RDF structure of PoolParty vocabularies

WORK IN PROGESS: NOT READY FOR PUBLICATION!

This page explains the RDF structure of vocabularies created in PoolParty, and the transformations that are applied to the RDF when it is imported into RVA.

Vocabulary representation as RDF in PoolParty

PoolParty's convenient user interface hides much of the complexity of representing a vocabulary in RDF. However, you or your users need to know about this representation if you wish to use SPARQL to run queries.

This page explains the cross-cutting concerns, and shows how PoolParty makes use of specific well-known RDF vocabularies.

Languages

During project creation, you specify one or more project languages, of which one must be designated as the "default language". The PoolParty user interface does not allow you to change the default language of a project once it has been created. However, you may subsequently add or remove other languages.

Almost all textual data you add to a project is added as an RDF "language-tagged string", i.e., as a string with an associated language code (RDF type `http://www.w3.org/1999/02/22-rdf-syntax-ns#langString`). See https://www.w3.org/TR/rdf11-concepts/ for further information about language-tagged strings. You can see this in practice by viewing a concept and then selecting the "Triples" tab. For example, every preferred label is shown with a language tag.

Some concept properties are added *without* a language tag. For example, if you add a SKOS notation, the value is added as an RDF "simple literal" (RDF type `xsd:string`). Conversely, the PoolParty user interface does not support adding language-specific SKOS notations for a concept.

Use of well-known RDF vocabularies

PoolParty creates and manages (user-defined) vocabularies using classes and properties drawn from a number of well-known RDF vocabularies; in particular:

- SKOS
- OWL
- RDF Schema (RDFS)
- Dublin Core Terms
- ADMS
- VoID

The following sections explain briefly how each of these vocabularies is used by PoolParty.

SKOS

References:

- SKOS Primer: https://www.w3.org/TR/skos-primer/
- SKOS Reference: https://www.w3.org/TR/skos-reference/

SKOS is used as the overarching structure of vocabularies in PoolParty. The PoolParty user interface presents vocabularies as consisting of "concept schemes" which contain "concepts"; these are represented directly as `skos:ConceptSchemes` and `skos:Concepts`.

Please note the information given in the "Languages" section above. For example: preferred labels (and other types of labels) are stored as language-tagged strings; notations are stored as simple literals.

OWL

Reference:

- OWL 2 Quick Reference Guide: http://www.w3.org/TR/owl2-quick-reference/

The SKOS data model is itself an "OWL Full ontology" (see the SKOS Reference), and so, for example, every SKOS Concept is an instance of `owl:Class`. Because the SKOS vocabulary provides almost all the necessary support for describing a vocabulary, PoolParty makes very little use of OWL properties. The one notable instance is in the description of deleted concept schemes and concepts. Such resources are annotated using the property `owl:deprecated`, with the property assigned the value `true`.

RDF Schema (RDFS)

Reference:

- RDF Schema: https://www.w3.org/TR/rdf-schema/

The only element of the RDFS vocabulary used by PoolParty is `rdfs:label`. This property is used to assign titles to concept schemes. [Note that the SKOS Reference explicitly states (https://www.w3.org/TR/skos-reference/#L1541) that no domain is specified for `skos:prefLabel`, and so PoolParty could use `skos:prefLabel` to assign titles to concept schemes – but it doesn't.] Note that a concept scheme's title is assigned in two ways: using both `rdfs:label` and `dcterms:title`.

Dublin Core Terms (DC Terms)

References:

- Definition: http://dublincore.org/documents/dcmi-terms/
- User guide: http://wiki.dublincore.org/index.php/User_Guide/Publishing_Metadata
  - No longer available at that URL: please use this for now: https://github.com/dcmi/repository/blob/master/mediawiki_wiki/User_Guide/Publishing_Metadata.md

The Dublin Core Terms vocabulary is used to assign a variety of metadata properties to concepts schemes and concepts. The following properties are used:

- `dcterms:title` (used to assign titles to concept schemes, in addition to the use of `rdfs:label`)
- `dcterms:description`
- `dcterms:subject`
- `dcterms:creator`
- `dcterms:publisher`
- `dcterms:contributor`
- `dcterms:created`
- `dcterms:modified`

In versions of PoolParty prior to Release 5.5, values of some of these properties violate the formal definition of those properties. To be specific: the properties `dcterms:creator`, `dcterms:publisher`, and `dcterms:contributor` are defined by DCMI to have range `dcterms:Agent` (see, e.g., http://purl.org/dc/terms/publisher), which in practice means that values of these properties must be resources, not literals (see, e.g., http://wiki.dublincore.org/index.php/User_Guide/Publishing_Metadata#dcterms:publisher; no longer available at that URL: please use this for now: https://github.com/dcmi/repository/blob/master/mediawiki_wiki/User_Guide/Publishing_Metadata.md). In releases before 5.5, PoolParty assigns string literals to these properties. In releases from 5.5 onwards, PoolParty assigns resources, and these resources in turn have a property (in SWC's own namespace) that assigns a string literal.

Some of these DC Terms properties (`dcterms:title`, `dcterms:created`, and `dcterms:modified`, as well as `dcterms:identifier` and `dcterms:license`) appear in the ADMS data (see below).

ADMS

Reference:

- Phil Archer's draft of ADMS, which corresponds to the version of the ADMS vocabulary used by PoolParty: http://philarcher.org/adms/
- W3C Working group note: https://www.w3.org/TR/vocab-adms/ (a later revision of the above)
- PoolParty documentation: https://help.poolparty.biz/doc/user-guide-for-knowledge-engineers/basic-features/working-with-ppt-projects/project-details/project-data-stored-as-adms-graph

PoolParty makes some limited use of ADMS, although in doing so, it does not follow the W3C's final Working group note, but rather, an earlier, draft version (see the references given above). Within each project, it defines a resource of type `adms:SemanticAsset`, and assigns various ADMS-related properties to that resource.

You typically don't get to see the ADMS data in a vocabulary whose data has been published through RVA. However, ADMS data is used to help "pre-populate" some of the fields contained in the form you see when you use RVA's "Add a new vocabulary from PoolParty".

VoID

Reference:

- W3C Interest group note: https://www.w3.org/TR/void/
- PoolParty documentation: https://help.poolparty.biz/doc/user-guide-for-knowledge-engineers/basic-features/working-with-ppt-projects/project-details/project-data-stored-as-void-graph

Within each project, a resource of type `void:Dataset` is defined, and various VoID-related properties (mostly drawn from DC Terms) are assigned to that resource.

You typically don't get to see the VoID data in a vocabulary whose data has been published through RVA. However, VoID data is used to help "pre-populate" some of the fields contained in the form you see when you use RVA's "Add a new vocabulary from PoolParty".

Custom schemes

PUT SOMETHING HERE. Discuss use of properties from custom schemes.

Named graphs

The RDF triples contained within each project constitute an RDF Dataset, and that dataset is stored in an individual repository within a triple store. The triple store of ARDC's PoolParty server is an instance of OpenRDF Sesame.

For each project, the RDF Dataset containing the project's RDF triples is partitioned into a number of graphs. In versions of PoolParty prior to Release 5.5, the project's main "content" – its concept schemes and concepts – is contained in the default graph. In releases from 5.5 onwards, it is stored in a named graph.

In versions of PoolParty prior to Release 5.5, the IRIs of a project's named graphs are exactly the same across all projects. (For example, each project's ADMS data is contained in a named graph with IRI `<http://www.w3.org/ns/adms>` .) From Release 5.5, the IRI of each named graph is determined in part by the project's "Base URL" (also known as "base URI") and "Project Identifier" (also known as "URI supplement"), which are entered during project creation in the "Advanced" tab. The IRIs of the named graphs are listed at the Release notes for PoolParty 5.5.

You typically don't have to pay attention to named graphs; RVA takes care of the details for you when necessary. How this applies during importing a vocabulary is explained in the next section. If you use the ARDC Vocabulary Editor Admin Tool, the predefined queries and updates are automatically adjusted to use the correct named graphs before they are sent to the PoolParty server.

RDF transformation during import

RVA provides convenient support for importing and publishing a vocabulary you have created in the ARDC PoolParty server. The process for doing this is described at the page Publishing a PoolParty vocabulary to the RVA portal. It is important to note that in order to import any vocabulary data into RVA, you must specify the creation of a *version* (using the "Add a version" button) as follows:

- The version's status may be either "current" or "superseded".
- The version must have access points generated from PoolParty: on the "Add a new version" dialog, click the toggle to select "Harvest version from from PoolParty".

The following sections describes how the RDF data stored in PoolParty is transformed into the RDF data presented through RVA.

Harvest

The first step in the process is to fetch the vocabulary from data; within RVA, this is known as "harvesting". RVA uses PoolParty's own API to export the vocabulary data; it fetches both current and deleted/deprecated concepts.

There were major changes in the representation of vocabulary data in PoolParty Release 5.5. The harvest process was modified so as to preserve the "end result" achieved with earlier releases. In particular, Releases from 5.5 onwards make extensive use of named graphs. Therefore, the harvester invokes the PoolParty export API in such a way that the assignment of triples to named graphs is discarded: the triples are exported in a format that represents triples as belonging only to the default graph.

Another change in Release 5.5 was to the representation of creator and contributor data represented using DC Terms. In earlier releases, the objects of triples with property `dcterms:creator` and `dcterms:publisher` were string literals; they are now resources. During harvest, those resources are transformed so as to have a triple of type `foaf:name` with object that is the (string) value you typed in.

Import

The next step in the process is to import the harvested vocabulary data into an OpenRDF Sesame repository. The content of each of the harvested files is uploaded into the repository. All imported triples are placed into the default graph.

Publish

In this context, "publish" means to make the vocabulary available through the Linked Data API (also known as SISSVoc). During this step, a set of SISSVoc endpoints are created, each of which queries the OpenRDF Sesame repository created during the import step.