Towards executable application profiles for European vocabularies

Eugeniu Costetchi and Willem van Gemert

Publications Office of the European Union

October 9, 2016

Abstract

This paper describes current work done at the Publication Office of the European Union in the area of automatic validation of controlled vocabularies using a SHACL implementation of application profiles (AP) such as SKOS-AP-EU. The same implementation serves as a source for generating human readable AP documentation.

The Publications Office of the EU is an interinstitutional office whose task is to publish the publications of the institutions of the European Union. Its core activities include production and dissemination of legal and general publications in a variety of paper and electronic formats, managing a range of websites providing EU citizens, governments and businesses with digital access to official information and data from the EU, including EUR-Lex, the EU Open Data Portal, EU Bookshop and TED (Tenders Electronic Daily), and ensuring long-term preservation of content produced by EU institutions and bodies. The Publications Office plays an active role in the metadata standardisation domain and provides access to its different interoperability solutions in its Metadata Registry (MDR)[1]. The MDR contains reference data assets (ontologies, named authority lists, XML schemas, application profiles, etc.) used by the different European Institutions.

Linked Open Data (LOD) is becoming increasingly a de-facto standard and a set of practices in the data publishing world mainly thanks to a rich set of standards (RDF-S[2], OWL[3], etc.), widely accepted ontologies (Dublin Core Terms[4], DCAT[5], SKOS(-XL)[6], Schema.org[7], etc.) and a strong community of practice.

[1] http://publications.europa.eu/mdr/
[2] https://www.w3.org/TR/rdf-schema/
[3] https://www.w3.org/TR/owl2-overview/
[4] http://dublincore.org/documents/dcmi-terms/
[5] https://www.w3.org/TR/vocab-dcat/
[6] https://www.w3.org/TR/skos-reference/skos-xl.html
[7] http://schema.org/

In order to enable interoperability and to promote wide usability of published data, the MDR has developed three application profiles (AP):

1. SKOS-AP-EU to shape thesauri and controlled authority lists,
2. DCAT-AP-OP for dataset descriptions in the EU Open Data Portal[8] context
3. ORG-AP-OP to control how the EU institutions/bodies are described and disseminated through the EU directory, the EU Whoiswho[9].

As APs serve mainly in the role of publishing and as exchange standard, common practice is to describe the APs as textual or tabular documents. They are further used as requirements for implementing the systems that generate or use the data.

This paper shortly explains the recent work at the MDR on expressing APs as SHACL shapes[10]. Doing so has two benefits: first, it allows automatic validation of data and second, the formal SHACL source serves as basis for automatic generation of human readable documentation.

Next we describe the MDR context and how the current work is motivated, the possible considered alternatives along with the rationale for choosing SHACL as the AP language and finally a few insights into what SKOS-AP-EU covers.

The need for validation arose in the first place because the authority lists (AT) data is currently maintained in XML format which is conform to the CAT XML schema[11]. Through a series of XSLT rules the data is transformed into RDF/XML format intending to correctly express RDF syntax and RDFS/OWL model instantiation semantics. Since it is a purely syntactic process there is a strong need to check whether the resulting data correctly instantiate the conceptual models and if it is according to the defined application profile.

Normally RDFS and OWL, the traditional languages to express conceptual models for RDF data, can be used to augment assertions during the query processing i.e. SPARQL under the RDFS and OWL entailment rules. However due to the open world assumption the models cannot be used for validation except in extreme cases of inconsistent data checked by standard reasoners.

SPARQL queries are suitable for validation and integrity checks, but such queries are not intuitive and thus difficult to maintain and construct. The language is designed in a way to serve the best data retrieval and selection needs, but not constraint validation. There are several languages that allow expressing and applying rules to RDF graphs: SPIN[12], ShEx[13], SHACL, SWRL[14], RIF[15]. From the set of available alternatives we evaluated and selected the ones that best suit our needs in terms of intuitiveness and expressiveness, with well defined semantics, allowing reaction descriptions (e.g. in case of a constraint violation) and, very important, with a decent implementation and tool support.

[8] http://data.europa.eu/euodp/en/data/
[9] http://europa.eu/whoiswho/public/index.cfm?lang=en
[10] https://www.w3.org/TR/shacl/
[11] http://publications.europa.eu/mdr/resource/documentation/schema/cat.html
[12] http://spinrdf.org/
[13] https://www.w3.org/Submission/shex-defn/
[14] https://www.w3.org/Submission/SWRL/
[15] https://www.w3.org/TR/rif-overview/

Specifically for the purpose of expressing and executing APs we have chosen SHACL, a successor of ShEx developed within the RDF Shapes Working Group[16], designed to express RDF constraints (called shapes). It is designed in an object oriented paradigm and allows embedded SPARQL expressions at different levels of shape definition which leads to highly expressive language. Nevertheless it has some limitations such as lack of support for named graphs or RDFS datasets directly. No straight forward mechanism for modularization and separation of validation constraints and model semantic axioms. Neither it has any constructive capabilities as for example in the case of SPIN that would allow detection and automatic correction of errors in the data. The sole purpose of this language is a versatile validation and integrity constraint checking.

Next we provide a few examples from SKOS-AP-EU, an application profile covering the core shapes for authority tables and thesauri (including EuroVoc[17]). There are also non-core shapes that are domain specific and vary from one authority table to another. For example spatial tables such as Countries, Places, Administrative Territorial Units (ATU) etc. use a spatial AP on top of SKOS-AP-EU.

SKOS-AP-EU is based primarily on the SKOS-XL model augmented with a few DCT properties. So for example the *skos:Core* class is instantiated with mandatory *skos:prefLabel*, *skos-xl:prefLabel* and *dct:created* properties.

In the authority tables each concept has a validity period. The labels attached to each concept also have validity periods. This means that a label may be used as preferred for a period of time, and then when a new preferred label is provided the old one becomes alternative and receives an end usage date at the same time.

A similar need to express start and end use dates exists for notations (i.e mappings to other contexts), notes (i.e. scope, editorial, history notes and definitions), and other SKOS properties. Because there are no classes in standard models representing the reified properties, they had to be defined. This is how the Euvoc ontology was born. It defines classes and properties that fulfil specific MDR needs which by design are not covered in the standard models. For example *euvoc:xlDefinition* property and *euvoc:XlNote* classes are used to express reified concept definitions carrying start and end use dates.

SKOS-AP-EU defines minimum and maximum cardinality constraints making properties on each class either mandatory, optional and/or sufficient. This mechanism ensures that there is exactly one start use date and maximum one optional end use date.

Another type of constraint is on property ranges limiting the possibility to either a controlled list of values, a certain class or data type. For example the *foaf:locality* need to take values from the Places authority table or the *dct:created* need to be of *xsd:date* data type.

We use property coordination constraints expressing if p_1 is used then p_2 must also be used. For example if there is an end date then there necessarily must be a start date. A similar relationship can be defined at the level of property cardinality or values. For example a start use date cannot be after the end use date of a concept; or a license cannot state share-alike requirement and permission to modify the work because that would lead to a contradiction.

[16] `https://www.w3.org/2014/data-shapes/charter`
[17] `http://eurovoc.europa.eu/drupal/`

We also use more complex constraints. For example SKOS constraint that the preferred label occurs exactly once per language or that the label *dct:type* for preferred labels is "standard" and not "short", "long", "acronym" or other.

SHACL allows us to express all the above mentioned constraint types and many others. Due to space limitation we do not cover other features such as shape generative composition, inheritance, scoping and other. To execute the dataset validation we have created a command line wrapper around the open source SHACL API offered by TopBraid.

Before concluding, we would like to mention also the fact that we generate human readable HTML documentation from the formal SHACL statements. This documentation covers exactly the same information and in the same form as the original tabular AP, i.e. for each class as set of expected properties with corresponding cardinality constraints and eventually the range limitation to a certain controlled list (e.g status).

This paper briefly describes the work done at the Metadata Registry (MDR) with regard to dataset validation using executable SHACL implementations of application profiles. The current approach can be replicated for any other AP and of course the practice can be further developed and refined. The benefits of using SHACL to express APs is the perfect alignment of the AP purpose and needs with intuitive and expressive language design. Moreover the formal expression of constraints not only allows automatic verification but also generation of human readable documentation of the AP.