EJP RD
European Joint Programme on Rare Diseases

H2020-SC1-2018-Single-Stage-RTD
SC1-BHC-04-2018
Rare Disease European Joint Programme Cofund

Grant agreement number 825575

Del 11.7
First update
Virtual platform of RD resources annotated with EJP ontological model

Organisation name of lead beneficiary for this deliverable:
Partner 01 – INSERM-Orphanet

Contributors:
AMC, BBMRI-ERIC, ELIXIR/EMBL-EBI, GUF, LUMC, UMCG, UPM

Due date of deliverable: month 24

Dissemination level:
Public

Table of Contents

1. Introduction

During Year 1, based on the first version of the EJP RD Resources Metadata Model, a first implementation of a common model to capture descriptions of several catalogues was produced and made machine readable (Deliverable 11.6). Catalogue providers (RD-Connect registries and biobanks Finder, BBMRI, JRC-ERDRI, Orphanet) described and exposed their metadata elements in a standardized way. This integration was also provided in an operational "Linked Data Platform" (LDP).

Grounded from this previous work, during Year 2 (Y2) the first implementation work was reshaped according to the evolution of the EJP RD Metadata Model (Deliverable 11.6), aligning with the new DCAT 2.0 standard released in February 2020. Furthermore, the methodology to leverage the abilities to incorporate new sources of data was improved, including the Pillar 4 [Work Package 19 (WP19)] Innovation Management Toolbox (IMT) and applying this methodology to ECRIN and EATRIS metadata.

2. Approach

During Y2, according to the annual work plan, the aim was to expand the implementation of the EJP RD Metadata Model at resource level, i.e., a model to expose structured, machine readable information about a container of content (the resource) to make it Findable, and the metadata Accessible, Interoperable and

Reusable.
First, the consistency of the model was checked, the adaptation to the current catalogues' sources regarding the needed update of the model itself was performed. A methodology ensuring that resources that would be added in the future to the Virtual Platform (VP) are mapped with the resources metadata model was created. Following the updated version of the Resources Metadata Model described under Deliverable 11.2, the built-in extendibility of the DCAT 2.0 model is now exploited. Thus, any extension of the model, that is further needed to describe a new resource to be added, is subsumed under one of DCAT's standard classes or properties. Standardised mapping descriptions, from a resource to the model, are used inorder to be fully compliant with DCAT 2.0. This approach will ensure incremental integration of new resources.

Adaptations to the preliminary design of the EJP RD VP were proposed, pertaining to incorporation of semantic components that make use of the metadata model (e.g., the FAIR data point, LDP) and thereby meet FAIR principles, and alignment with initial Application Programme Interface (API) designs to support the use of the EJP RD Resources Metadata Model.
In cross Work Foci (WF) efforts (both metadata WF and query builder WF) several services related to this work were released.

2.1. Workshop

In spring 2020, a virtual workshop was organised to discuss various issues identified in year 1. During this virtual workshop, some of the modelling issues in the metadata model were addressed which led to use DCAT 2.0 and an extension of DCAT 2.0. The general implementation design for the metadata model was brainstormed as well.

2.2. Update to the designed methodology

The EJP RD resources metadata model is based on ontologies and semantic web technology. RDF (Resource Description Framework, a World-Wide-Web-Consortium recommendation) is used to implement this model. RDF is a common data model for ontologies and ontologized metadata and therefore convenient for general management of metadata within the VP. It reuses web hyperlink technology to scale up easily to global levels, RDF data can be exchanged in several interchangeable file formats, including JSON and 'turtle'. Figure 1 shows the metadata implementation design that was used in year 1 to get RDF files for resource descriptions from the catalogues. The method 1 of figure 1 shows how the resources metadata model is implemented in a catalogue with JSON (JavaScript Object Notation) objects of resource descriptions as an intermediate step. In method 2, the catalogue owner generates RDF (turtle) files based on the metadata model. The generated RDF files are then published in the project specific LDP. However, towards the end of year 1, various limitations with this design came across. In the content generation process, especially in method 1, it was not clear to catalogue owners which artifacts they must update in order to update their resource descriptions in the LDP. Query limitations of the LDP itself arose as well. These issues motivated the improvement of the general implementation design for the metadata model. Figure 2 shows the updated design. The resource description generation and consumption parts were separated. The catalogue

owners can choose to use either RDF or the metadata model compliant JSON schema to describe their resources. These artifacts are then published into the LDP that was setup for the EJP RD. The users of these resources then consume these artifacts from the LDP and store them in an RDF database (so-called a 'triple store', because RDF consists of subject-predicate-object triples). The proposal to use a triple store allows to overcome the query limitation issues in the LDP. In the proposed design it is the catalogue managers' responsibility to update the resource descriptions and publish them in the LDP, and the consumer's responsibility to update the triple store whenever updates of the resources are available in the LDP. How the process will be managed for users of the EJP RD VP will be investigated as part of developing the VP specifications.

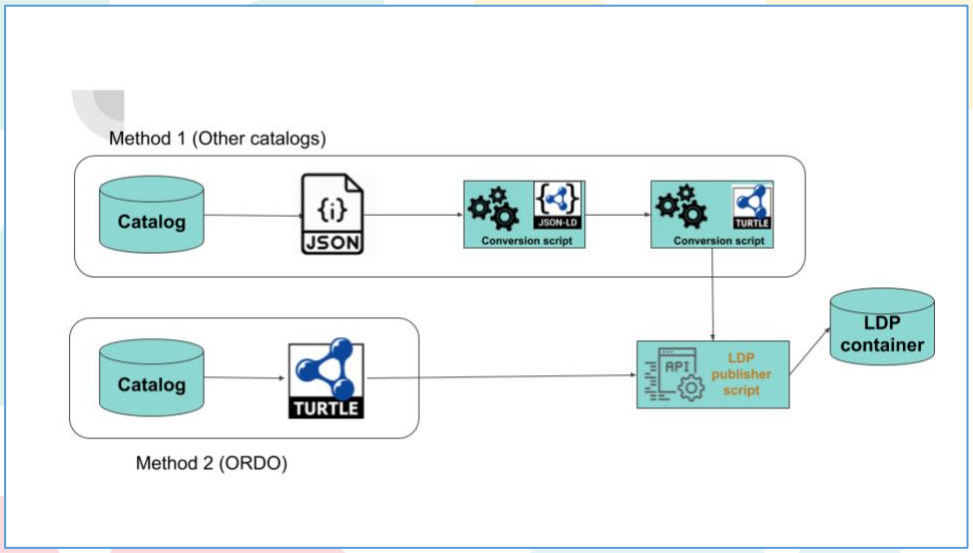

Figure 1. Metadata implementation design year 1

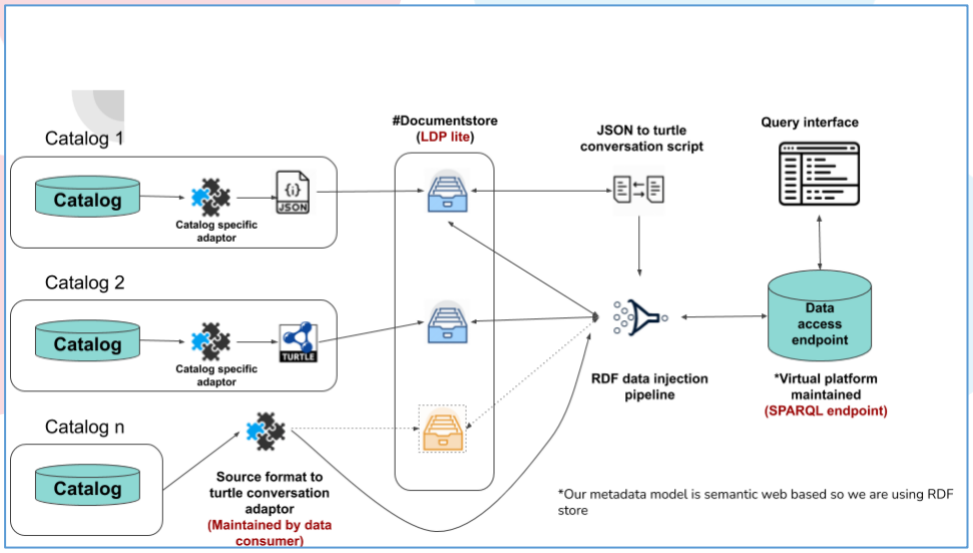

Figure 2. Proposed metadata implementation design

3. Methodology applied to Innovation Management Toolbox (IMT)

3.1. Data preparation

This work was performed in collaboration with WP19 partners. During the series of teleconferences organised by WP19 partners, an excel template capturing the IMT metadata description was jointly developed. Figure 3 shows the headers of this template that are well aligned with the properties of the metadata model. The WP19 partners used this template to capture the metadata of IMT resources. During the data capturing process feedbacks on the quality of the captured metadata content were provided.

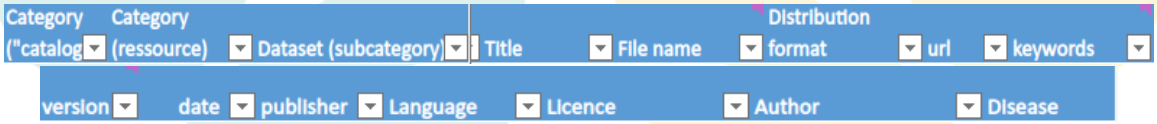

Figure 3. Headers of the excel template used to capture IMT resource descriptions

3.2. Resource metadata model extension

The resources described by IMT are of type 'Regulatory documents'. It was agreed with WP19 partners that the most common 'type' definition for these resources is 'Guideline'. The DCAT 2.0 *Resource* class was extended with a 'Guideline' sub class and the *part of* property with 'guideline' sub property. The figure 4 shows the extension made to the DCAT 2.0 vocabulary. To increase the interoperability of the custom DCAT 2.0 extension, annotation with external class mappings to the extended classes was also performed whenever it is possible. The figure 5 shows the mapping annotation of guideline class.

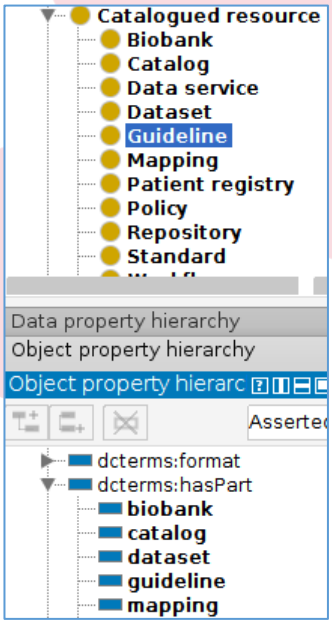

Figure 4. DCAT 2.0 extension with other types of resources

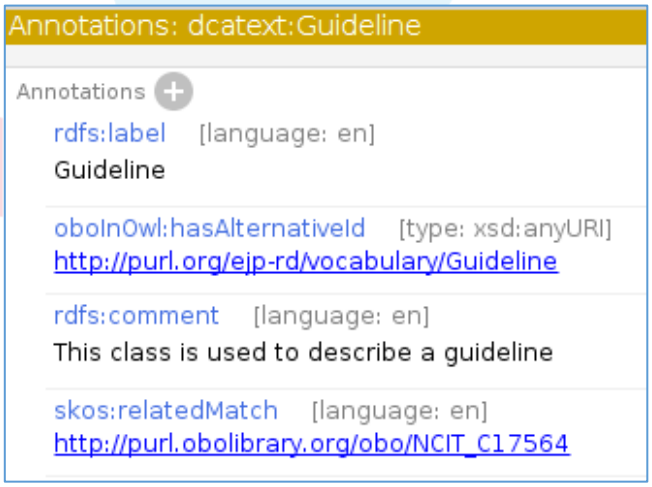

Figure 5. Mapping annotation for `Guideline` class

3.3. Populating the FAIR data point

A pipeline was developed to convert and store IMT metadata in machine readable RDF and an access point conform to the FAIR data point (FDP) software specification was implemented to publish the metadata (figure 6). The FDP is software that allows to describe resources and interrogate the description via a REST API. Its specification follows the DCAT 2.0 standard. By default, the FDP only allows to describe 'Dataset' resources. However, the FDP can be configured to describe other types of resources. For the IMT use case, the FDP was configured to describe 'Guideline' resources. Figure 6 shows the pipeline to populate the FDP with IMT resources. As a first step IMT descriptions were extracted from the Excel sheet provided by WP19 partners. The Excel content was converted into RDF which is compatible with the resource metadata model, then the RDF was stored via the FDP by using the API provided by the FDP software. The database that stores the IMT metadata in RDF provides the W3C-recommended SPARQL query language for RDF. The FDP describes if and how the database is externally accessible. The FDP instance set up for the IMT resources can be accessed via this link http://ejprd.fair-dtls.surf-hosted.nl:8090/.

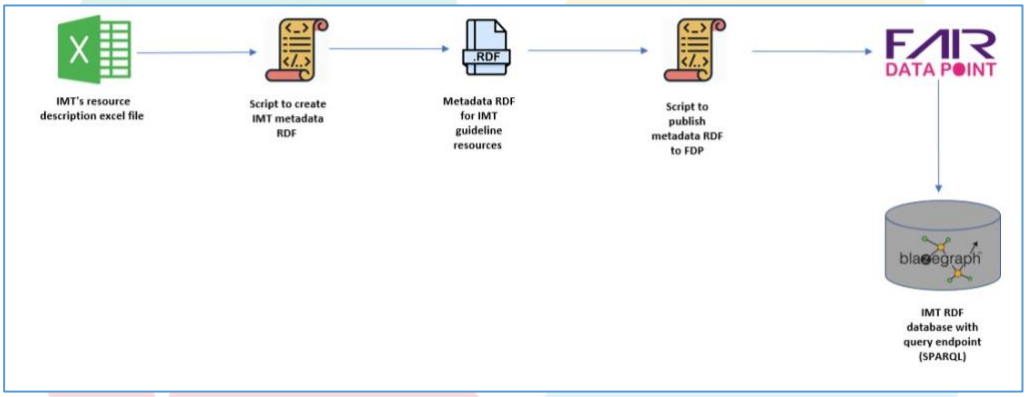

Figure 6. IMT resources population pipeline

4. Planned methodology for ECRIN and EATRIS

The European Infrastructure for Translational Medicine (EATRIS) has a portfolio of services including expertise and platforms to foster translational research and innovation. It can be seen as a catalogue of resources and services to be made discoverable through the Virtual Platform.

The European Clinical Research Infrastructure Network (ECRIN) has an online database including country-specific information on regulatory and ethical requirements in clinical research across Europe. ECRIN provides the CAMPUS Database that encompasses up-to-date regulatory and ethical requirements applicable to clinical research projects.

ECRIN maintains a Clinical Research Metadata Repository (MDR) as well, (CRMDR.org) designed in a FAIR manner (https://ecrin-mdr.online/index.php/Project_Overview)
CRMDR.org is currently under refactoring and will be processed during next phases.

(screen capture https://ecrin.org/tools/clinical-research-metadata-repository Dec. 2020)

The CAMPUS database is a catalogue of regulatory information and contents, described in categories (Medical Devices, Medicinal Products for Human Use, Nutrition) with subcategories and defined metadata concepts (Competent Authority, Ethics Committee, Study Specific Requirements, National Legislation, Definition) and related subsets.

Therefore, the same methodology as applied to IMT is applicable to these resources: extension of DCAT 2.0 EJP RD metadata model and further integration into the common model.

A metadata model implementation task for ECRIN and for EATRIS is planned to be carried out in early 2021. For these resources the same approach as that used for the IMT resource will be applied. Teleconferences will be organised to capture ECRIN and EATRIS resources description in an excel file, based on the IMT excel template. These descriptions will then be published on a dedicated FDP of ECRIN and EATRIS.

5. Federated endpoint

The FDP software was used to store descriptions of the catalogues for several reasons. The resource descriptions in the FDP are captured and thereby accessible via the DCAT 2.0 vocabulary, which is defined in RDF and therefore easy to integrate with ontologised metadata. The FDP could be easily configured to use the EJP RD specific metadata model. The FDP also implements some of the LDP container features, again using the extendibility of RDF. Finally, the FDP can be configured to store the resources descriptions to the triple store. Links to these FDP instances of these catalogues are listed below.

FAIR data point instance of Orphanet catalogue:

- http://ejprd.fair-dtls.surf-hosted.nl:8084

FAIR data point instance of BBMRI-ERIC catalogue:

- http://ejprd.fair-dtls.surf-hosted.nl:8082

FAIR data point instance of IMT catalogue:

- http://ejprd.fair-dtls.surf-hosted.nl:8090

Although, it is already possible to do a federated query over these FDPs via their SPARQL endpoints (SPARQL is a use-case independent query language for RDF), in the EJP RD a query API was implemented on top of these SPARQL endpoints to provide the VP with an additional interface for discovery queries. The specification of this API is provided by the Query Builder team within the EJP RD. Figure 7 shows the design used to implement this API specification. FedX was used to facilitate the federation of the query across different SPARQL endpoints. The FedX framework was configured to query relevant SPARQL endpoints to be used for the API calls. In this configuration, the SPARQL endpoint of the ORDO (Orphanet Rare Diseases Ontology) was also included ensuring that the query will also make use of disease knowledge from the ORDO. The resultant API implementation basically translates EJP RD specific API calls to SPARQL queries. Then the SPARQL query results are translated into JSON objects specified by the EJP RD Query Builder team. This resultant API implementation can also be used by the EJP RD query portal.

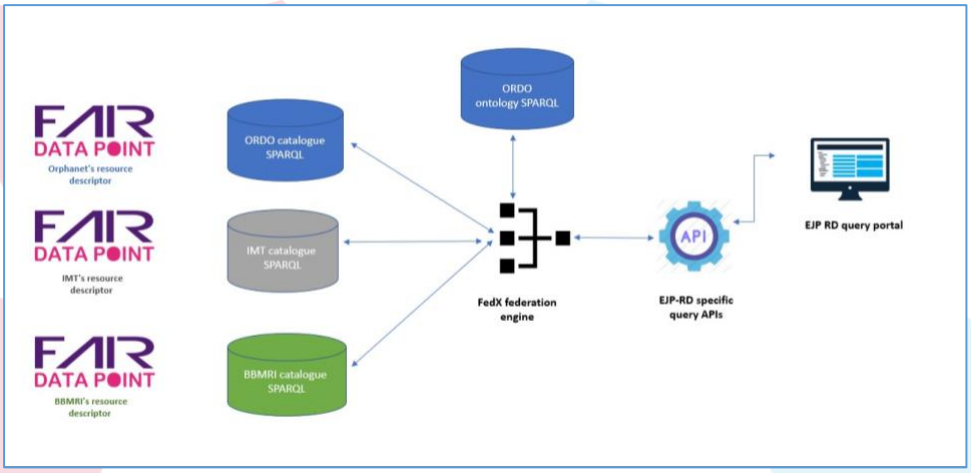

Figure 7. Federation design using FedX and EJP RD specific API

6. Implementation of the model into a Query Builder Proof of Concept

During Year 2, based on the evolution of the Metadata Model at resource level, a prototype "Proof of Concept" (PoC) was also set up in the context of the Query Builder Work Focus. This PoC, involving BBMRI, ERDRI and Orphanet, illustrates the capacity to

implement the model using a different technical setup. Each resource involved developed an API endpoint using the model and standardized specification. The common API relies on JSON exchange to perform queries.

The Query Builder API has been described into the related GitHub repository :

- https://github.com/ejp-rd-vp/query_builder_api

The API specification is based on a standard OpenAPI V3.03 (http://spec.openapis.org/oas/v3.0.3) and was designed to be oriented towards the EJP RD resource metadata schema (focus on catalogue, location, organization and resource items)

6.1. Orphanet process

Using the same process which exposes Orphanet data into the LDP, Orphanet has developed several webservices, compliant with the EJP RD query builder API specification and metadata model. The webservices, implemented to the Query Builder PoC allows to retrieve information about Biobanks and Registries that is accessible from Orphanet's catalogue using ORPHAcodes denoting specific rare diseases as input.

During this development phase a dedicated web server was set up(http://155.133.131.171:8080/ Apache Tomcat 9.0.36), which is used to run:

- A blazegraph instance (SPARQL Endpoint) with ORDO and Orphanet datasets.
- The query builder API Implementation:
  - The API is queryable accordingly to the specifications by using ORPHAcode.
  - The API's expected return is a JSON response which contains the corresponding dataset (could be empty) and using the EJP RD Metadata Model (see figure 8 for an example, http://155.133.131.171:8080/Orphanet/resource/search?orphaCode=558)

```
resourceResponses:
  0:
    id:            "101485"
    type:          "BiobankDataset"
    description:   "Biobank of Marfan biobank"
    name:          "Marfan biobank"
    homepage:      "http://www.orpha.net/consor/cgi-bin/OC_Exp.php?Lng=en&Expert=362573"
  1:
    id:            "73803"
    type:          "BiobankDataset"
    description:   "Biobank of MD-NET: Muscle Tissue Culture Collection (MTCC) (EuroBioBank partner)"
    name:          "MD-NET: Muscle Tissue Culture Collection (MTCC) (EuroBioBank partner)"
    homepage:      "http://www.orpha.net/consor/cgi-bin/OC_Exp.php?Lng=en&Expert=229349"
  2:
    id:            "67063"
    type:          "PatientRegistryDataset"
    description:   "PatientRegistry of RaDiCo-MARFAN: National cohort on Marfan syndrome and apparent diseases"
    name:          "RaDiCo-MARFAN: National cohort on Marfan syndrome and apparent diseases"
    homepage:      "http://www.orpha.net/consor/cgi-bin/OC_Exp.php?Lng=en&Expert=205886"
apiVersion:        "v0.2"
page:
  number:          0
  size:            10
  totalPages:      0
  totalElements:   3
```


Figure 8. JSON returned by the Orphanet query builder API with query parameters ORPHAcode=558

6.1.1. ORPHAcodes expansion webservices

Most catalogue resources use "disease concept" or "group of diseases concept" to annotate their content. Mostly a specific registry or biobank will be annotated with a limited amount of rare disease concepts. (For instance, a registry or a biobank could be annotated with "Fabry Disease" - disease concept - or "Lysosomal disease" - group of disease concept -). The same registry or biobank could be also be part of different catalogues, using different disease concepts annotation. Therefore, when performing a query across several catalogues, using a single ORPHAcode, the user will obtain only results that are annotated with this same specific ORPHAcode.

In order to improve the number of results when performing a query, especially by using "group of diseases concept" instead of a single or a specified list of "disease concept", there is a clear need to exploit the classification and hierarchy within a terminology.

ORDO (Orphanet Rare Diseases Ontology) organizes rare disease concepts (or entity) by using classifications and hierarchy (see figure 9).

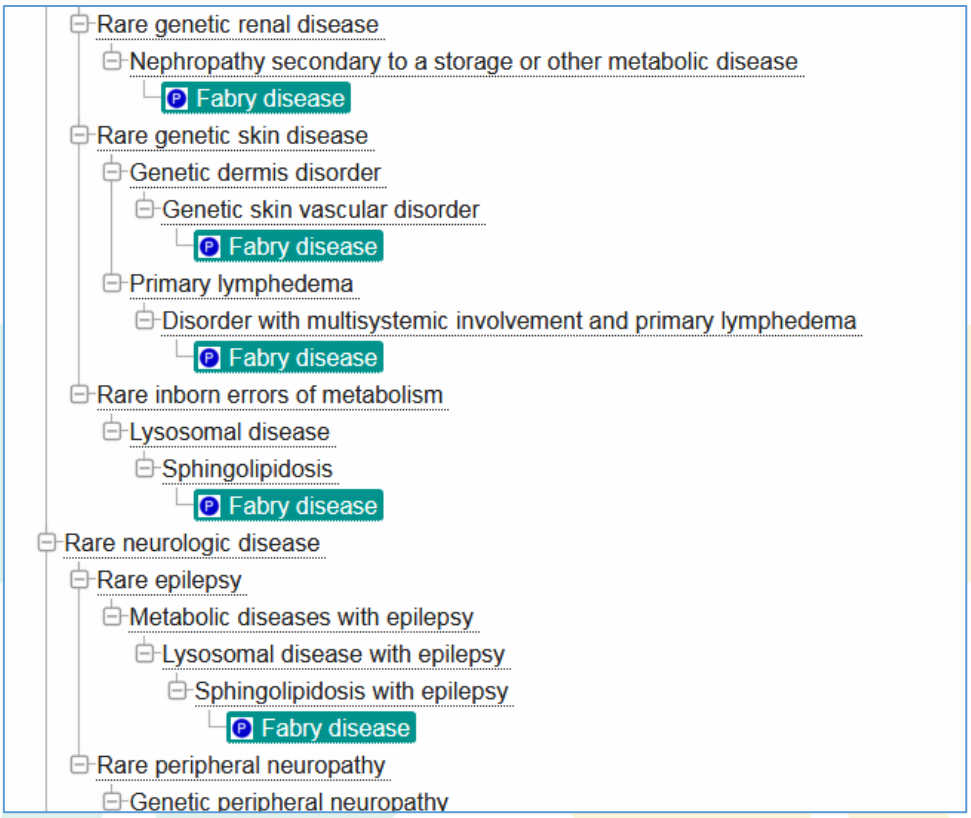

Figure 9. Partial view of multi-classified concept "Fabry disease" (ORPHA:324) in ORDO, using Ontology Look up service EBI (http://www.orpha.net/ORDO/Orphanet_324)

By using ORDO directly in queries submitted to a SPARQL Endpoint that is set to use inferences, the hierarchy of the ontology could be used to automatically expand a SPARQL query. This approach can be resource consuming from a computational point of view for certain complex queries and is not easily directly applied to a JSON API strategy. A common solution is to identify classes of queries that are commonly used and prepare an endpoint for these (internally the endpoint is still accessible for development purposes).

Therefore, Orphanet developed dedicated webservices which allow to exploit the hierarchy organisation from any node composed by an ORPHAcode. An ORPHAcode, which is unique, could be related to a disease concept or a group of disorders or a subtype.

A first webservice allow to obtain from this node all the N+1 and N-1 relations to upper or lower concept:
http://155.133.131.171:8080/Mapper/Inferred/**{ORPHAcode}**

For instance, for Marfan syndrome ORPHAcode:558 the result will return subtype (Marfan Type 1 and Type 2) and related N+1 to Marfan Syndrome based on whole ORDO hierarchies (see figure 10)

- http://155.133.131.171:8080/Mapper/Inferred/558/

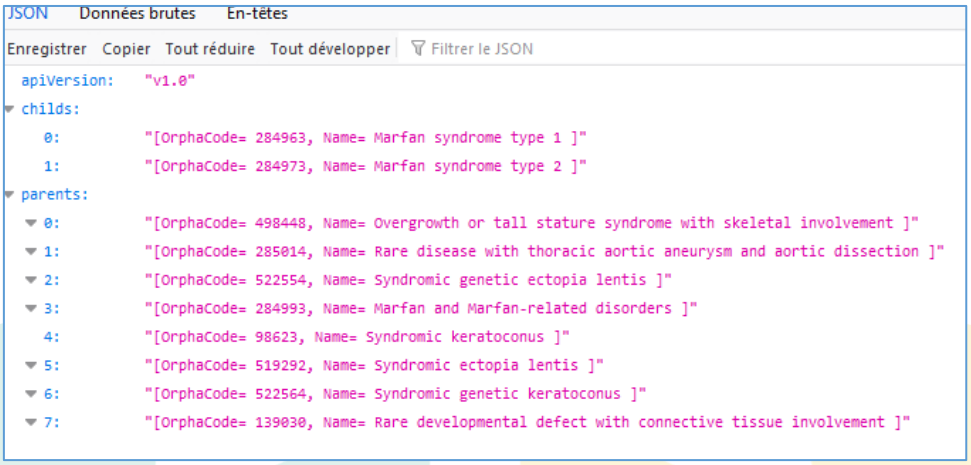

Figure 10. JSON returned by the webservice, using Marfan syndrome ORPHAcode:558

In order to allow customized queries, a webservice which permits to obtain only lower or upper levels from a dedicated node was also developed. The upper levels are reachable with an "ascendant" webservice http://155.133.131.171:8080/Mapper/Ascendant/**{ORPHAcode}** (see figure 11).

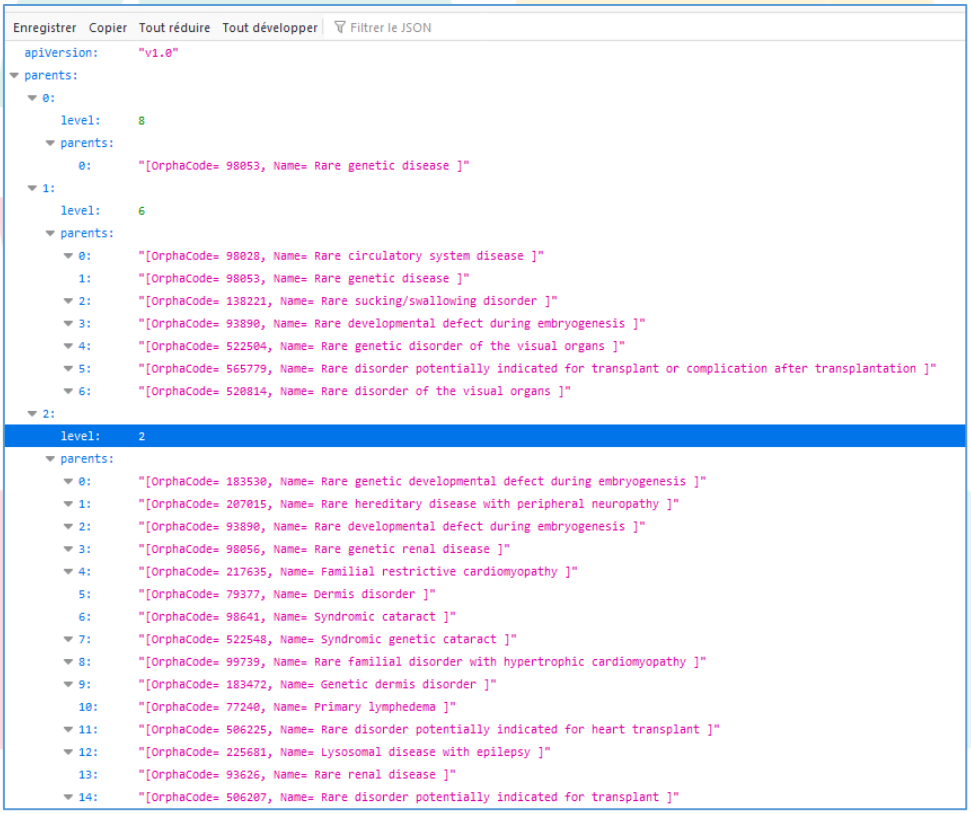

Figure 11. Partial view of the JSON returned by the webservice, using Fabry Disease ORPHAcode:324 http://155.133.131.171:8080/Mapper/Ascendant/324

A rank parameter could be also used in order to limit the hierarchies' path to a specific rank level:
http://155.133.131.171:8080/Mapper/Ascendant/**{ORPHAcode}/{rank integer}**

- http://155.133.131.171:8080/Mapper/Ascendant/324/2 will stop after N+2
- http://155.133.131.171:8080/Mapper/Ascendant/324/3 will stop after N+3.

The lower levels could be reach by using a "descendant" webservice:
http://155.133.131.171:8080/Mapper/Descendant/**{ORPHAcode}**

- http://155.133.131.171:8080/Mapper/Descendant/558/

As well, a rank limit could be used:
http://155.133.131.171:8080/Mapper/Descendant/**{ORPHAcode}/{rank integer}**

- http://155.133.131.171:8080/Mapper/Descendant/68362/2

By using the webservices, from any ORPHAcode as query entry, this let an application (API or software) to retrieve several other ORPHAcodes within the hierarchy and then use those codes in the context of an API. This way, any resources annotated with a relevant code could be present in the results. For instance, if a resource such as a registry is annotated with the ORPHAcode for "Fabry disease" and a second resource is annotated with the ORPHAcode for "lysosomal disease", using the webservices mechanism both results should be returned to the user.

6.1.2. ORPHAcodes mapping webservices

Orphanet provides in ORDO (Orphanet Rare Diseases Ontology) several mappings to other terminologies, including OMIM, ICD-10, MeSH, MedDRA, UMLS and SNOMED-CT. The mappings are expertised, using semi-automated mapping tools and a validation by the Orphanet curation team. Each mapping is assessed and indicated if it is an Exact match or NTBT/BTNT (Narrower term, Broader term) from ORPHAcodes concept to the considered terminologies.

When using directly the ontology, the mappings are directly available (with an exception with SNOMED CT mappings, due to specific licensing conditions).

Nevertheless, the integration of an ontology needs specific competences and technical setup. Therefore, in order to ease the integration of mappings in any application and API, Orphanet provides also dedicated mapping webservices which could be queried more easily.

A first webservice returns all available mappings for a specific ORPHAcodes: (for instance, ORPHAcodes 558 Marfan Syndrome)

- http://155.133.131.171:8080/OrphanetMapper/WSREST/MappingFromOrphanet/558

Then webservices are available to obtain mappings from a terminology to the ORPHAcodes:

1) Mapping from ICD10 to ORPHAcodes:

http://155.133.131.171:8080/OrphanetMapper/WSREST/MappingICD10ToOrphanet/**{ICD id}**

For instance:

- http://155.133.131.171:8080/OrphanetMapper/WSREST/MappingICD10ToOrphanet/Q87.4 (see figure 12)

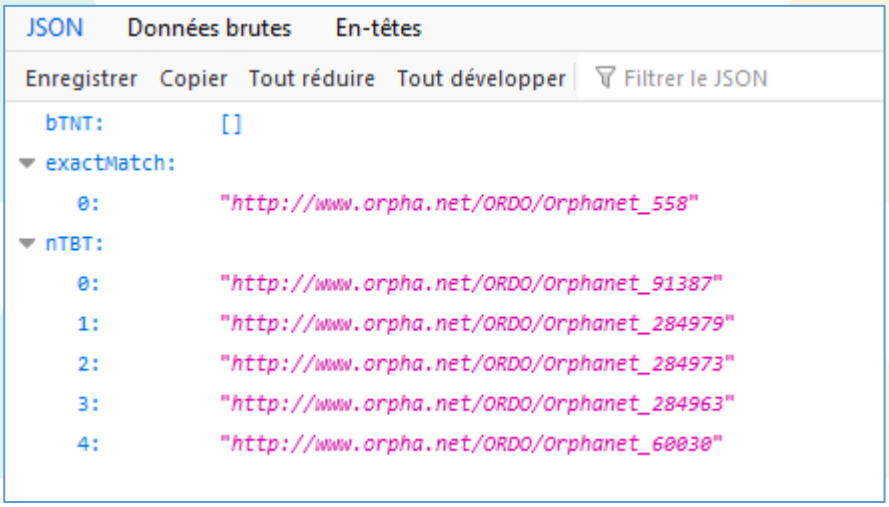

Figure 12. ORPHAcodes mapped to Q87.4 ICD code, JSON returned by using ICD code and mapping precision (Exact Match or BTNT (broader term to narrower term) or NTBT (narrower term to broader term)

The same methodology is applied to other available terminologies.

2) Mapping from MedDRA to ORPHAcodes:
http://155.133.131:8080/OrphanetMapper/WSREST/MappingMedDRAToOrphanet/**{Medranum}**

3) Mapping from UMLS to ORPHAcodes:

http://155.133.131:8080/OrphanetMapper/WSREST/MappingUMLSToOrphanet/**{UMLSNum}**

4) Mapping from MeSH to ORPHAcodes :

http://155.133.131:8080/OrphanetMapper/WSREST/MappingMeSHToOrphanet/**{MeshNum}**

5) Mapping from OMIM to ORPHAcodes:

http://155.133.131:8080/OrphanetMapper/WSREST/MappingOmimToOrphanet/**{OMIMNum}**

7. Discussion

The Virtual platform of RD resources annotated with the EJP RD ontological model has been updated by (i) implementing the previously developed metadata model for annotating RD resources using the DCAT 2.0 vocabulary and extensions thereof, and the Linked Data platform (ii) developing a pipeline that demonstrates how resources can expose their resource annotations in terms of this resource metadata model and query that with the generic query language SPARQL, (iii) developing an architecture to demonstrate how the model can be used by the JSON-based EJP RD query builder API. The metadata model and metadata management services are predominantly based on semantic web technologies to provide a generic and scalable backbone for ontological annotation of resources and usage thereof. An advantage is that these relatively simple services communicate directly in terms of ontologies: specificity is a result of the self-describing data and metadata that they communicate. On top of these semantic services, resource providers and system designers can provide other interfaces to integrate with other commonly used technologies, such as JSON-based common APIs. An example of this process was shown in the section 5 with the EJP RD query builder API implementation. Implementing yet another interface on top of semantic services such as the FAIR data point and SPARQL endpoints might be an overhead for systems designers. In the coming year, an exploration of how existing software components like "garlic" API server can make this process easier will be performed. There is still a gap between the semantic metadata model and the JSON schema that is used in sections 2 and 5. Efforts to close this gap will be made in the coming year together with the EJP RD Query Builder Work Focus team. The implementation of the metadata model will continue through the collaboration with this Work Focus.

By design and without strong constraints to any technical choices at implementation level (FAIR data point, SPARQL Endpoint or JSON based API), the metadata model and associated services could be used for annotation of catalogues at resource level. Furthermore, the "continuum" between annotation layers ranging from catalogues to data records will be part of the continued crosstalk between the Metadata Work Focus, its common data element and mapping subgroups, the EJP RD query builder work focus, and the FAIRification stewards. The Overall Architecture and FAIRification Work Foci will monitor the crosstalk and provide oversight.