PREMIS OWL: Introduction, Implementation Guidelines & Best Practices

Sam Coppens (IBBT-Ugent-MMLab), Sebastien Peyrard(BnF), Rebecca Guenther (LOC), Kevin Ford (LOC), Tom Creighton (FamilySearch)

September 27th, 2011

PREMIS OWL: INTRO & BEST PRACTICES

In this document we discuss PREMIS OWL, the new semantic binding of the PREMIS 2.1 Data Dictionary.

INTRODUCTION TO PREMIS 2.1

PREMIS is a preservation standard based on the OAIS reference model, which is in fact provenance metadata supplemented with technical metadata and rights metadata to support preservation actions. This standard is currently in version 2.1, as the *PREMIS Data Dictionary for Preservation Metadata*. An XML schema is provided that implements the data dictionary for digital preservation. This preservation standard is described by a data model, which consists of five semantic units or classes important for digital preservation purposes:

- *Intellectual Entities*: 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 to the rights of an object.
- *Rights*: description of one or more rights, permissions of an object or an agent.

A new version is under development which will change the data model to make intellectual entities another level of object, rather than a separate entity. *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 on Figure 1. 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. Binary metadata, technical metadata, fixity metadata and structural metadata are encapsulated in the PREMIS data dictionary via the description of the *object* entity.

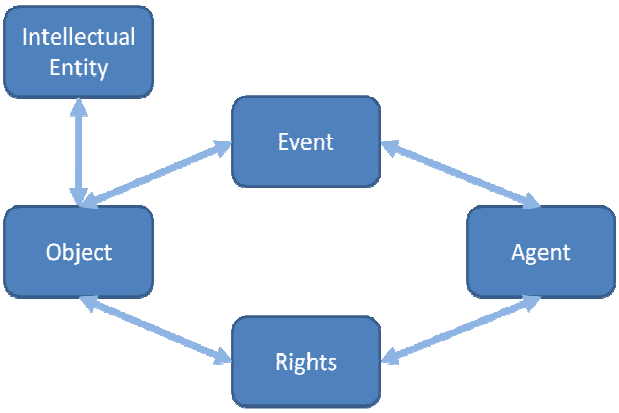

Figure 1: Data Model of PREMIS (version 2.1)

WHY PREMIS OWL?

Looking at the data model, one can notice it is dynamically relating the five entities to each other. Until now an XML schema was available that implemented the PREMIS 2.1 data dictionary. The PREMIS XML Schema is great for creating, validating and storing preservation metadata for a particular representation, whereas the same information in RDF can be more easily interlinked, especially between assets coming from different repositories. It also allows one to go beyond the Information Package level, by providing a standardized way to query a whole dataset using the SPARQL language and protocol.

For instance, the XML schema uses the identifiers of the entities to relate those to each other. As a consequence, the relations between the entities are directed and not bidirectional. Implementing the data dictionary using the Web Ontology Language (OWL), allows us to relate the entities directly to each other, without the need of referring to an identifier of the entity. Another advantage of using semantic web technologies to implement the PREMIS 2.1 data dictionary is that the relations can be made bidirectional using inverse properties.

For all these reasons, the OWL design of PREMIS should not be considered as a replacement for the XML Schema: the two of them should rather be considered complementary.

DESIGN PERSPECTIVE

For the implementation of the formalised ontology of the PREMIS 2.1 data dictionary, the XML schema of the data dictionary is used as a starting point. It needs to be stressed that PREMIS is intended to model the types of information needed for long term preservation, though on top of that, PREMIS can be used to disseminate this preservation information. When designing the OWL ontology of the PREMIS 2.1 the choice was hence made to stick as closely as possible to the data dictionary of PREMIS 2.1. The reason for this is that information loss is unacceptable, when using PREMIS OWL. The data dictionary of PREMIS 2.1 was developed by experts in the domain of long-term preservation, and every element has its own clearly defined semantics. One cannot just import other vocabularies into the ontology implementing PREMIS 2.1, because then the clearly defined semantics of the replaced elements are lost.

PREMIS OWL

In this section the PREMIS OWL ontology is explained, and some design decisions made are discussed. For describing the PREMIS OWL ontology, we give an overview of every entity defined by PREMIS, which are modelled as separate classes in OWL, list the deviations from the PREMIS 2.1 Data Dictionary, and give an overview of the links to other semantic units, because this linking to other semantic units has changed for every unit. In XML one was forced to link to identifiers of objects, when relating these semantic units. In RDF and OWL one can use object properties, which will use the URIs of the instances to link directly to them.

GENERAL DEVIATIONS FROM THE PREMIS 2.1 DATA DICTIONARY

In the PREMIS data dictionary, the notions of identifier and extension are globally defined, even if they are represented by different semantic units in the Data Dictionary. We tried to reflect this in the ontology by defining generic classes / properties.

IDENTIFIERS

In PREMIS, all identifiers are similar in terms of structure: each Identifier has a value, and a type giving the domain in which the identifier is unique. They also play the same role: to identify uniquely something, whether it be an instance of Object, Event, Agent, Rights statement, a license, or a dependency. In PREMIS OWL, we decided to use a generic premis:identifier property, linking something to its identifier; and a generic Identifier class, allowing you to declare explicitly your premis:identifierType and Value if you want to. More on this in Section 'Implementation Guidelines and Best Practices – Linking to other Vocabularies'

EXTENSIONS

In PREMIS, an Extension is really meant to be outside the scope of PREMIS: it is a container unit with no defined components. Therefore, there is no need to define a dedicated class for each particular Extension.

Therefore, all particular extension defined in the data dictionary (namely, significantPropertiesExtension, creatingApplicationExtension, objectCharacteristicsExtension, environmentExtension, signatureInformationExtension, eventOutcomeDetailExtension, agentExtension, rightsExtension) are modelled as a single Extension class.

OBJECT

The *Object* class describes a unit of information in digital form, as shown in Listing 1. It is related to the *Intellectual Entity* class. Typically, the intellectual entity consists of the descriptive metadata, and the objects related to this intellectual entity are the multimedia files representing this content under digital form. For instance, the intellectual entity can be a theatre performance described using DC, the objects are related to this theatre performance, which can be a photo of an actor, a video showing a piece of the performance, or a review published in some newspaper. The descriptive metadata used for describing the intellectual entities are very domain-specific. For this, there already exist a lot of descriptive metadata models, and in version 2.1 and earlier it was considered out of scope for PREMIS. Note that a version 3.0 is under development that will make intellectual entities another level of object and thus not a separate class. When the new version is released, the OWL ontology will also be revised to make *Intellectual Entity* a subclass of *Object*.

An *Object* class knows three disjoint subclasses:

- *File*: a file is an ordered sequence of bytes that is known to the system.
- *Bitstream*: Contiguous or non-contiguous data within a file that has meaningful properties for preservation purposes
- *Representation*: a representation is a set of files with structural metadata needed for a complete manifestation of an intellectual entity.

The *Object* class possesses all the necessary features to describe the object on the different levels. This description on the different levels is a recommendation of the OAIS reference model. Both the *File* and *Bitstream* subclasses must have at least a predicate *objectCharacteristics*, linking to the *ObjectCharacteristics* instance, which gives the necessary technical and binary metadata. The *Bitstream* subclass is also a subclass of *ore:AggregatedResource*, because a file can be described as an aggregation of *Bitstream* instances. The *File* subclass is a subclass of *ore:AggregatedResource* and *ore:Aggregation*. The reason for this is that a file can be an aggregation of bitstream, but it can also be part of a representation, which aggregates some files. The *Representation* subclass is also a subclass of *ore:Aggregation*, as it consists of different *File* instances.

An object can be described further into detail using: *preservationLevel*, as some repositories offer the opportunity to define a preservation level for an object; *significantProperties*, defining some significant properties of the object, which need to be preserved when, e.g., migrating the data; *originalName*, for indicating the original names of the packages delivered to the repository; *environment*, which describes the environment the user needs to render the content and interact with the content; and *signatureInformation*, for storing digital signatures generated during ingest.

DEVIATIONS FROM THE PREMIS 2.1 DATA DICTIONARY

The PREMIS 2.1 Data Dictionary defined for every object a mandatory ***objectCategory***, denoting the category of the object: bitstream, file or representation. In the PREMIS XML schema this is implemented as an xsi:type instead of as an explicit data value. This is left out of the PREMIS OWL ontology, because this information is captured implicitly by subclasses of the Object class: *Bitstream*, *File*, and *Representation*.

The predicate ***preservationLevelRole*** is linked to a SKOS vocabulary, published by the Library of Congress at http://id.loc.gov/vocabulary/preservationLevelRole. The property for denoting the used message digest algorithm, i.e., ***messageDigestAlgorithm*** under fixity and ***signatureMethod***, in describing digital signatures are also linked to a SKOS vocabulary of the LOC, published at http://id.loc.gov/vocabulary/cryptographicHashFunctions.

LINKS TO OTHER SEMANTIC UNITS

For linking *objects* to other *objects*, *events*, *intellectual entities*, and *rights statements*, The PREMIS 2.1 Data Dictionary defined *relationship*, for relating two *objects*, *linkingEventIdentifier*, for linking an *object* to an *event*, *linkingIntellectualEntityIdentifier*, for relating an *object* to an *IntellectualEntity*, and *linkingRightsStatementIdentifier*, for linking an *object* to a *RightsStatement*.

The ***relationship*** element relates two or more *objects* to each other. These relationships can be structural or derivational. For denoting the relationship type and subtype, the Library of Congress will publish a SKOS vocabulary based on the suggested values of the Data Dictionary at http://id.loc.gov/. This vocabulary will publish the SKOS Concepts also as subproperties of the *relationship* property. Another option is to define your

own SKOS vocabulary. More on this in Section 'Implementation Guidelines and Best Practices – Linking to other Vocabularies'.

The *linkingEventIdentifier* property, *linkingIntellectualEntityIdentifier property*, and *linkingRightsStatementIdenfitier*, property are replaced resp. by the ***linkingEvent*** object property, ***linkingIntellectualEntity*** object property, and the ***linkingRightsStatement*** object property.

EXAMPLE

```
@prefix rdf:    <http://www.w3.org/1999/02/22-rdf-syntax-ns#>.
@prefix rdfs:   <http://www.w3.org/2000/01/rdf-schema#>.
@prefix owl:    <http://www.w3.org/2002/07/owl# .
@prefix premis: <http://multimedialab.elis.ugent.be/users/samcoppe/ontologies/Premis/premis.owl#>.

<object1>                          a                    premis:File;
        premis:preservationLevel                        <object1PreservationLevel>;
        premis:significantProperties                    <object1SignificantProperties>;
        premis:objectCharacteristics                    <object1ObjectCharacteristics>;
        premis:originalName                             "0001h.tif";
        premis:storage                                  <object1Storage>;
        premis:environment                              <object1Environment>;
        premis:linkingEvent                             <event2>;
        premis:linkingRightsStatement                   <rightsstatement1>;
        premis:linkingIntellectualEntity                <dublinCoreDescription1>.

<object1PreservationLevel>         a                    premis:PreservationLevel;
        premis:preservationLevelValue                   "full";
        premis:preservationLevelRole  <http://id.loc.gov/vocabulary/preservationLevelRole/requirement>;
        premis:preservationLevelDateAssigned            "2010-07-29T14:41:28".

<object1SignificantProperties>     a                    premis:SignificantProperties;
        premis:significantPropertiesType                "behavior";
        premis:significantPropertiesValue               "hyperlinks traversable".

<object1ObjectCharacteristics>     a                    premis:ObjectCharacteristics;
        premis:compositionLevel                         "0";
        premis:fixity                                   <object1Fixity>;
        premis:size                                     "20800896";
        premis:format                                   <object1Format>;
        premis:creatingApplication                      <object1CreatingApplication1>;
        premis:objectCharacteristicsExtension           <object1CharacteristicsExtension>.

<object1Fixity>                    a                    premis:Fixity;
        premis:messageDigestAlgorithm  <http://id.loc.gov/vocabulary/cryptographicHashFunctions/md5>;
        premis:messageDigest                            "36b03197ad066cd719906c55eb68ab8d";
        premis:messageDigestOriginator                  "LocalDCMS".

<object1Format>                    a                    premis:Format;
        premis:formatDesignation                        <object1FormatDesignation>;
        premis:formatRegistry                           <object1FormatRegistry>.

<object1FormatDesignation>         a                    premis:FormatDesignation;
        premis:formatName                               "image/tiff";
        premis:formatVersion                            "6.0".

<object1FormatRegistry>            a                    premis:FormatRegistry;
```

```
        premis:formatRegistryName          "PRONOM";
        premis:formatRegistryKey           <http://reference.data.gov.uk/id/file-format/10>;
        premis:formatRegistryRole          "specification".

<object1CreatingApplication1>   a          premis:CreatingApplication;
        premis:creatingApplicationName     "Adobe Photoshop";
        premis:creatingApplicationVersion  "CS2";
        premis:dateCreatedByApplication    "2006-09-20T08:29:02".

<object1Storage>                a          premis:Storage;
        premis:contentLocation             <object1ContentLocation>;
        premis:storageMedium               "disk".

<object1ContentLocation>        a          premis:ContentLocation;
        premis:contentLocationType         "filepath";
        premis:contentLocationValue        "amserver".

<object1Environment>            a          premis:Environment;
        premis:environmentCharacteristic   "recommended";
        premis:environmentPurpose          "render";
        premis:environmentPurpose          "edit";
        premis:software                    <object1Software1>;
        premis:hardware                    <object1Hardware1>.

<object1Software1>              a          premis:Software;
        premis:swName                      "Adobe Acrobat";
        premis:swVersion                   "5.0";
        premis:swType                      "renderer".

<object1Hardware1>              a          premis:Hardware;
        premis:hwName                      "Intel x86";
        premis:hwType                      "processor";
        premis:hwOtherInformation          "60 mhz minimum".
```

Listing 1: Example of a PREMIS OWL Object instance in N3 notation

EVENT

An *event* aggregates all the information about an action that involves one or more *objects*. Actions that modify *objects* should always be recorded as *events*.

The *Event* class is described at least by an *eventType*, e.g. capture, creation, or migration, and an *eventDateTime*. 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, or partial success. These properties are able to describe any *event* altering an *object*.

DEVIATIONS FROM THE PREMIS 2.1 DATA DICTIONARY

A difference to the data dictionary involves the **eventIdentifier**, which was obligatory. This is replaced by an optional *Identifier* class. More details on identifiers are given in Section 'Implementation Guidelines and Best Practices – Identifiers'.

The ***eventType*** property is linked to a SKOS vocabulary, denoting the types of an event, published by the Library of Congress. The SKOS vocabulary is published at http://id.loc.gov/vocabulary/preservationEvents.

LINKS TO OTHER SEMANTIC UNITS

The *Event* class can be related to an *Agent* class or *Object* class via the resp. properties ***linkingAgent*** and ***linkingObject***. For denoting the agent role in the event, the Library of Congress publishes a SKOS vocabulary, where the SKOS Concepts are made subproperties of *linkingAgent.* The SKOS vocabulary will be published at http://id.loc.gov/. Of course, you could also define your own SKOS vocabulary. More on this in Section 'Implementation Guidelines and Best Practices – Linking to other Vocabularies'.

EXAMPLE

```
@prefix rdf:    <http://www.w3.org/1999/02/22-rdf-syntax-ns#>.
@prefix rdfs:   <http://www.w3.org/2000/01/rdf-schema#>.
@prefix owl:    <http://www.w3.org/2002/07/owl#>.
@prefix premis: <http://multimedialab.elis.ugent.be/users/samcoppe/ontologies/Premis/premis.owl#>.

<event1>                          a                  premis:Event;
        premis:identifier                            <event1ID>;
        premis:eventType                  <http://id.loc.gov/vocabulary/preservationEvents/migration>;
        premis:eventDateTime                         "2010-08-06T00:00:00.002";
        premis:eventDetail                           "ImageMagick";
        premis:eventOutcomeInformation               <event1OutcomInformation>;
        premis:linkingIssuer                         <agent1>;
        premis:linkingObject                         <object1>;
        premis:linkingObject                         <object2>.

<event1ID>                        a                  premis:Identifier;
        premis:identifierType                        "LocalDCMS";
        premis:identifierValue                       "E002.1".

<event1OutcomeInformation>        a                  premis:EventOutcomeInformation;
        premis:eventOutcome                          "successful".
```

Listing 2: Example of a PREMIS OWL Event instance in N3 notation

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 properties needed to describe the *Agent* class are *agentIdentfier* and *agentType*. Optionally, an agent can also be described using the *agentName*.

LINKS TO OTHER SEMANTIC UNITS

An *agent* can hold or grant one or more *rights*. It may carry out, authorise, or compel one or more *events*. An *agent* can only create or alter an *object* through an *event* or with respect to a *rights statement*. The relationships between an *agent* and an *object* through an *event* or *rights* entity make it possible to describe the whole provenance of an object. An agent can be linked to an event or rights statement using the ***linkingEvent*** and ***linkingRightsStatement*** properties. Listing 3 gives an example of such an *Agent* instance.

EXAMPLE

```
@prefix rdf:     <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs:    <http://www.w3.org/2000/01/rdf-schema#> .
@prefix owl:     <http://www.w3.org/2002/07/owl#> .
@prefix premis: <http://multimedialab.elis.ugent.be/users/samcoppe/ontologies/Premis/premis.owl#>.

<agent1>                    a           premis:Agent;
        premis:identifier               <agent1ID>;
        premis:agentType                "person";
        premis:agentName                "name";
        premis:linkingEvent             <event1>;
        premis:linkingObject            <object1>;
        premis:linkingObject            <object2>.

<agent1ID>                  a           premis:Identifier;
        premis:identifierType           "OpenID";
        premis:identifierValue          "http://some.openID.url
```

Listing 3: Example of a PREMIS OWL Agent instance in N3 notation

RIGHTS

The minimum core rights information that a preservation repository must contain, is what rights or permissions the repository has regarding the objects within the repository. These may be granted by copyright law, by statute, or by a license agreement with the rights holders. *Rights* entities can be related to one or more *objects* and one or more *agents*.

Every *Rights* instance can be related to different *RightsStatements*. A *RightsStatement* knows three subclasses: the *Copyright* subclass, the *License* subclass, and the *Statute* subclass. These three subclasses offer the necessary metadata for describing, rights information, i.e., *copyrights*, *licenses*, and *statutes*. Every *RightsStatement* is described at least by a *rightsStatementIdentifier*, and has also the optional property *rightsGranted*, which describes the actions the granting agency has allowed the repository.

DEVIATIONS FROM THE PREMIS 2.1 DATA DICTIONARY

The PREMIS 2.1 Data Dictionary defined for every *RightsStatement* a mandatory ***rightsBasis***. This is left out of the PREMIS OWL ontology, because this information is captured implicitly by subclasses of the *RightsStatement* class: the *Copyright* subclass, the *License* subclass, and the *Statute* subclass.

LINKS TO OTHER SEMANTIC UNITS

The *RightsStatement* class can be related to an *Object* class or *Agent* class via the optional, repeatable object properties: ***linkingObject*** and ***linkingAgent***. Listing 4 gives an example of such a rights instance.

EXAMPLE

```
@prefix rdf:     <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs:    <http://www.w3.org/2000/01/rdf-schema#> .
@prefix owl:     <http://www.w3.org/2002/07/owl#> .
@prefix premis: <http://multimedialab.elis.ugent.be/users/samcoppe/ontologies/Premis/premis.owl#>.

<rights1>                       a       premis:License;
        premis:identifier               <http://some.base.uri/rights/resource/dissemination>;
        premis:licenseInformation       <licenseInformation1>;
        premis:rightsGranted            <rightsGranted1>;
        premis:linkingObject            <object1>;
        premis:linkingObject            <object2>;
        premis:linkingContact           <agent1>.

<licenseInformation1>           a       premis:LicenseInformation;
        premis:identifier               <http://some.base.uri/license/resource/dissemination>;
        premis:licenseTerms             "Here comes the actual text of the license.";
        premis:licenseNote              "These objects may be disseminated.".

<rightsGranted1>                a       premis:LicenseInformation;
        premis:act                      <license1identifier>;
        premis:termOfGrant              <license1termofgrant>.

<license1termofgrant>           a       premis:TermOfGrant;
        premis:startDate                "2009-09-01T08:30:00".
```

Listing 4: Example of a PREMIS OWL Rights instance in N3 notation

LINKS TO OTHER VOCABULARIES

Note: the considerations in the end of the document give you hints on the how you could implement the ontology. Some modelling options remain open to discussion during the review period. See http://premisontologypublic.pbworks.com/w/page/45987628/Questions%20for%20reviewers for a dedicated page on which you can leave your comments.

ORE:AGGREGATION AND ORE:AGGREGATEDRESOURCE

The PREMIS OWL *Bitstream* class is a subclass of *ore:AggregatedResource*, the *File* class is a subclass of both the *ore:Aggregation* class and the *ore:AggregatedResource* class. Finally, the PREMIS OWL *Representation* class is subclassed to the *ore:Aggregation* class. By declaring this, we allow the use of *ore:aggregates* or *ore:isAggregatedBy* properties to express structural relationships between objects.

DCTERMS:AGENT AND FOAF:AGENT

The PREMIS OWL *Agent* class is subclassed to *foaf:Agent* and *dcterms:Agent*.

RDFS:LABEL

Some of the datatype properties in PREMIS OWL (typically *Name and *Value properties) could be replaced by *rdfs:label*. This way, the preservation information can be queried using the *rdfs:label* property, whenever there is a *Name or *Value in the RDF dataset. This would make the model more concise – one single *rdfs:label* property whenever you have a *Name or *Value property – but would make the OWL ontology very different from the original Data Dictionary on the other hand.

IMPLEMENTATION GUIDELINES AND BEST PRACTICES

IDENTIFIERS

HOW DO I DECLARE MY IDENTIFIER TYPES AND VALUES IN RDF?

You can describe extra identifiers for every PREMIS Entity using the *identifierType and identifierValue* properties. This information is optional, and is **only** relevant when the extra identifiers aren't URIs. Otherwise that would be redundant, since every instance of Object, Event, Rights or Agent in RDF will already have its own URI as the subject of any declaration. So, if the archive has globally unique URIs, these will be used for identifying the instances and no extra information will be needed about them.

To know how to handle your identifiers, you should check if your identifierType belongs to a registered URI scheme or not. The list of valid URIs are given by the IANA: see http://www.iana.org/assignments/uri-schemes.html. In particular:

- All URLs beginning with "http:" are indeed URIs, so they need not be re-expressed using the premis:identifier property. Therefore PURLs and any other HTTP-based identifiers are valid URIs.
- URNs and INFO:URIs are also registered URIs.
- Identifiers beginning with "ark:" and "doi:" are not valid URIs

If your identifiers are not URIs, you can choose to convert your identifiers into valid URIs. In the latter case, the use of HTTP URIs is recommended when another URI scheme (like INFO:URIs) has not already been used to solve the problem. For example:

- The ARK identifiers are not valid URI schemes (at least not the mandatory part beginning with "ark:"), but using an HTTP basis for the identifier can easily convert them into a URI. According to this, "ark:/12148/bpt6k70861t" identifying a digital premis:representation can easily be converted into a URI by adding the host resolver to it, like <http://ark.bnf.fr/ark:/12148/bpt6k70861t>.
- For DOIs, a HTTP-version has been designed, see http://www.crossref.org/CrossTech/2011/04/content_negotiation_for_crossr.html. Using it instead of the "doi:" identifier converts it to a URI.

If you want to keep your identifiers as character strings, you have 2 options, described below.

IDENTIFIERS IN PREMIS OWL: ALTERNATIVE DESIGNS

The first option is closer to the PREMIS data dictionary but is more verbose:

```
<event1> premis:identifier <event1-ID>.
<event1-ID> a premis:Identifier;
        premis:identifierType "LocalDCMS";
```

premis:identifierValue "E002.1".

The following mechanism instead is more concise and therefore more convenient, but means that you have to define your own properties in a local controlled vocabulary so that you can use them in your assertions:

<event1> someImplementor:localDCMSIdentifier "E002.1".

In the latter case you should declare the following in its own controlled vocabulary that

someImplementor:localDCMSIdentifier *rdfs:subPropertyOf premis:identifier*.

LINKING TO OTHER VOCABULARIES

As you probably have noticed, the PREMIS OWL ontology relies heavily on some SKOS vocabularies. The Library of Congress will publish several controlled vocabularies at http://id.loc.gov/. At this moment, the following vocabularies are published and used in the ontology:

preservationLevelRole: http://id.loc.gov/vocabulary/preservationLevelRole

messageDigestAlgorithm: http://id.loc.gov/vocabulary/cryptographicHashFunctions

eventType: http://id.loc.gov/vocabulary/preservationEvents

In the PREMIS OWL ontology many other properties are also linked to SKOS vocabularies, which will soon be published by the Library of Congress. These vocabularies will be based on the suggested values of the PREMIS 2.1 Data Dictionary. Of course you are still able to include your own vocabularies. For interoperability reasons, these vocabularies should be linked to the LOC vocabularies, as explained in the section below. The LOC vocabularies are also open for suggestions of new terms.

The properties of the PREMIS OWL ontology needing a vocabulary are:

relationship: This property links two objects to each other. The relationship can be derivational or structural. The vocabulary will refine the relationship property with subproperties, specifying the relationship type and subtype, e.g. *isSourceOf*, *isPartOf*, etc.

linkingAgent: This property will link events and right statements to agents. The vocabulary will publish subproperties of the *linkingAgent*, denoting the role of the agent in relation to the event or rights statement, e.g., *linkingCreator*, *linkingGrantor*, etc.

linkingObject: This property will link events and rights statements to objects. The role of the object can be captured by the subproperties of this *linkingObject* property, e.g. *linkingSourceObject*, *linkingOutcomeObject*, etc.

I HAVE MY OWN CONTROLLED VOCABULARY FOR PREMIS VALUES. WHAT SHOULD I DO?

1. If the corresponding vocabulary exists in id.loc.gov, check if one existing term does not match the semantics of your own value.

2. If you think these values are relevant for other implementors to use, you can make a request to id.loc.gov to update the vocabulary (contact URL: http://id.loc.gov/authorities/contact.html)

3. If these values are not relevant, you need to define your own vocabulary locally, and link it to the existing controlled vocabularies / PREMIS ontology as much as possible by using *rdfs:subClassOf* and *rdfs:subPropertyOf* mechanisms.

Two examples:

If you use an implementation-specific identifierType, you will have to declare each of your identifier type as a property, which itself is a *rdfs:subPropertyOf* the *premis:identifier* property. This way, an explicit link is made between your local vocabulary and the PREMIS ontology.

If you use a repository-specific Event, you may need to declare your own someImplementor:SpecificEvent class, and declare it as a *rdfs:subclassOf premis:Event*.

LINKING FORMATS AND SOFTWARE TO THE UDFR AND PRONOM REGISTRIES

With the UDFR and PRONOM every format will get a unique URI. In terms of semantics, this is dealt with in the *FormatRegistry* semantic container, the format URI being specifically referenced under *formatRegistryKey*.

We can either re-express all the Data Dictionary hierarchy, or directly declare the instances of *pronom:file-format* as possible ranges for *premis:format*:

```
<object1ObjectCharacteristics1>  premis:format  <object1Format1>.
<object1Format1>                 a              premis:Format;
        premis:formatRegistry                   <object1FormatRegistry1>.
<object1FormatRegistry1>         a              premis:FormatRegistry;
        premis:formatRegistryName               "PRONOM";
        premis:formatRegistryKey                <http://reference.data.gov.uk/id/file-format/10>;
        premis:formatRegistryRole               "specification".
```

or

```
<object1ObjectCharacteristics1>  premis:format  <http://reference.data.gov.uk/id/file-format/10>.
```

In the latter case, one is not able to express the *formatRegistryRole*.