Development of tools for Data Management

António Terra
Instituto Superior Técnico
antonio.terrao@tecnico.ulisboa.pt

ABSTRACT

With the large volumes of data currently being produced, the scientific community is posed with the challenge of how to manage large quantities of data. Data management aids researchers with the tasks of analysing, searching, and storing data. In order to push for better data management, and compliance with the open science principles, most funding agencies and research institutions are now requiring that grant applications be accompanied by a Data Management Plan (DMP). A DMP is a document that details how data is to be created, shared, published and preserved.

However, not all researchers have the necessary expertise or time to create a DMP and subsequently apply its recommendations. Additionally the existence of multiple DMP templates leads to a generalized lack of standardization in existing DMP documents.

The objective of this project is to tackle the challenge posed by the multitude of DMP templates created by funding agencies and research institutes. To achieve this objective the proposed approach is to collect DMP documents with the assistance of the Data Management service of ELIXIR Portugal and store a machine-actionable representation in a triplestore repository. Simultaneously the knowledge that is expressed in the stored DMP documents will be verified that covers what is required in the DMP template.

Author Keywords

Research Data Management; Data Management Plan; Elixir; BioData.pt;

CCS Concepts

•**Software and its engineering** → Collaboration in software development;

INTRODUCTION

The technical advances in computing power and sensory devices leads to extremely large volumes of data generated by scientific experiments and simulations. It is common to have a single simulation on a supercomputer generate terabytes of data, and for experiments to collect multiple petabytes of data. This volume, complexity and diversity of data cause scientists great hardship and waste of productive time to explore their scientific goals [13].

With this in mind it is necessary to create ways to manage this data. The scientific community develop the Data Management [16] that ensures that the story of a researcher's data collection process is organized, understandable, and transparent. One of the most important concepts in Data Management is that of lifecycle [16], that is the sequence of stages that a particular unit of data has to go through from its creation until its demise. Data management tries to make data publicly accessible to endorse Open science. There are four foundational principles for good Data Management: Findability, Accessibility, Interoperability and Reusability (FAIR) [18].

One way funding agencies and research institutions found to deal with the challenges of Data Management was to introduce the concept of Data Management Plan (DMP). A DMP is a document that accompanies a project. A DMP [1] states what data will be created and how, and outlines the plans for sharing and preservation, noting what is appropriate given the nature of the data and any restrictions that may need to be applied.

Funding agencies and research institutions have therefore started to demand that any funding applications or projects must be accompanied by a DMP. This poses several challenges for researchers. Researchers do not have the necessary expertise or time to create a DMP and subsequently apply its recommendations [6]. Additionally there is a multitude of existing DMP templates that adds to the lack of standardization in DMP creation. In their attempt to guide researchers in the DMP creation process most funding agencies and research institutions have created specific DMP templates for their research fields with varying levels of detail.

An evolution of the concept of DMP is the machine-actionable Data Management Plan (maDMP) [7] attempts to tackle the DMP creation issue. With the maDMP, the original DMP is fitted with actionable features, and should have both a human and machine-readable representation. This allows systems and services to take advantage of the knowledge expressed in the DMP, and aid researchers by automating parts of the DMP creation process.

However in order for systems and services to fully take advantage of the knowledge expressed in DMP, it was necessary to standardize the information. The DMP Common Standard data model [6] comprises of a universal core set of elements that define a DMP. The DMP Common Standard data model has several implementations, of those is the DMP Common Standard Ontology (DCSO)[2], which resorts to semantic technology to provide a representation of the DMP.

[1] DCC DMP: http://www.dcc.ac.uk/resources/data-management-plans [Retrieved 03/01/2020]

[2] DMP Common Standard Ontology Repository: https://github.com/RDA-DMP-Common/RDA-DMP-Common-Standard/tree/master/ontologies

Ontologies [9] define a common vocabulary for researchers who need to share information in a given domain. Ontologies can be represented through a multitude of Ontology representation languages [11]. These describe the ontology in a machine-readable way.

The initial objective of this theses was to explore the possibility that the knowledge expressed in a standardized DMP, can be reused to create a DMP that complies with a given DMP template. But during development that objective was seen as not viable because of all the maintenance that will be required in the future when a DMP template is updated or a new one is created. So the new objective is to represent the information from different DMP templates in a standard format, complient with the DCSO.

STATE OF THE ART

With a growing volume of research data being generated [13], there is also a growing need on how to manage, store, search and analyse research data. The open science movement, that promote the share and re-use of data, and the creation of the FAIR principles were the catalysts to the creation of DMP.

The funding bodies to comply with the open sciece and FAIR principles are asking grant applicants to provide data plans and each one have its own requirements for data plans. For example: The Arts and Humanities Research Council (AHRC)[3] requires a Technical Plan that summarize the Digital Outputs and Digital Technologies; The Biotechnology and Biological Sciences Research Council (BBSRC)[4] requires a statement on data sharing that include concise plans for data management and sharing as part of research grant proposal or provide explicit reasons why data sharing is not possible or appropriate; And the European Commission Horizon 2020 that offers a template for a FAIR that is findable, accessible, interoperable and re-usable, data management.

Research Data Management

The volume of data produced from scientific research is growing increasingly fast. Today it is common to have a single simulation generate terabytes of data and for experiments to collect multiple petabytes of data. These large quantities of research data may be valuable, but pose a complex challenge when in regards to its management, storage, search and analysis [13].

However, dealing with extremely large quantities of data is not the only challenge. Scientific research data presents three challenges [13]: (1) Multi-scale data, which refers to data generated at different scales. For example biological processes can be modeled at a DNA sequence level, molecular level or as protein complexes; (2) Diversity of data, which manifests itself in scientific projects that involve multiple diverse domain sciences; and (3) Depending on the application domains, different data models and data formats are used.

Research data management [17] concerns the organisation of data, from its entry to the research cycle through to the dissemination and archiving of valuable results. It aims to ensure reliable verification of results, and permits new and innovative research built on existing information. This is obtained by having descriptive names for variables, descriptive names for files and folders, unique identifiers for study participants and saved study workflows that describe the analysis methodology [16]. An example of a strategy is the usage of Data Management Plans explained in Section 2.2.

Data Lifecycle

The concept of data lifecycle [16], in data management, is often used to help researchers understand the scope and meaning of data management. The data lifecycle is the sequence of stages that a particular unit of data goes through from its initial generation to its eventual archival or deletion.

It can be divided in five stages: (1) the creation/collection, is the first stage of the lifecycle where the data is gathered or created; (2) the processing, after getting the data from the first stage is necessary to make sure it is clean and ready to be used; (3) the analysis, the data is used and analysed as needed; (4) the preservation, in this phase, data is stored without further processing or deleted if the data is no longer useful in any way; and (5) giving access to data, in this phase the data should be publicly available so it can be re-used.

Open Science and FAIR principles

Open science [10] is the efforts by researchers, governments, research funding bodies to the scientific community to make the primary outputs of publicly funded research results publicly accessible as a mean to accelerate research, enhancing transparency and collaboration, and fostering innovation.

With the objective of good data management are four foundational principles [18] [2] ,(1) Findability, datasets should be described, identified and registered or indexed in a clear and unequivocal manner; (2) Accessibility, datasets should be accessible through a clearly defined access procedure, ideally using automated means. Metadata should always remain accessible; (3) Interoperability, data and metadata are conceptualised, expressed and structured using common, published standards and (4) Reusability, characteristics of data and their provenance are described in detail according to domain- relevant community standards, with clear and accessible conditions for use (FAIR).

These principles not only apply to data but to all digital research objects such as algorithms, tools and workflows, since all components of the research process must be available to ensure transparency, reproducibility and reusability.

Data Management Plan

In order to push for better data management, and compliance with the open science and FAIR principles, most funding bodies and research institutions are now requiring that grant applications be accompanied by a DMP [7].

A DMP is a document that accompanies a project. A DMP states what data will be created and how, and outlines the plans for sharing and preservation, noting what is appropriate given the nature of the data and any restrictions that may need to be applied.

[3] AHRC: https://ahrc.ukri.org/

[4] BBSRC: https://bbsrc.ukri.org/

DMP Creation

DMP creation can be performed in many ways. Many funding bodies and research institutions have created their own DMP templates [5] [6] that researchers are encouraged to comply with when applying for funding.

There are ten simple rules [5] to create a DMP that is easily understood by others and put to use by your research team: (1) Determine the Research Sponsor Requirements; (2) Identify the Data to Be Collected; (3) Define How the Data Will Be Organized; (4) Explain How the Data Will Be Documented; (5) Describe How Data Quality Will Be Assured; (6) Present a Sound Data Storage and Preservation Strategy; (7) Define the Project's Data Policies; (8) Describe How the Data Will Be Disseminated; (9) Assign Roles and Responsibilities; (10) Prepare a Realistic Budget.

The DMP Common Standard data model [6] is a product of a Research Data Alliance (RDA) working group, the DMP Common Standards Working Group[7]. Its objective was to create a data model defining a universal core set of elements to define a DMP, because of the lack of a standard for what information a DMP should contain. The data model allows for customisation and extension using existing standards and vocabularies to follow best practices developed in various research communities.

The DMP Common Standard Working group set out to provide reference to implementations of the data model using popular representation languages (e.g., JSON, XML, RDF). This was meant to allow for tools and systems involved in processing research data to read and write information to and from a DMP, and thus comply with the maDMP concept.

Machine-actionable Data Management Plan

In many cases the DMP is created and updated not when the data is actually produced, but when it is required for reporting. This leads the DMP to be created after the project end, where important data may no longer be available, and many of the answers can be generic [7]. The general perception is that a DMP is a annoying administrative exercise and does not support data management activities [8]. Questions can remain unanswered, or the answers can be overly generic due to the use of free-form text.

The concept machine-actionable data management plan (maDMP) was created to enable the automation of DMP creation, what helped with the problems detailed previously. The maDMP [7] is concept where a standard DMP is fitted with dynamic features. For example, a DMP might sees some of its sections being completed automatically with information obtained from other tools. This can potentially result in a DMP created and updated during the execution of a project, where more information is available, which leads to researchers having to handle less bureaucratic tasks and reduce the overall workload. With the maDMP there is also the potential for both funders and repositories to validate the compliance of a DMP with any potential research data management guidelines automatically.

The DMP is meant to be created and consumed by multiple stakeholders, so it is necessary to involve all stakeholders throughout the research data management lifecycle. The maDMP will facilitate the structuring of information, but this has to be complemented by the expertise of the various stakeholders.

With the goal of improving the experience for all involved by exchanging information across research tools and systems and embedding data management plans in existing workflows, the following ten rules for machine-actionable data management plans were proposed [8]: (1) Integrate DMPs with the workflows of all stakeholders in the research data ecosystem; (2) Allow automated systems to act on behalf of stakeholders; (3) Make policies (also) for machines, not just for people; (4) Describe—for both machines and humans—the components of the data management ecosystem; (5) Use PIDs and controlled vocabularies; (6) Follow a common data model for machine-actionable DMPs; (7) Make DMPs available for human and machine consumption; (8) Support data management evaluation and monitoring; (9) Make DMPs updatable, living, versioned documents; (10) Make DMPs publicly available.

Semantic Technology

Semantic technology uses formal semantics to help computers understand language and process information the way humans do. They are able to store, manage and retrieve information based on meaning and logical relationships.

Ontology is a part of the semantic technology, that can describe concepts, relationships between things, and categories of things in a formal and structured form.

One of the definitions of ontology, from Rudi Studer [14], is that an "ontology is an explicit representation of a conceptualization. This conceptualization includes a set of concepts, their definition and their inter-relationships. Preferably this conceptualization is shared or agreed." Ontologies [9] are a way to represent information, defining categories, relations between concepts and entities. Some of the reasons to create a ontology are to share common understanding of the structure of information, to enable reuse of domain knowledge, to make domain assumptions explicit, to separate domain knowledge from the operational knowledge and to analyze domain knowledge.

Often an ontology is not the goal itself, but a definition of a set of data to be used in other applications. Basically ontologies are created to limit the complexity of information and to facilitate the resolution of problems within a given domain. To develop an ontology it is necessary to define the classes, arrange the classes in a taxonomic hierarchy, define the slots and describe allowed values for these slots and fill the values for the slots instances.

[5] European Commission (EC) H2020 DMP Template: https://ec.europa.eu/research/participants/data/ref/h2020/call_ptef/pt/2018-2020/h2020-call-pt-ria-ia-2018-20_en.pdf [Retrieved at 23/12/2020]

[6] Science Europe DMP Template: https://www.scienceeurope.org/media/jezkhnoo/se_rdm_practical_guide_final.pdf [Retrieved at 23/12/2020]

[7] DMP Common Standards WG: https://www.rd-alliance.org/groups/dmp-common-standards-wg [Retrived 04/01/2020]

The use of ontologies can be found in many research fields, such as: knowledge management, intelligent information integration, e-commerce, cooperative information system, database integration. The reason behind the popularity of ontologies, lies in their promise of a shared and common understanding of some domain, which can be communicated across people and computers [15].

Ontology Representation Languages

A Ontology representation language [11] "must describe meaning in a machine-readable way. Therefore, an ontology language needs not only to include the ability to specify vocabulary, but also the means to formally define it in such a way that it will work for automated reasoning". In [15] it is proposed this classification of ontology representation languages: (1) traditional ontology languages; (2) Web standards; and (3) Web-based ontology specification languages.

OWL has the biggest community of users, however it lacks of supporting tools, with only two tools, Protégé and OWL Validator.

OWL [4] is compatible with early ontology languages, including SHOE, DAML+OIL, and provides the engineer more power to express semantics. It includes conjunction, disjunction, existentially, and universally quantified variables, which can be used to carry out logical inferences and derive knowledge.

Although having some drawbacks like: (1) some constructs are very complex; and (2) reasoning is not efficient as there is a trade-off against time-complex cost. It is the Ontology Language chosen to be used to represent the knowledge graphs during the development of this theses.

Terse RDF Triple Language (Turtle) [1] is a syntax and file format for expressing data in RDF data model. Turtle provides levels of compatibility with the N-Triples format as well as the triple pattern syntax of the SPARQL W3C Recommendation.

SPARQL

The knowledge graphs created in the context of the solution (see section 4) and stored need to be analyzed, to do that it is necessary to query them, SPARQL is the W3C candidate recommendation query language for RDF. SPARQL[8] is a graph-matching query language for RDF. It can navigate in RDF graphs data through graph pattern matching. In this process, simple patterns can be combined into more complex ones, which explore more elaborate relationships in the data.

A SPARQL query [12] consists of three parts: (1) The pattern matching , that includes optional parts such as union of patterns, nesting, filtering values of possible matchings, and choose the data source to be matched by a pattern. (2) The solution modifiers, allows to modify the output values applying operators such as projection, distinct, order, limit, and offset. (3) The output can be of different types, yes/no queries, selections of values of the variables, construction of new triples from these values, and descriptions of resources.

[8]SPARQL: https://www.w3.org/TR/rdf-sparql-query/

SPARQL [3] has been a core focus of research and development for Semantic Web technologies, with various research proposals, benchmarks, open-source and commercial tools emerging to address the challenges of processing SPARQL queries efficiently, at large scale and in distributed environments. These advances in SPARQL technology and tools have been paralleled by the deployment of public SPARQL endpoints on the Web, there are over four hundred endpoints announced as found in [9].

PROBLEM CONTEXT

The volume of research data generated is growing increasingly fast, which in turn makes the task of managing and curating said data, all the much harder. Research Data Management (RDM) (see section 2.1) is a research topic focused on addressing some of the challenges associated with managing large volumes of research data.

The concept of Data Management Plan (DMP) falls under the realm of RDM, as it was idealized as a tool to help manage data throughout the lifecycle of a research project, whilst assisting in the adherence to the concepts of open science and the FAIR Data principles. A DMP states what data will be created and how, and outlines the plans for sharing and preservation, noting what is appropriate given the nature of the data and any restrictions that may need to be applied as explained in more detail in Section 2.2.

Funding agencies to adopt the concepts of Open Science and FAIR Data Principles started requiring a DMP with each submission of application for funding. With the objective of standardize the DMP multiple templates were created. But the question-answers in the DMP templates are in free-text format what lead to sometimes being answered generically.

This thesis focused on a generic problem of information representation. As to tackle the challenge posed by the multitude of DMP templates created by funding bodies and research institutes, the work developed in the context of this thesis, explored the possibility that the knowledge expressed in a standardized DMP, could be reused to automatically generate multiple DMP representations what would comply with individual DMP templates.

In the beginning of the project were imposed the usage of DSW as the tool to create DMP and that the machine-actionable DMP representation is a serialization of DCS application profile, the DCSO that is a domain ontology, knowledge graphs will be created from that ontology.

During the development of the thesis the main objective has changed because the first one has not viable because of the high maintenance necessary to keep up to date, any time a template is updated or created a new one, the application would need to be updated. So a new approach was chosen, instead of from a standardized DMP being created multiple representations for each DMP template, the new approach the user fills a questionnaire based on one DMP template of his

[9]SPARQL Endpoints Status: https://labs.mondeca.com/sparqlEndpointsStatus/inde
[Retrieved in 03/01/2020]

choice, then that questionnaire is used to create a standardized DMP compliant with the DCSO.

To achieve this objective there are two main problems to resolve, (1) a maDMP collection and creation workflow is necessary to be possible to access and use the information contained in DMP documents. Real DMP documents will be collected using DSW in the context of the BioData.pt programme "Ready For BioData Management?". (2) A service to visualize knowledge graphs, to facilitate the readability of maDMP to humans. The proposed solution to each of this problems is detailed in chapter 4.

PROPOSED SOLUTION

This section details the proposed solution to the problems raised in section 3. It is divided in three sections, the first it will explain what are the processes that constitute the workflow to create maDMP documents, the second section explain the solution to visualise maDMP documents, and the third section explains what is ELIXIR and BioData.pt (ELIXIR Portugal), because this theses is been developed with BioData.pt therefore are some constrains imposed such as the usage of DSW as a DMP creation tool.

Workflow to create a maDMP

To create a visualization service is necessary to have DMP documents represented as maDMP documents, for that a workflow to create DMP documents and convert them in maDMP is going to be developed. The workflow was divided in three main processes as follows:

(1) The first process is to create a DMP. DSW has imposed as the tool to create DMP documents because it is used by BioData.pt, and real DMP documents will be collected from the advanced workshop from the programme "Ready For BioData Management?".

DSW have a simple and intuitive interface, and make the questionnaires based on the information needed to fill a DCSO that is a serialization of the DCS.

(2) Having the DMP created it was necessary to convert it in a maDMP, because DSW only exported the DMP in JSON or PDF. So a application to convert a JSON file to a OWL file representative of the DCSO would have to be created.

Afterwards during a RDA Hackathon the functionality to export the DMP as an knowledge graph was implemented, were the application developed has used as a base.

(3) Having the maDMP created is necessary to preserve it. To store the knowledge graphs is required a solution that stores them permanently and that permits SPARQL queering to access the knowledge express in the maDMP, for this a triplestore is going to be used.

But is necessary to import the knowledge graphs to the triplestore, other than manually because it is time consuming and error-prone. So a service that automatically gather the DMP documents from DSW (already converted in as a knowledge graph) and import them to the triplestore would have to be developed.

The maDMP collection and creation workflow in the end should be accordant with the following description. Where it starts with the user requesting a form to create a DMP in the DSW instance, them fills it and submit. Meanwhile in the remote machine a application that verifies if new Documents were created in DSW and export them to the triplestore. Them the user can access the triplestore and verify that the DMP created is accessible as a knowledge graph.

Service to Visualize DMP

The DMP as a knowledge graph is hard to read for a human, so a service to display that information is necessary.

The initial solution was to use an already existing application that could display the information of a knowledge graph when given a SPARQL Endpoint, but multiple problems occurred, so a application that can achieve this objective would have to be developed.

IMPLEMENTATION

Create the DMP

As proposed in section 4.1 DMP documents will be created using DSW where the user has to answer a questionnaire that contains the information needed to fill the DCSO, but the output of DSW is a JSON file what is not practical to analyze with queries. So a service to convert that JSON in a OWL file was created, as explained in Section 5.1.1.

Shortly after the development of the converting service a RDA hackathon was organized as explained in Section 5.1.2, where a team solve the same problem I was trying to solve, by implementing a export as knowledge graph option in DSW that works with a new version of the DCSO that was created during the same hackathon.

Convert Data Stewardship Wizard Document to a DMP Common Standard Ontology

General explanation

The DSWtoDCSO aims to create an instance of a DCSO based in the replies from a DSW questionnaire. To achieve that it was created a DSW UUID mapping, to know the path to each reply of the questionnaire.

This UUID mapping has created by hand (from the DSW website for each question was gathered the respective UUID). The mapping was created in a form of a tree.

The tree is traversed recursively in order to obtain all the values for the creation of all the individuals and his data and object properties.

In the beginning of the application is created a new Ontology that imports the DCSO, and then were created the individuals and his Data and Object Properties given the information retrieved from the DSW Document.

The DSWtoDCSO application used the OWL API to create and update the ontology.

DSW UUID Mapping

The UUID Mapping of the DSW Document is a JSON with two main JSONs:

- answerReply - JSON with UUID of fixed options replies as key and the reply string as value
- questions - Array of JSONs, each JSON have two keys: question, were the value is the name of field, and uuid, were the value is a JSON with the properties for the field as keys (this properties can be another field with his properties, making it like a Tree) and the value is the uuid of the DSW document, in addition to the properties of the field is an extra property (quantity) that contains the uuid for the number of instances of that field exists (if only can exist one this property is an empty string).

Gather Data from DSW Document

With the DSW UUID Mapping is created a HashMap with the content of the questions JSON referred above to be used to go through all the properties for each field. From the DSW Document is crerated two HashMaps, one with the value for each path, and other with the required level for each property.

Given a category name (in this case DMP, Contact, Contributor, Project, Cost or Dataset) the Tree of properties for that category is covered recursively in order to create a JSONObject for each field with the values of his properties, than is stored in a HashMap.

Creating Individuals and his properties

Before create the individuals all the data and object properties present in the Ontology are retrieved.

To create the individuals all the replies of the questionnaire will be check. For each category, in the HashMap that stored the reply values, check all the elements of the JSONObject and if is a list than it is a new individual. It is called the method to create a new individual, then the method to create the data properties.

The method to create the data properties check for all the properties of that JSONObject if they are a data property for the class of the individual and associate the data property to the individual.

Then it is called a method to check if there are more individuals inside this element (this function will run recursively) If there are the individuals are created with his respective data and object properties.

The method to create the object properties verifies if exists a object property between the individual created and his father from the tree.

In the end the application will check if are object properties for the DMP because in the mapping his object properties are not encapsulated in the DMP JSONObject, so is checked all individuals if are a object property of each other (I only do it for the DMP because is known that he is the only one for this case).

maDMP Hackathon

A RDA hackathon on machine-actionable Data Management Plans took place between 27th and 29th May 2020 and gathered 71 participants from 21 countries across the world. The main goal was to use the RDA Common Standard for maDMPs, one of the task forces part of the RDA DMP Common Standard Working Group (DCS-WG), in a variety of settings, but the specific topics were determined by the participants, the range of topics can be grouped into the following categories:

- Integration of Data Management Plan (DMP) tools, e.g., exchanging maDMPs between DMP tools
- Other integrations, e.g., exchange of maDMPs between other services, such as repositories, etc.
- Mapping of maDMPs to funder templates, e.g,. Science Europe, NSF, etc.
- New serialisations, e.g., OWL ontology

One of the groups in the Hackathon tackle the problem of exporting DMP documents from DSW to an DCSO, the same problem my application referred in section 5.1.1 was trying to solve.

To implement the functionality DSW uses Jinja2 templates for DMP exports from questionnaires that are based on a certain Knowledge Model (KM). To allow export of maDMP in JSON according to the DCSO they had to extend and adjust the common KM to be more compliant and create a Jinja2 template for assembling JSON files from questionnaires.

Preserve the DMP

After solve the problem how to create a DMP was necessary to store them, for that a triplestore was used, the Apache Jena Fuseki was chosen for that and installed in a remote machine.

But to export every document from DSW to the triplestore manually was very labor intensive given that every time a new document as created it was needed to be exported, what can lead to errors like missing information because a document was forgotten.

So to solve this a Java application was created to automatically export the documents from DSW to the triplestore, as explain in more detail in Section 5.2.1.

DSWExport

DSWExport is a Java application that aims to download the most recent Document from each Questionnaire from the DSW, then the Documents are uploaded to a Apache Fuseki Server, where they can be queried. The Fuseki Server is the default installation but with some visual changes as the change of the logo and title of the page.

The application starts by asking the bearer token to the DSW, after was received it the application enters a infinite loop, each iteration occurs 10 minutes after the last iteration ends, where it ask for all the questionnaires from the DSW, for now only accept turtle and RDF/XML files. Then begins a new loop, for each questionnaire, the application asks for the questionnaire name and documents UUIDs to the DSW using the questionnaire UUID, then choose the most recent document and verify if it was already uploaded to the Fuseki Server, if not then the document is downloaded and uploaded to the Fuseki Server. All the communication between the

application DSWExport and DSW or between DSWExport and Fuseki Server are through HTTP requests.

The architecture of DSWExport comprises three Packeges:

- DSWexport - It contains the main class and the class responsible for the HTTP connections using Unirest.
- DSWComm - It is responsible for the communication with the DSW.
- FusekiComm - It is responsible for the communication with the Fuseki Server.

Visualize the DMP

With the DMP stored in a triplestore it was necessary to show that the information can be access and utilized, so with that purpose in mind I tried to use a visualization application for knowledge graphs.

The first one to be tested was Pubby but I was having some troubles with it, as explained in Section 5.3.1. So some alternatives to Pubby were tested, the alternatives was LODDY [10] and MadsHolten/sparql-visualizer [11] but I could not get them working properly.

As I could not find any solution a application to visualize the contents of the knowledge graphs present in a Fuseki server was created as explained in Section 5.3.2.

Pubby interface

To serve the ontologies gathered in the Apache Fuseki Server as a Linked Data, it has created a Pubby interface. It has created a Web Server with Jetty and it has used the Pubby frontend.

But not everything goes as expected, Pubby gets the information of my ontology, but when I try to open a encapsulated value like hasContact it gives the error 404 not found, but the turtle output of the Pubby has that information.

Ontology Visualizer

With the purpose of showing the content of the Ontologies gathered in the Fuseki Server in a way that is easy readable by a human and given that I was not able to understand the problem with Pubby referred in section 5.3.1, a React Application was developed.

The React application is composed by two parts, the first one is responsible for fetching all the Datasets present in a Fuseki Server, this is achived by a HTTP request using the library Axios, and list them into a table.

The second part is after choosing a Dataset from Fuseki it makes a Query to the server, again using a HTTP request with Axios, the results are received as a Turtle document so for an easy access to the values a Library (frogcat/ttl2jsonld) has used to convert Turtle in a JSON Object. Then the results are showed in a form of a table that contains all the subjects and respective values of the given Ontology, because some of the values are an object with pairs of subjects/values it is created a table inside the value cell, to create the full table it has used a recursive function to traverse the JSON Object, where it checks if the value of a given subject is an object and if it is calls it self again, if it has a non-object value returns the new row to the table.

This React application is integrated in the Fuseki Server webpage via a button the redirects to the application.

[10] LOBBY: https://bitbucket.org/art-uniroma2/loddy/src/master/ [retrieved in 10/12/2020]

[11] MadsHolten: https://github.com/MadsHolten/sparql-visualizer [retrieved in 10/12/2020]

CONCLUSION

This theses begin with the plan of create a view for each DMP template from a standard knowledge graph, but another approach was taken because the first one was not practical given the amount of maintenance it will require when a template is updated or when a new template is created.

The new approach aims to create a standard knowledge graph, based on the DCSO, with the information from the multiple different templates. To achieve this, two major steps had to be completed, (1) a creation of a workflow to create a maDMP, the workflow goes from the creation of a DMP through the conversion to a machine-actionable DMP and it preservation in a triplestore; (2) a creation of service to visualize maDMP;

To create the DMP documents the BioData.pt DSW instance was used, both synthetic and real DMP documents were created.

To convert the DMP to a machine-actionable a partnership with the developers of DSW was made with the objective of using their interface and create a way to export the answers of the questionnaires as a knowledge graph, as explained in Section 5.1.

To preserve the DMP a Apache Fuseki Server was created to store all the knowledge graphs, and a application that exports the Documents from DSW and import them into the Fuseki Server was developed, as explained in Section 5.2.

To visualize maDMPs multiple tools were tested but was not able to configure them properly to work as needed, as explained in more detail in Section 5.3, so a new application was created to access the information on each knowledge graph stored in Fuseki Server.

System Limitations and Future Work

Given the time restrain the system have some limitations, such as the application DSWExport seen in Section 5.2.1 only exports Turtle and RDF files from DSW but the Fuseki Server accept a wider variety of file types. Another limitation is the knowledge graphs created from DSW contains only one entity with all the information instead of creating multiple entities of each class with a respective information. Other limitation is that the OntologyVisualyzer only displays all information of a given knowledge graph.

In the future the application DSWExport should export every file type suported by the Fuseki Server, and the OntologyVisualyzer could support more specific queries and reasoning. Now that base is created it is possible to create services that take advantage of the information contained in the knowledge graphs.

ACKNOWLEDGMENTS

I would like to thank my family for their friendship, encouragement and caring over all these years, for always being there for me through thick and thin and without whom this project would not be possible. I would also like to thank my grandparents, aunts, uncles and cousins for their understanding and support throughout all these years.

I would also like to acknowledge my dissertation supervisors Prof. José Borbinha and João Cardoso for their insight, support and sharing of knowledge that has made this Thesis possible.

Last but not least, to all my friends and colleagues that helped me grow as a person and were always there for me during the good and bad times in my life. Thank you.

To each and every one of you – Thank you.

REFERENCES

[1] David Beckett, Tim Berners-Lee, Eric Prud'hommeaux, and Gavin Carothers. 2014. RDF 1.1 Turtle. *World Wide Web Consortium* (2014).

[2] M. Boeckhout, G. Zielhuis, and A. Bredenoord. 2018. The FAIR guiding principles for data stewardship: fair enough? *European Journal of Human Genetics* 26 (2018), 931–936.

[3] Carlos Buil-Aranda, Aidan Hogan, Jürgen Umbrich, and Pierre-Yves Vandenbussche. 2013. SPARQL Web-Querying Infrastructure: Ready for Action?. In *The Semantic Web – ISWC 2013*, Harith Alani, Lalana Kagal, Achille Fokoue, Paul Groth, Chris Biemann, Josiane Xavier Parreira, Lora Aroyo, Natasha Noy, Chris Welty, and Krzysztof Janowicz (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 277–293.

[4] Diana Kalibatiene and Olegas Vasilecas. 2011. Survey on Ontology Languages. In *Perspectives in Business Informatics Research*, Janis Grabis and Marite Kirikova (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 124–141.

[5] William K. Michener. 2015. Ten Simple Rules for Creating a Good Data Management Plan. *PLoS Comput Biol* 11 (10/2015 2015). DOI: http://dx.doi.org/10.1371/journal.pcbi.1004525

[6] Tomasz Miksa, João Cardoso, and José Borbinha. 2018. Framing the scope of the common data model for machine-actionable Data Management Plans. In *2018 IEEE International Conference on Big Data (Big Data)*. IEEE, 2733–2742.

[7] Tomasz Miksa, Andreas Rauber, Roman Ganguly, and Paolo Budroni. 2017. Information Integration for Machine Actionable Data Management Plans. *International Journal of Digital Curation* 12 (09 2017), 22. DOI:http://dx.doi.org/10.2218/ijdc.v12i1.529

[8] Tomasz Miksa, Stephanie Simms, Daniel Mietchen, and Sarah Jones. 2018. Ten simple rules for machine-actionable data management plans (preprint). (Feb. 2018). DOI: http://dx.doi.org/10.5281/zenodo.1172673

[9] Natalya F Noy, Deborah L McGuinness, and others. 2001. Ontology development 101: A guide to creating your first ontology. (2001).

[10] OECD. 2015. Making Open Science a Reality. 25 (2015). DOI:http://dx.doi.org/https://doi.org/https://doi.org/10.1787/5jrs2f963zs1-en

[11] J R G Pulido, Miguel Garcia-Ruiz, R. Herrera, Maria Cabello, Steve Legrand, and David Elliman. 2006. Ontology languages for the semantic web: A never completely updated review. *Knowledge-Based Systems* 19 (11 2006), 489–497. DOI: http://dx.doi.org/10.1016/j.knosys.2006.04.013

[12] Jorge Pérez, Marcelo Arenas, and Claudio Gutierrez. 2006. Semantics and Complexity of SPARQL. *Lecture Notes in Computer Science The Semantic Web - ISWC 2006* (2006), 30–43. DOI: http://dx.doi.org/10.1007/11926078_3

[13] Arie Shoshani and Doron Rotem. 2009. Scientific data management. Challenges, technology, and development. *Scientific Data Management: Challenges, Technology, and Deployment* (01 2009). DOI: http://dx.doi.org/10.1201/9781420069815

[14] Rudi Studer, V.Richard Benjamins, and Dieter Fensel. 1998. Knowledge engineering: Principles and methods. *Data & Knowledge Engineering* 25, 1 (1998), 161 – 197. DOI:http://dx.doi.org/https://doi.org/10.1016/S0169-023X(97)00056-6

[15] Xiaomeng Su and Lars Ilebrekke. 2002. A comparative study of ontology languages and tools. In *International Conference on Advanced Information Systems Engineering*. Springer, 761–765.

[16] Alisa Surkis and Kevin Read. 2015. Research data management. *Journal of the Medical Library Association: JMLA* 103, 3 (2015), 154.

[17] A. Whyte and J. Tedds. 2011. Making the Case for Research Data Management.

[18] Mark D Wilkinson, Michel Dumontier, IJsbrand Jan Aalbersberg, Gabrielle Appleton, Myles Axton, Arie Baak, Niklas Blomberg, Jan-Willem Boiten, Luiz Bonino da Silva Santos, Philip E Bourne, and others. 2016. The FAIR Guiding Principles for scientific data management and stewardship. *Scientific data* 3 (2016).