus decide to transfer part of its collection to another archive. Interoperability of the preservation metadata guarantees in these situations that the preservation metadata can be moved together with the transferred objects.

This OWL ontology allows one to provide a Linked Data-friendly, LOC-endorsed serialization of the PREMIS Data Dictionary version 2.2. This can be leveraged to have a Linked Data-friendly data management function for a preservation repository, allowing for SPARQL querying. It integrates PREMIS information with other Linked Data compliant datasets, especially format registries, which are now referenced from the PREMIS ontology (for instance, the Unified Digital Format Registry and PRONOM). Thus information can be more easily interconnected, especially between different repository databases. Thus, our preservation information can be enriched now.

PREMIS OWL also introduces 24 preservation vocabularies that were developed by the Library of Congress. This feature makes the preservation metadata interoperable

amongst the archives. Each archive has its specific preservation processes in place to fulfill its tasks. These specific processes need to be described in the preservation metadata. Each archive can describe its preservation processes using its own SKOS vocabularies. To be conform to PREMIS OWL, these SKOS vocabularies need to be linked to the 24 preservation vocabularies introduced in PREMIS OWL. This will make the preservation metadata interoperable. This feature is needed as archived objects are more and more being exchanged among archives across the world.

References

1. Berners-Lee, T.: Linked Data. In: W3C Design Issues (2006). http://www.w3.org/DesignIssues/LinkedData.html
2. Boyko, A., Kunze, J., Littman, J., Madden, L., Vargas, B.: The bagIt file packaging for-mat (V0.96) (2009). https://confluence.ucop.edu/download/attachments/16744580/BagItSpec.pdf?version=1
3. Consultative Committee for Space Data Systems. Reference model for an open archival Information system (OAIS) (2002). http://public.ccsds.org/publications/archive/650x0b1.pdf
4. RLG/OCLC Working Group on Digital Archive Attributes. Trusted digital repositories: attributes and responsibilities (2002). http://www.oclc.org/programs/ourwork/past/trustedrep/repositories.pdf
5. Ferreira, M., Baptista, A.A., Ramalho, J.C.: An intelligent decision support system for digital preservation. Int. J. Digit. Libr. **6**(4), 295–304 (2007)
6. Gil, Y., Miles, S., Belhajjame, K., Deus, H., Garijo, D., Klyne, G., Missier, P., Soiland-Reyes, S., Zednik, S.: PROV model primer. Tech. rep. W3C (2012). http://www.w3.org/TR/prov-primer/
7. Gil, Y., Cheney, J., Groth, P., Hartig, O., Miles, S., Moreau, L., da Silva, P.P., Coppens, S., Garijo, D., Gomez, J.M., Missier, P., Myers, J., Sahoo, S., Zhau, J.: Provenance XG final report (2010). http://www.w3.org/2005/Incubator/prov/XGR-prov/
8. Hartig, O., Zhao, J.: Publishing and consuming provenance metadata on the web of linked data. In: Proceedings of the 3rd International Provenance and Annotation Workshop IPAW (2010). http://olafhartig.de/files/HartigZhao_Provenance_IPAW2010_Preprint.pdf
9. Hunter, J., Choudhury, S.: PANIC: an integrated approach to the preservation of composite digital objects using semantic web services. Int. J. Digit. Libr. **6**(2), 174–183 (2006)
10. McGuinness, D., van Harmelen, F.: OWL web ontology language: overview. W3C recommendation. World Wide Web Consortium (2004). http://www.w3.org/TR/owl-features/
11. Messina, A., Boch, L., Dimino, G., Bailer, W., Schallauer, P., Allasia, W., Basili, R.: Creating rich metadata in the TV broadcast archives environment: the PrestoSpace project. In: IEEE AXMEDIS06 Conference Second International Conference on Automated Production of Cross Media Content for Multi-Channel Distribution. pp. 193–200 (2006)
12. Miles, Alistair, Pérez-Agüera, José R.: SKOS: simple knowledge organisation for the web. Cat. Classif. Q. **43**(3), 69–83 (2007). doi:10.1300/J104v43n03_04
13. PREMIS working group.: PREMIS data Dictionary for preservation metadata—version 2.0 (2008). http://www.loc.gov/standards/premis/v2/premis-2-0.pdf