SRC-AP-VB3: an application profile for metadata assets maintained in VocBench3

Eugeniu Costetchi

13/09/2019

Publications Office of the European Union

Table of Contents

1. Introduction

This objective of this work is to define an Application Profile that can be used for the source code of reference metadata authored with VocBench3. The scope of the application profile is originally intended for, but not limited to, the reference metadata assets managed by the Publications Office of the European Union. *Reference metadata* assets refer to thesauri, taxonomies, authority tables, reference tables, controlled vocabularies, etc. Examples of such assets maintained by the Publications Office are EuroVoc thesaurus, Corporate Body, Language and Country authority lists. The complete list of assets can be found on EU Vocabularies Website.

An *Application Profile* (AP) is a specification that re-uses terms from one or more base standards, adding more specificity by identifying mandatory, recommended and optional elements to be used for a particular application, as well as recommendations for controlled vocabularies to be used.

The Application Profile specified in this document is based on the specification of the *Simple Knowledge Organization System (SKOS)*[1]. SKOS is an *RDF*[2] vocabulary designed to facilitate interoperability between controlled vocabularies published on the Web as Linked Open Data. Additional classes and properties from other well-known vocabularies are re-used where necessary.

The work does not cover implementation issues like mechanisms for edit or publish controlled vocabularies and expected behaviour of systems implementing the Application Profile other than what is defined in the Conformance Statement.

The Application Profile is intended to facilitate controlled vocabularies exchange and therefore the classes and properties defined in this document are only relevant for the controlled vocabularies to be exchanged; there are no requirements for communicating systems to implement specific technical environments. The only requirement is that the systems can export and import data in RDF in conformance with this Application Profile.

2. Context of use

The use cases that this specification intends is to facilitate authoring with VocBench3 of all thesauri and controlled vocabularies managed by the Publications Office of the European Union.

The basic use case involves the following actors:

- EuroVoc team, in charge of the maintenance of EuroVoc thesaurus, and the publication and dissemination of thesaurus maintained by other EU institutions.
- MDR team, in charge of the maintenance and dissemination of the authority tables shared among EU institutions.
- Other teams of the EU institutions who wish to edit their classifications, taxonomies, thesauri and authority tables as Linked Open Data, to facilitate their reuse.

3. Terminology used in the application profile

In the following sections, classes and properties are grouped under headings "mandatory", "recommended" and "optional". These terms have the following meaning.

- *Mandatory class*: a receiver of data MUST be able to process information about instances of the class; a sender of data MUST provide information about instances of the class.
- *Recommended class*: a receiver of data MUST be able to process information about instances of the class; a sender of data MUST provide information about instances of the class, if it is available.

[1] Bechhofer, S., & Miles, A. (2009). SKOS Simple Knowledge Organization System Reference.

[2] Wood, D., Lanthaler, M., & Cyganiak, R. (2014). RDF 1.1 Concepts and Abstract Syntax.

- *Optional class*: a receiver MUST be able to process information about instances of the class; a sender MAY provide the information but is not obliged to do so.
- *Mandatory property*: a receiver MUST be able to process the information for that property; a sender MUST provide the information for that property.
- *Recommended property*: a receiver MUST be able to process the information for that property; a sender SHOULD provide the information for that property if it is available.
- *Optional property*: a receiver MUST be able to process the information for that property; a sender MAY provide the information for that property but is not obliged to do so.

The meaning of the terms MUST, MUST NOT, SHOULD and MAY in this section and in the following sections are as defined in RFC 2119[3].

In the given context, the term "processing" means that receivers must accept incoming data and transparently provide these data to applications and services. It does neither imply nor prescribe what applications and services finally do with the data (parse, convert, store, make searchable, display to users, etc.).

4. Vocabularies used in the application profile

This Application Profile reuses classes and properties from various existing specifications. Classes and properties specified in the next sections have been taken from the following namespaces.

Table 1. List of ontologies and their namespace definitions

Ontology	Prefix	URI
Simple Knowledge Organization System	skos	http://www.w3.org/2004/02/skos/core#
Simple Knowledge Organization System eXtension for Labels	skoxl	http://www.w3.org/2008/05/skos-xl#
DCMI Metadata Terms	dct	http://purl.org/dc/terms/
Dublin Core Metadata Element Set	dc	http://purl.org/dc/elements/1.1/
Publications Office Extensions Ontology	euvoc	http://publications.europa.eu/ontology/euvoc
Lexicon Model for Ontologies	lemon	http://lemon-model.net/lemon
OWL 2 Web Ontology Language	owl	http://www.w3.org/2002/07/owl#
RDF Schema Vocabulary	rdfs	http://www.w3.org/2000/01/rdf-schema#
Resource Description Framework	rdf	http://www.w3.org/1999/02/22-rdf-syntax-ns#
XML Schema Definition	xsd	http://www.w3.org/2001/XMLSchema#

5. Graphical representation of the application profile

The graphical representation of the Application Profile is provided in the form of an UML diagram. It is depicted in Figure ... The boxes represent classes while the arrow connections represent properties establishing

[3] IETF. RFC 2119. Key words for use in RFCs to Indicate Requirement Levels. http://www.ietf.org/rfc/rfc2119.txt

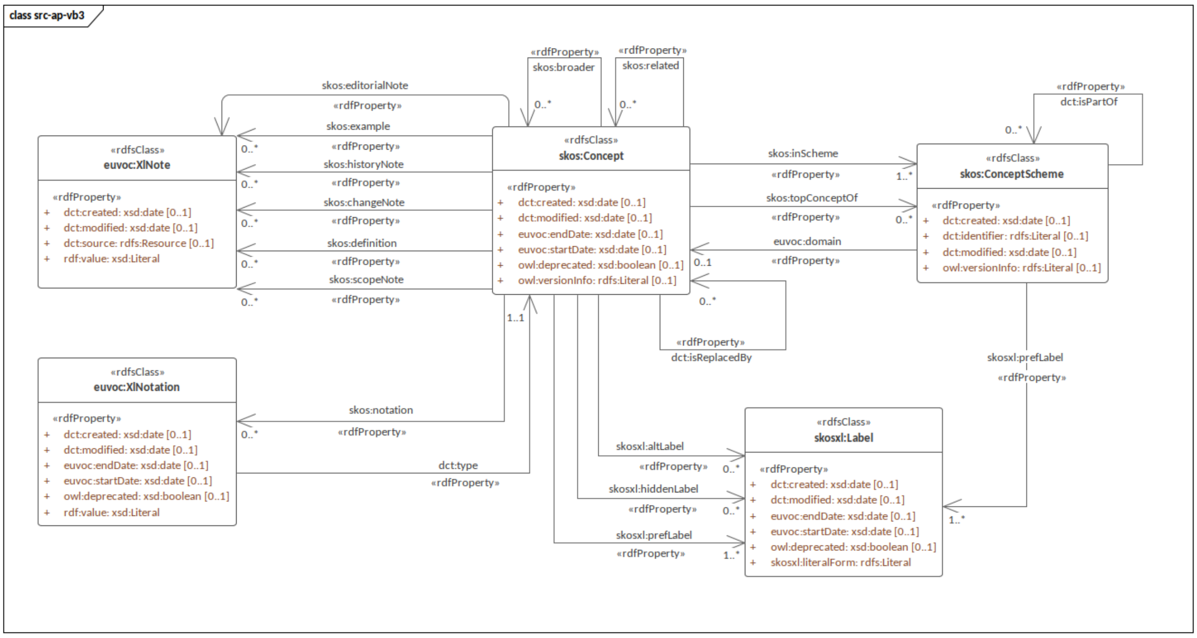
relations to other classes. The attributes inside boxes represent properties providing either literal data values or relation to other classes that omitted from the diagram. Both, arrows and attributes, are labelled with the property names prefixed with the namespace where they are defined.

The cardinality specifications on the connector (_ .. _) arrows and next to the attributes, marked in square brackets ([ ]) represent constraints on how the property may be employed on the class instances and has a normative meaning. The first number means minimum cardinality constraint and the second means maximum cardinality constraint. The minimum cardinality constraint is zero (0 .. _) for optional properties and one (1 .. _) for mandatory properties. The maximum cardinality constraint is usually unspecified (_ .. *) or limited to one (_ .. 1) and has no normative value in this application profile. If the cardinality is not specified then the implied meaning is exactly one (1..1).

The stereotypes, marked in the double angle brackets (<< >>), used in the diagram are based on RDFS model. They have an indicative and not a semantic value. The "mandatory", "optional" and "recommended" stereotypes are normative and have the meaning defined in the terminology section .

In curly brackets ({ }) are provided the value constraints on the property values. These constraints refer to controlled list of values that should meet a minimum set of requirements.

Figure 1. UML class diagram of SRC-AP-VB3 application profile



6. src-ap-vb3.xmi

euvoc:XlNotation

A notation is a string of characters used to uniquely identify a concept within a specified context. Like the skosxl:Label class reifies SKOS label statements, XlNotation reifies SKOS notation statements. This class permits, if needed, to maintain the historical view of the values and add additional provenance descriptions.

Table 2. Properties

Name	Type	Cardinali	Definition
dct:created	xsd:date	0..1	Date of creation of the resource.
dct:modified	xsd:date	0..1	Date of modification of the resource.

Name	Type	Cardinali	Definition
euvoc:endDate	xsd:date	0..1	End of the validity period. If a resource has an end date then it must be marked as deprecated.
euvoc:startDate	xsd:date	0..1	Beginning of the validity period.
owl:deprecated	xsd:boolean	0..1	States whether the resource is current or deprecated. By deprecating a resource, it means that it should not be used in new documents. Deprecation is a feature commonly used in versioning software to indicate that a particular feature is preserved for backward-compatibility purposes, but may be phased out in the future.
rdf:value	xsd:Literal	1..1	The literal form of the notation.
dct:type	skos:Concept	1..1	Specify the context where a specified notation is considered unique.

euvoc:XlNote

Like the skosxl:Label class reifies SKOS label statements, XlNote reifies SKOS note statements (i.e. skos:editorialNote, skos:example, skos:historyNote, skos:definition, skos:scopeNote and skos:changeNote). This class permits, if needed, to maintain the historical view of the values and add additional provenance descriptions.

Table 3. Properties

Name	Type	Cardinali	Definition
dct:created	xsd:date	0..1	Date of creation of the resource.
dct:modified	xsd:date	0..1	Date of modification of the resource.
dct:source	rdfs:Resource	0..1	A related resource from which the described resource is derived. The described resource may be derived from the related resource in whole or in part. Recommended best practice is to identify the related resource by means of a string conforming to a formal identification system.
rdf:value	xsd:Literal	1..1	The literal form of the note.

rdfs:Resource

All things described by RDF are called resources, and are instances of the class rdfs:Resource. This is the class of everything.

rdfs:Literal

The class rdfs:Literal is the class of literal values such as strings and integers. Property values such as textual strings are examples of RDF literals.

skos:Concept

A SKOS concept can be viewed as an idea or notion; a unit of thought. However, what constitutes a unit of thought is subjective, and this definition is meant to be suggestive, rather than restrictive. The notion of a

SKOS concept is useful when describing the conceptual or intellectual structure of a knowledge organization system, and when referring to specific ideas or meanings established within a KOS. Note that, because SKOS is designed to be a vehicle for representing semi-formal KOS, such as thesauri and classification schemes, a certain amount of flexibility has been built in to the formal definition of this class.

Table 4. Properties

Name	**Type**	**Cardinali**	**Definition**
dct:created	xsd:date	0..1	Date of creation of the resource.
dct:modified	xsd:date	0..1	Date of modification of the resource.
euvoc:endDate	xsd:date	0..1	End of the validity period. If a resource has an end date then it must be marked as deprecated.
euvoc:startDate	xsd:date	0..1	Beginning of the validity period.
lemon:context	rdfs:Resource	1..1	Denotes the pragmatic, discursive or technical context of a concept or a constraint on the concept properties.
owl:deprecated	xsd:boolean	0..1	States whether the resource is current or deprecated. By deprecating a resource, it means that it should not be used in new documents. Deprecation is a feature commonly used in versioning software to indicate that a particular feature is preserved for backward-compatibility purposes, but may be phased out in the future.
owl:versionInfo	rdfs:Literal	0..1	An owl:versionInfo statement generally has as its object a string giving information about this version. This statement does not contribute to the logical meaning of the resource.
skos:inScheme	skos:ConceptScheme	1..*	A concept scheme in which the concept is included. A concept may be a member of more than one concept scheme.
skos:definition	euvoc:XlNote	0..*	A statement or formal explanation of the meaning of a concept.
skos:historyNote	euvoc:XlNote	0..*	A note about the past state/use/meaning of a concept.
skos:broader	skos:Concept	0..*	A concept that is more general in meaning. Broader concepts are typically rendered as parents in a concept hierarchy (tree).
skos:example	euvoc:XlNote	0..*	An example of the use of a concept.
dct:isReplacedBy	skos:Concept	0..*	A related resource that supplants, displaces, or supersedes the described resource.
skosxl:altLabel	skosxl:Label	0..*	An alternative lexical label for a resource. Acronyms, abbreviations, spelling variants, and irregular plural/singular forms may be included among the alternative labels for a concept.
skosxl:hiddenLabel	skosxl:Label	0..*	A lexical label for a resource that should be hidden when generating visual displays of the resource, but should still be accessible to free text search operations. Mis-spelled terms are normally included as hidden labels.

Name	Type	Cardinali	Definition
skos:topConceptOf	skos:ConceptScheme	0..*	The property skos:hasTopConcept is, by convention, used to link a concept scheme to the SKOS concept(s) which are topmost in the hierarchical relations for that scheme.
skosxl:prefLabel	skosxl:Label	1..*	The preferred lexical label for a resource, in a given language. No two concepts in the same concept scheme may have the same preferred label in a given language.
skos:notation	euvoc:XlNotation	0..*	A notation is a string of characters such as "T58.5" or "303.4833" used to uniquely identify a concept within the scope of a given concept scheme or within a specified context.
skos:related	skos:Concept	0..*	A concept with which there is an associative semantic relationship.
skos:changeNote	euvoc:XlNote	0..*	A note about a modification to a concept.
skos:editorialNote	euvoc:XlNote		A note for an editor, translator or maintainer of the vocabulary.
skos:scopeNote	euvoc:XlNote	0..*	A note that helps to clarify the meaning of a concept.
skos:broader	skos:Concept	0..*	A concept that is more general in meaning. Broader concepts are typically rendered as parents in a concept hierarchy (tree).
dct:isReplacedBy	skos:Concept	0..*	A related resource that supplants, displaces, or supersedes the described resource.
skos:related	skos:Concept	0..*	A concept with which there is an associative semantic relationship.

skos:ConceptScheme

A SKOS concept scheme can be viewed as an aggregation of one or more SKOS concepts. Semantic relationships (links) between those concepts may also be viewed as part of a concept scheme. This definition is, however, meant to be suggestive rather than restrictive, and there is some flexibility in the formal data model stated below. Thesauri, classification schemes, subject heading lists, taxonomies, 'folksonomies', and other types of controlled vocabulary are all examples of concept schemes. Concept schemes are also embedded in glossaries and terminologies.

Table 5. Properties

Name	Type	Cardinali	Definition
dct:created	xsd:date	0..1	Date of creation of the resource.
dct:identifier	rdfs:Literal	0..1	An unambiguous reference to the resource within a given context. Recommended best practice is to identify the resource by means of a string conforming to a formal identification system.
dct:modified	xsd:date	0..1	Date of modification of the resource.

Name	Type	Cardinali	Definition
owl:versionInfo	rdfs:Literal	0..1	An owl:versionInfo statement generally has as its object a string giving information about this version. This statement does not contribute to the logical meaning of the resource.
euvoc:domain	skos:Concept	0..1	Indicates the subject of the controlled vocabulary. This property has a similar function as the dct:subject and dcat:theme.
dct:isPartOf	skos:ConceptScheme	0..*	A related resource in which the described resource is physically or logically included.
skosxl:prefLabel	skosxl:Label	1..*	The preferred lexical label for a resource, in a given language. No two concepts in the same concept scheme may have the same preferred label in a given language.
dct:isPartOf	skos:ConceptScheme	0..*	A related resource in which the described resource is physically or logically included.

skosxl:Label

The class skosxl:Label is a special class of lexical entities. An instance of the class skosxl:Label is a resource and may be named with a URI. An instance of the class skosxl:Label has a single literal form. This literal form is an RDF plain literal (which is a string of UNICODE characters and an optional language tag [

RDF-CONCEPTS

]). The property skosxl:literalForm is used to give the literal form of an skosxl:Label. If two instances of the class skosxl:Label have the same literal form, they are not necessarily the same resource.

Table 6. Properties

Name	Type	Cardinali	Definition
dct:created	xsd:date	0..1	Date of creation of the resource.
dct:modified	xsd:date	0..1	Date of modification of the resource.
euvoc:endDate	xsd:date	0..1	End of the validity period. If a resource has an end date then it must be marked as deprecated.
euvoc:startDate	xsd:date	0..1	Beginning of the validity period.
owl:deprecated	xsd:boolean	0..1	States whether the resource is current or deprecated. By deprecating a resource, it means that it should not be used in new documents. Deprecation is a feature commonly used in versioning software to indicate that a particular feature is preserved for backward-compatibility purposes, but may be phased out in the future.
skosxl:literalForm	rdfs:Literal	1..1	The literal form of an skosxl:Label. An instance of the class skosxl:Label has one and only one literal form.

xsd:boolean

The boolean data type is used to specify a true or false value.

xsd:date

The date data type is used to specify a date. The date is specified in the following form "YYYY-MM-DD" where:

- YYYY indicates the year
- MM indicates the month
- DD indicates the day

Note: All components are required!

7. Requirements for controlled vocabularies

The following is a list of requirements for the controlled vocabularies that are used in this application profile as controlled list of values.

Controlled vocabularies should:

- Be published under an open licence.
- Be operated and/or maintained by an institution of the European Union, by a recognised standards organisation or another trusted organisation.
- Be properly documented.
- Have labels in multiple languages, ideally in all official languages of the European Union.
- Have terms that are identified by dereferenceable(4) URIs with each URI resolving to descriptions about the term.
- Have associated persistence and versioning policies.

These criteria have been applied to select controlled vocabularies used in this application profile.

8. Used controlled vocabularies

This section provides the list of controlled vocabularies that should be used as values of the corresponding properties.

(4) Dereferencing HTTP URIs (https://www.w3.org/2001/tag/doc/httpRange-14/2007-05-31/HttpRange-14)