



Development of tools for Data Management

António Melo da Terra

Thesis to obtain the Master of Science Degree in
Information Systems and Computer Engineering

Supervisors: Prof. Doutor José Luís Brinquete Borbinha
Eng. João Manuel Fernandes Cardoso

Examination Committee

Chairperson: Prof. Doutor Miguel Nuno Dias Alves Pupo Correia
Supervisor: Prof. Doutor José Luís Brinquete Borbinha
Member of the Committee: Doutor Daniel Pedro de Jesus Faria

November 2020

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.

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.

Keywords

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

Contents

1	Introduction	1
1.1	Introduction	3
1.2	Thesis Statement	4
1.3	Outline	4
2	State of The Art	5
2.1	Research Data Management	7
2.1.1	Data Lifecycle	8
2.1.2	Open Science and FAIR principles	8
2.2	Data Management Plan	9
2.2.1	DMP Creation	9
2.2.1.A	DMP Common Standard	10
2.2.2	Machine-actionable Data Management Plan	10
2.3	Semantic Technology	11
2.3.1	Ontology Representation Languages	11
2.3.2	SPARQL	13
2.3.3	Technical Resources	13
2.3.3.A	Ontology creation API	13
2.3.3.B	DMP creation tools	14
2.3.3.C	Triplestores	14
2.3.3.D	Knowledge graphs visualization	15
3	Problem Context	16
3.1	Problem context	17
4	Proposed Solution	18
4.1	Workflow to create a maDMP	19
4.2	Service to Visualize DMP	20
4.3	ELIXIR	20
4.3.1	ELIXIR Portugal	22

4.3.1.A	Data Resources	23
4.3.1.B	Training	23
4.3.1.C	Software tools	23
4.3.1.D	Computing	24
4.3.2	Data management in Elixir	24
5	Implementation	26
5.1	Create the DMP	27
5.1.1	Convert Data Stewardship Wizard Document to a DMP Common Standard Ontology	27
5.1.1.A	General explanation	27
5.1.1.B	DSW UUID Mapping	27
5.1.1.C	Gather Data from DSW Document	28
5.1.1.D	Creating Individuals and his properties	28
5.1.2	maDMP Hackathon	29
5.2	Preserve the DMP	29
5.2.1	DSWExport	30
5.3	Visualize the DMP	31
5.3.1	Pubby interface	31
5.3.2	Ontology Visualizer	32
5.4	Workflow Demonstration	33
5.4.1	Convert Data Stewardship Wizard Document to a DMP Common Standard Ontology	33
5.4.2	DSWExport	34
5.4.3	Ontology Visualizer	35
6	Conclusion	41
6.1	Conclusions	43
6.2	System Limitations and Future Work	43

List of Figures

4.1	Workflow BPMN	20
5.1	DSWtoDCSO diagram	28
5.2	DSWExport BPMN	30
5.3	Pubby interface	31
5.4	Pubby hasContact values	32
5.5	Pubby Turtle Output	32
5.6	DSW questionnaire for synthetic DMP	33
5.7	DSW list of DMPs	34
5.8	DSW Document to be used in the demonstration	35
5.9	Input changed to the Document Downloaded	36
5.10	Execution of DSWtoDCSO	36
5.11	Visualization of new Ontology File	37
5.12	Start Fuseki Server	37
5.13	Fuseki is empty	38
5.14	Execute DSWExport	38
5.15	Fuseki Server after DSWExport execution	39
5.16	Start Ontology Visualizer	39
5.17	Ontology Visualizer	40
5.18	DMP example in Ontology Visualizer	40

List of Tables

2.1	Ontology Languages Categories	12
-----	-----------------------------------------	----

Acronyms

DSW	Data Stewardship Wizard
DCSO	DMP Common Standard Ontology
DCS	DMP Common Standard
DMP	Data Management Plan
RDA	Research Data Alliance
KM	Knowledge Model

1

Introduction

Contents

1.1 Introduction	3
1.2 Thesis Statement	4
1.3 Outline	4

1.1 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 [1].

With this in mind it is necessary to create ways to manage this data. The scientific community develop the Data Management [2] 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 life-cycle [2], 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) [3].

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 DMP¹ 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 [4]. 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) [5] 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 [4] comprises of a universal core set of elements that define a DMP. The DMP Common Standard data

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

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.

Ontologies [6] 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 [7]. These describe the ontology in a machine-readable way.

1.2 Thesis Statement

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, compliant with the DCSO.

1.3 Outline

The theses is organised as follows: Section 2 describes the current state of the art of Research Data Management and techniques that help achieve a better Data management such as Data Management Plans. This gives a background information about the problem this theses is trying to solve. Section 3 describes the problem context, where the objectives of the theses are explained. Section 4 describes the solution proposed for each of the problems found during the theses development. Section 5 explains how each solution has developed and shows a demonstration of the final workflow created. Section 6 concludes the thesis, explaining what happened during development and future work.

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

2

State of The Art

Contents

2.1 Research Data Management	7
2.2 Data Management Plan	9
2.3 Semantic Technology	11

With a growing volume of research data being generated [1], 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 Data Management Plan (DMP).

The funding bodies to comply with the open science 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)¹ requires a Technical Plan that summarize the Digital Outputs and Digital Technologies; The Biotechnology and Biological Sciences Research Council (BBSRC)² 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.

This section is divided in three sub sections: Section 2.1 focuses on providing a description of what is Research Data Management because the objective of the theses is to facilitate to achieve a better Research Data Management. Section 2.2 defines what are Data Management Plans because they will be the type of files used through the theses. Lastly, in section 2.3 the Semantic Technology is addressed because a ontology will be used to create knowledge graphs that represent Data Management Plans, in this case it will be used a serialization of DMP Common Standard (DCS) mention on Section 2.2.1.A. The serialization to be used is the DMP Common Standard Ontology (DCSO)³. The DCSO implements the data model through semantic technology, and in particular it resorts to the Web Ontology Language (OWL) [8], a web-based ontology Specification language (see section 2.3.1).

2.1 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 [1].

However, dealing with extremely large quantities of data is not the only challenge. Scientific research data presents three challenges [1]: (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

¹AHRC: <https://ahrc.ukri.org/>

²BBSRC: <https://bbsrc.ukri.org/>

³DMP Common Standard Ontology Repository: <https://github.com/RDA-DMP-Common/RDA-DMP-Common-Standard/tree/master/ontologies>

and data formats are used.

Research data management [9] 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 [2]. An example of a strategy is the usage of Data Management Plans explained in Section 2.2.

2.1.1 Data Lifecycle

The concept of data lifecycle [2], 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.

2.1.2 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 [3] [11] ,(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.

2.2 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 [5].

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. ADMP should include the following information⁴: (1) Description of the data to be collected / created; (2) What data will be collected, processed and/or generated; (3) Which methodology and standards will be applied; (4) Whether data will be shared/made open access; (5) How data will be curated and preserved (including after the end of the project); and (6) Ethics and Intellectual Property concerns or restrictions. It is also important to describe the metadata of the data and metadata standards should be chosen at the beginning of the project. The metadata should be generated throughout the project life cycle during the collecting data stage [12].

2.2.1 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 however software tools that aid researchers in DMP creation. Two of the most popular are DMPonline⁷ and DMPTool⁸. However there are more tools that are used for this purpose, such as the Data Stewardship Wizard (DSW)⁹ (see section 2.3.3.B) or Open Aire's Argos¹⁰.

There are ten simple rules [13] 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.

⁴DCC FAQ-DMP: <http://www.dcc.ac.uk/resources/data-management-plans/faq-data-management-plans> [Retrieved in 03/01/2020]

⁵European 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]

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

⁷DMPonline: <https://dmponline.dcc.ac.uk/>

⁸DMPTool: <https://dmptool.org/>

⁹DSW: <https://ds-wizard.org/>

¹⁰Open Aire's Argos: <http://catalogue.openaire.eu/service/openaire.argos>

2.2.1.A DMP Common Standard

The DMP Common Standard data model [4] is a product of a Research Data Alliance (RDA) working group, the DMP Common Standards Working Group¹¹. 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.

2.2.2 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 [5]. The general perception is that a DMP is a annoying administrative exercise and does not support data management activities [14]. 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 [5] 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 [14]: (1) Integrate DMPs with the workflows of all stakeholders in the research data ecosystem; (2) Allow automated systems to act on behalf of

¹¹DMP Common Standards WG: <https://www.rd-alliance.org/groups/dmp-common-standards-wg> [Retrieved 04/01/2020]

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.

2.3 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 [15], 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 [6] 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.

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 [16].

2.3.1 Ontology Representation Languages

A Ontology representation language [7] "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 [16] it is proposed this classification of ontology representation languages: (1) traditional ontology languages; (2) Web

standards; and (3) Web-based ontology specification languages. In the Table 2.1 are described the various ontology languages.

(1) Traditional Ontology Languages	Enriched predicate logic: The central modeling primitive in these languages are predicates, KIF and CycL are examples of this category of language;
	Frame-base: Classes (frames) are the central modeling primitives, Ontolingua and F-logic are examples of languages from this category;
	Description logic: Define concepts in terms of descriptions that specify the properties the objects must satisfy in order to belong to that concept, Loom is an example of this type of language;
	Others: The language Telos does not belong to any of the groups.
(2) Web Standards	XML is the universal format for structured documents and data on the Web but it is not an ontology language and XML-schema is created mainly for verification of XML so it will not be viewed as ontology language.
	RDF is an infrastructure for encoding, exchange and reuse of structured metadata, but RDF does not have any mechanisms for defining relationships between resources, properties and statements (the three types of object supported), but RDFS can define relationships so can be used directly to describe ontologies.
(3) Web-based Ontology Specification Languages	These are languages created with web standards as basis, the most relevant ontology specification languages are the DAML+OIL and OWL and both are based in the RDF.

Table 2.1: Ontology Languages Categories

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

OWL [17] 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) [18] 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.

2.3.2 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. SPARQL¹² 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 [19] 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.

SPARQL [20] 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 in¹³.

2.3.3 Technical Resources

Considering the ontology representation languages described in section 2.3.1 are expressed in triples, it becomes necessary to analyse the software frameworks and services that either handle triples or provide triplestores [21]. A triplestore is a purpose-built database for the storage and retrieval of data expressed in triples. This section list some of the most relevant software frameworks and services for triple handling and storage.

2.3.3.A Ontology creation API

The OWL API [22] has been designed to meet the needs of people developing OWL based applications, OWL editors and OWL reasoners. It is a high level API that is closely aligned with the OWL 2 specification. So it will be used to open an ontology and populate it with entities in the application that convert a DSW Document in JSON to a knowledge graph (see in section 5.1).

The OWL API consists of a set of interfaces for inspecting, manipulating and reasoning with OWL ontologies. The OWL API supports loading and saving ontologies is a variety of syntaxes. However,

¹²SPARQL: <https://www.w3.org/TR/rdf-sparql-query/>

¹³SPARQL Endpoints Status: <https://labs.mondeca.com/sparqlEndpointsStatus/index.html> [Retrieved in 03/01/2020]

none of the model interfaces in the API reflect, or are biased to any particular concrete syntax or model. This means that an ontology is simply viewed as a set of axioms and annotations.

Besides the model interfaces for representing entities, class expressions and axioms, it is necessary to have certain machinery that allow applications to manage ontologies. Where the `OWLOntologyManager` provides a central point for creating, loading, changing and saving ontologies. This centralised management design allows client applications to have one access point to ontologies, to provide redirection mechanisms and other customisations for loading ontologies, and allows client applications to monitor all changes that are applied to any loaded ontologies.

2.3.3.B DMP creation tools

DSW [23] is a tool to create data management plans by filling questionnaires, that tries to alleviate the negative view of data management planning by focusing primarily on the benefits of data management for the research project itself and the researcher, not on the obligations. The DSW clearly indicates the effect of each answer on the adherence to the principles describing that data should be Findable, Accessible, Interoperable and Reusable for machines and for humans (FAIR principles) in all its questions, guiding researchers who are searching for good ways to make their results FAIRer. DSW have taken the approach where most questions are closed questions with a limited set of possible answers. And based on the answer that is selected by the user, follow-up questions will be added to the questionnaire. Also, some answers may be obtained from linked services.

2.3.3.C Triplestores

The Apache Jena framework¹⁴ provides a extensive Java libraries for helping developers develop code that handles RDF, RDFS, RDFa, OWL and SPARQL in line with published W3C recommendations. A model can be sourced with data from files, databases, URL or a combination of these. SPARQL is fully supported for querying the models. Apache Jena Fuseki is an HTTP interface to RDF data. It supports SPARQL for querying and updating. The project is a sub-project of Jena and is developed as servlet. It can run as a operating system service, as a Java web application (WAR file), and as a standalone server. Fuseki is tightly integrated with TDB (a component of Jena for RDF storage and query. It supports the full range of Jena APIs) to provide a robust, transactional persistent storage layer, and incorporates Jena text query. It can be used to provide the protocol engine for other RDF query and storage systems.

RDF API for PHP (RAP)¹⁵ can store statements in system memory in arrays or in a relational database. It implements SPARQL query.

¹⁴Jena: <https://jena.apache.org>

¹⁵RAP: <http://wifo5-03.informatik.uni-mannheim.de/bizer/rdxfapi/>

Sesame¹⁶ is a framework for querying and analyzing RDF data. It has an RDF query language SeRQL. Sesame can be sourced with data from memory, disk or a relational database.

4Store¹⁷ is a triplestore that is a storage and query engine. It does not provide many features over and above RDF storage and SPARQL queries.

Mulgara¹⁸ is a triplestore scalable and transaction-safe. It can be queried via iTQL and SPARQL. Mulgara is not based on a relational database, it is optimized for metadata management, using RDF triples.

AllegroGraph¹⁹ is a triplestore designed to store RDF tuples and implements SPARQL query. AllegroGraph does not use relational database. It loads triples through RDF/XML, N-Quads2 and N-Triples.

2.3.3.D Knowledge graphs visualization

Protégé²⁰ is software tool that aids in the development of ontologies. It is Java based, extensible, and provides a plug-and-play environment that makes it a flexible base for rapid prototyping and application development. It fully supports the latest OWL 2 Web Ontology Language and RDF specifications from the World Wide Web Consortium.

Pubby²¹ can be used to add Linked Data interfaces to SPARQL endpoints. It is designed to provide a Linked Data interface to RDF data sources that can be accessed only by SPARQL client applications that use the SPARQL protocol.

Pubby will handle requests to the mapped URIs by connecting to the SPARQL endpoint, asking it for information about the original URI, and passing back the results to the client. It also handles various details of the HTTP interaction, such as the 303 redirect required by Web Architecture, and content negotiation between HTML, RDF/XML and Turtle descriptions of the same resource.

¹⁶Sesame: <https://rdf4j.org>

¹⁷4Store: <https://github.com/4store/4store>

¹⁸Mulgara: <http://mulgara.org>

¹⁹AllegroGraph: <https://allegrograph.com>

²⁰Protégé: <https://protege.stanford.edu>

²¹Pubby: <http://wifo5-03.informatik.uni-mannheim.de/pubby/> [Retrieved 16/10/2020]

3

Problem Context

Contents

3.1 Problem context	17
-------------------------------	----

3.1 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 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?" explained in Section 4.3. (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.

4

Proposed Solution

Contents

4.1 Workflow to create a maDMP	19
4.2 Service to Visualize DMP	20
4.3 ELIXIR	20

This chapter details the proposed solution to the problems raised in chapter 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.

4.1 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?" as explained in Section 4.3.

DSW have a simple and intuitive interface, and make the questionnaires based on the information needed to fill a DMP Common Standard Ontology (DCSO) that is a serialization of the DCS explained in Section 2.2.1.A.

(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 diagram in Figure 4.1. 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.

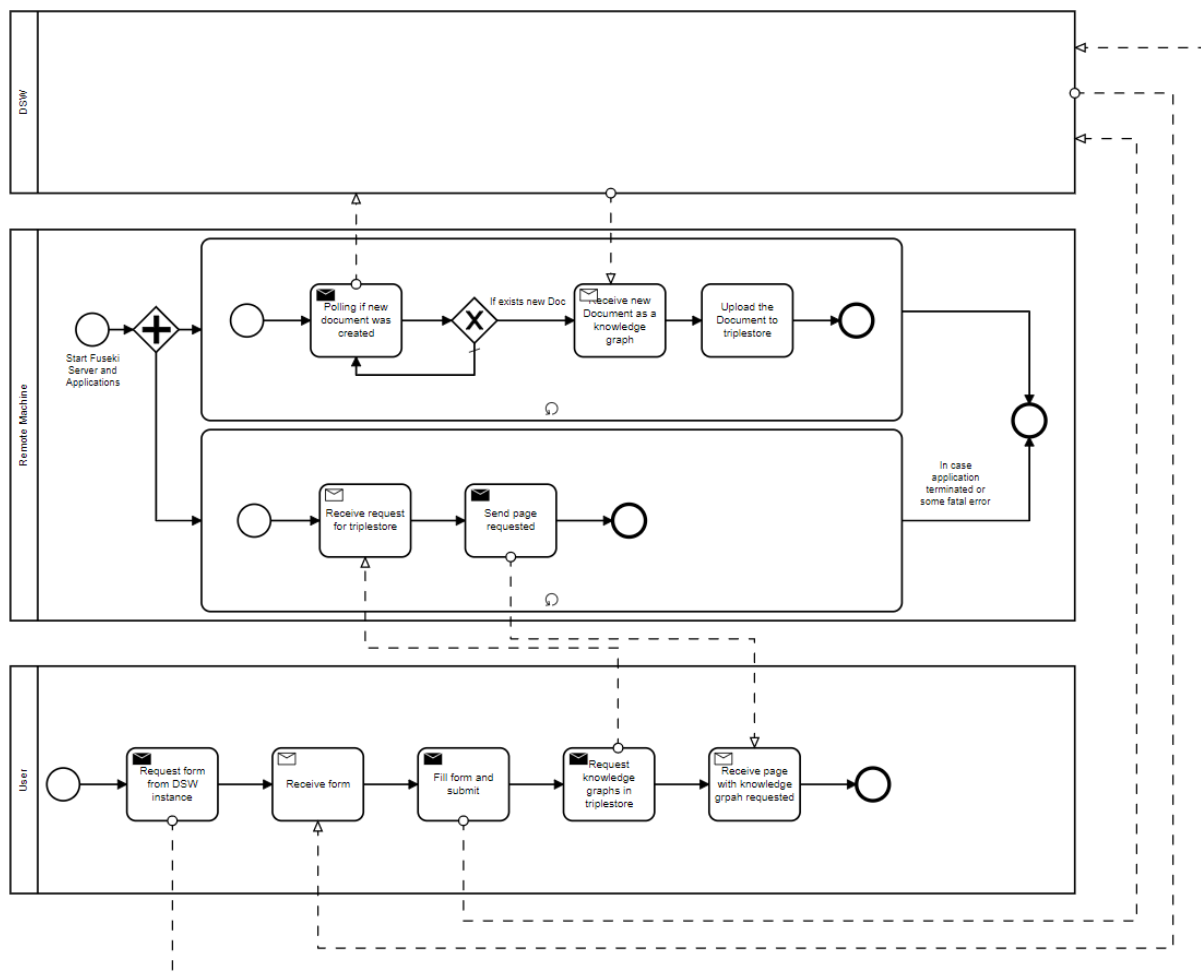


Figure 4.1: Workflow BPMN

4.2 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.

4.3 ELIXIR

ELIXIR is a European intergovernmental organisation whose objective is to bring together life science resources (like databases, software tools, training materials, cloud storage and supercomputers), and

coordinating them so that they form a single infrastructure¹. With this infrastructure it is easier to both find and share data, exchange expertise and agree on best practices to help researchers in the life science community.

ELIXIR comprises of 23 member states, that are organised in a Hub and Nodes model. Where a Hub² can be interpreted as the "headquarters" and plays the following role:

- Accommodation of executive management and administrative staff;
- Development and delivery of the scientific strategy of ELIXIR;
- Coordination of Node services;
- Coordination and support of ELIXIR governance bodies and technical committees;
- Collaboration with third party biomedical science infrastructures;
- Coordination of communications and external relations activities of ELIXIR;
- Support of institutions within the Nodes;
- Collaboration with both funders and policy-makers, on a national and European level.

Nodes in ELIXIR can be interpreted as a network of organisations that work within a member state.

In ELIXIR activities are divided into five areas called 'Platforms'³. They are: Data, Tools, Interoperability, Compute and Training.

The Data Platform⁴ focuses on drive the use, re-use and value of life science data. It aims to do this by providing users with robust, long-term sustainable data resources within a coordinated, scalable and connected data ecosystem. The Data Platform provides three services: ELIXIR Core Data Resources, ELIXIR Deposition Databases and Database services listing.

- ELIXIR Core Data Resources: European data resources that are of fundamental importance to research in the life sciences and are committed to the long-term preservation of data;
- ELIXIR Deposition Databases: repositories recommended for the deposition of life sciences experimental data;
- Database services listing: this list is updated as Nodes finalise or review their Service Delivery Plans.

¹About us: <https://elixir-europe.org/about-us> [Retrieved in 03/01/2020]

²ELIXIR Hub: <https://elixir-europe.org/about-us/who-we-are/hub> [Retrieved in 03/01/2020]

³ELIXIR Platforms: <https://elixir-europe.org/platforms> [Retrieved in 20/12/2019]

⁴Data: <https://elixir-europe.org/platforms/data> [Retrieved in 04/01/2020]

The Tools Platform goal⁵ aims to improve the discovery, quality and sustainability of software resources. It offers three services: bio.tools, Biocontainers and OpenEBench.

- Bio.tools: search the registry of software tools and data resources for life sciences;
- Biocontainers: browse the list of software you can run on any operating system;
- OpenEBench: browse tools for benchmarking and monitoring software.

The Compute Platform⁶ has the goal of integrating cloud, compute, storage and access services for the life-science research community. This infrastructure allows researchers access, share and analyse data from different sources across Europe. The Compute Platform offers a single service. The Authentication and Authorization Infrastructure (AAI) service provides a user access and identity management service for academia and industry.

The Interoperability Platform⁷ has the objective of helping both humans and machines to discover, access, integrate and analyse biological data. It encourages the life science community to adopt standardised file formats, metadata, vocabularies and identifiers and works internationally to achieve its goals.

Finally the Training Platform⁸ aims to strengthen national training programmes, grow bioinformatics training capacity and competence across Europe, and empower researchers to use services and tools provided by ELIXIR.

ELIXIR also hosts several Communities⁹, that bring together experts across Europe to develop standards, services and training within specific life science domains. The Communities also provide feedback on the Platform services, thus ensuring that these services remain practical and useful.

4.3.1 ELIXIR Portugal

ELIXIR Portugal is a consortium of four Portuguese research institutions which are part of the national biological information network, BioData.pt¹⁰.

The main focus of ELIXIR Portugal is the Woody Plants domain and aims to provide data, tools, standards, and training in this domain and thus contribute to build an ELIXIR framework that is of added-value to all woody plant based industries. As found in ¹¹

ELIXIR Portugal provides four categories of services: (1) Data Resources, (2) Training, (3) Software Tools and (4) Compute.

⁵Tools: <https://elixir-europe.org/platforms/tools> [Retrieved 04/01/2020]

⁶Compute: <https://elixir-europe.org/platforms/compute> [Retrieved 04/01/2020]

⁷Interoperability: <https://elixir-europe.org/platforms/interoperability> [Retrieved 04/01/2020]

⁸Training: <https://elixir-europe.org/platforms/training> [Retrieved 04/01/2020]

⁹Communities: <https://elixir-europe.org/communities> [Retrieved in 20/12/2019]

¹⁰BioData.pt: <https://biodata.pt/> [Retrieved in 04/01/2020]

¹¹ELIXIR Portugal: <https://elixir-europe.org/about-us/who-we-are/nodes/portugal> [Retrieved in 20/12/2019]

4.3.1.A Data Resources

CorkOakDB¹² [24] aims to integrate the knowledge generated from fundamental and applied studies about *Quercus suber*, with a focus on genetics. CorkOakDB features the first draft genome of *Quercus suber* and allows genome browsing and gene search. It also incorporates other types of data from cork oak scientific research, including gene expression data from publicly available datasets.

Plant Experimental Assay Ontology (PEAO)¹³ [25] is an attempt at enabling the creation of a repository of data produced by plant experimental assays. The representation of the data using an ontology elicits the preservation of the semantic relationships between the entities represented therein, which facilitates the interpretation of the results and the integration of data produced by different experiments.

4.3.1.B Training

Gulbenkian Training Programme in Bioinformatics (GTPB)¹⁴ runs face-to-face Bioinformatics Training Courses regularly at the Instituto Gulbenkian de Ciência. The Programme consists in a series of short, intensive hands-on courses delivered and fully documented in English. The design of the courses is based on sets of carefully chosen exercises, flanked by short lectures and participative interaction sessions.

4.3.1.C Software tools

PHYLOViZ¹⁵ [26] is a JAVA application and also an online application. It allows the analysis of sequence-based typing methods that generate allelic profiles and their associated epidemiological data. For representing the possible evolutionary relationships between strains, PHYLOViZ uses the goeBURST algorithm, a refinement of eBURST algorithm by Feil et al, and its expansion to generate a complete minimum spanning tree (MST).

sRNA Portal¹⁶ workflow is an online resource integrating a workflow for identifying, predicting and annotating plant small RNAs (sRNA), and a database to store sRNA datasets with standardized metadata. The workflow (miRPursuit) is based on publicly available tools and allows the automated sequential application of data processing modules for analysis of sRNA datasets, including data preprocessing, filtering, annotation and reporting output.

Yeast Search for Transcriptional Regulators And Consensus Tracking (YEASTRACT)¹⁷ [27] is a curated repository of approximately 175.000 regulatory associations between transcription factors (TF) and

¹²CorkOakDB: <http://corkoakdb.org/>

¹³PEAO: <https://bitbucket.org/PlantExpAssay/ontology/src/master/>

¹⁴GTPB: <http://gtpb.igc.gulbenkian.pt/bicourses/index.html> [Retrieved in 20/12/2019]

¹⁵PHYLOViZ: <http://www.phyloviz.net/>

¹⁶sRNA: <https://srna-portal.biodata.pt/> [Retrieved in 20/12/2019]

¹⁷YEASTRACT: <http://www.yeasttract.com/>

target genes in *Saccharomyces cerevisiae*, based on more than 1580 bibliographic references. It also includes the description of 310 specific DNA binding sites shared among 183 characterized TFs.

4.3.1.D Computing

ELIXIR PT Computing Services¹⁸ is provided by Portuguese academic institutions, with complementary services hosted at the Google Cloud and Amazon Web Services. It is managed by Instituto de Engenharia de Sistemas e Computadores - Investigação e Desenvolvimento (INESC-ID) and includes creation of virtual machines, deployment of instances and deployment and management of instances at Google Cloud¹⁹, or any other OpenStack²¹ platform, running biocomputing software for a fee.

4.3.2 Data management in Elixir

ELIXIR Europe has an ongoing effort in providing Data Management services. The Data Management Working Group²² was established with the focus on activities that spread the expertise on data management between nodes and that make this expertise available to life science researchers and their projects. Its goals are to identify existing resources for Data Management training, make new materials for Data Management training, advertise ELIXIR data management expertise to the research community, and help researchers get the most out of their data with the least risk.

The DSW²³ [23] is a tool that emerged from the Data Management Working Group. It is a data managing tool that focuses on Data Management Plan (DMP) (see section 2.2) creation, by exposing users to a hierarchical set of questions. These questionnaires enable the DSW to assess and optimize the FAIRness (see section 2.1) of a given DMP.

BioData.pt, and subsequently the Portuguese ELIXIR Node, has also strove to create a Data Management service. One of their offers is a programme called "Ready for BioData Management?"²⁴. Its objective is to empower researchers and institutions in managing their data more effectively and efficiently. Three different versions of the programme exist: (1) One-Day Workshop for a Introduction to Data Management Plans, (2) One-Day Course for Advanced Data Management Plans, and (3) Class Modules for a Introduction to Data Management In Science and Demystifying Data Management Plans. BioData.pt has plans to create an One-Day Advanced DMP Workshop, where the goal will be to have participants create a DMP with their own data. These DMP documents are then to be stored and liable for processing by BioData.pt. This One-Day Advanced DMP Workshop is set to have its first edition in

¹⁸BioData.pt Computing Services: <https://biodata.pt/resource/elixir-pt-computing-services> [Retrieved in 20/12/2019]

¹⁹Google Cloud: ²⁰ [Retrieved 05/01/2020]

²¹Openstack: <https://www.openstack.org/> [Retrieved 05/01/2020]

²²ELIXIR Data Management Plans Group: <https://elixir-europe.org/platforms/training/data-management-group> [Retrieved in 20/12/2019]

²³DSW: <https://ds-wizard.org/>

²⁴Ready for BioData Management?: <http://ready4biodatamanagement.biodata.pt/> [Retrieved in 04/01/2020]

January 2020. During the Advanced DMP Wrokshop the DMP will be created using DSW, so with this purpose a DSW instance was created.

5

Implementation

Contents

5.1 Create the DMP	27
5.2 Preserve the DMP	29
5.3 Visualize the DMP	31
5.4 Workflow Demonstration	33

This chapter offers both a detailed description of the implementation, as well as a demonstration of each steps in the proposed solution, presented in chapter 4.

5.1 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 Research Data Alliance (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.

5.1.1 Convert Data Stewardship Wizard Document to a DMP Common Standard Ontology

5.1.1.A 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.

As it can be seen in the figure 5.1;

5.1.1.B 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

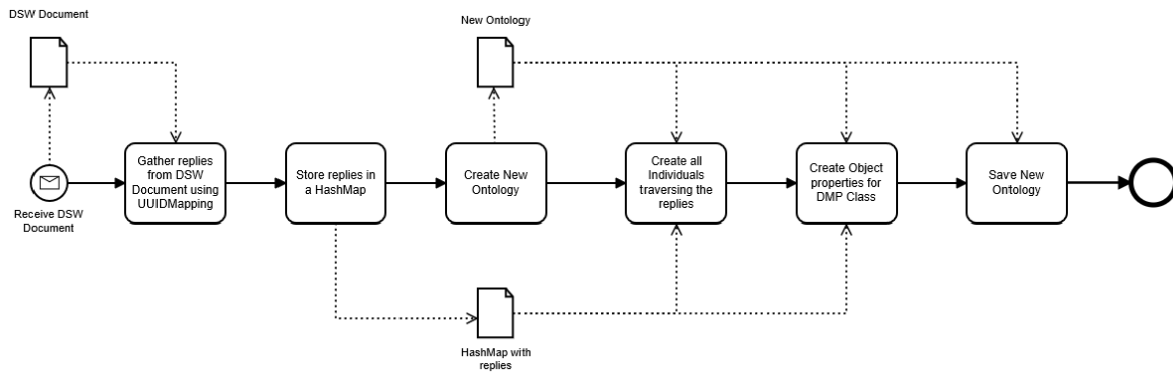


Figure 5.1: DSWtoDCSO diagram

- 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).

5.1.1.C 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 created 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.

5.1.1.D 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).

5.1.2 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.

5.2 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.

5.2.1 DSWEExport

DSWEExport 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 DSWEExport and DSW or between DSWEExport and Fuseki Server are through HTTP requests.

This process can be observed in the figure 5.2.

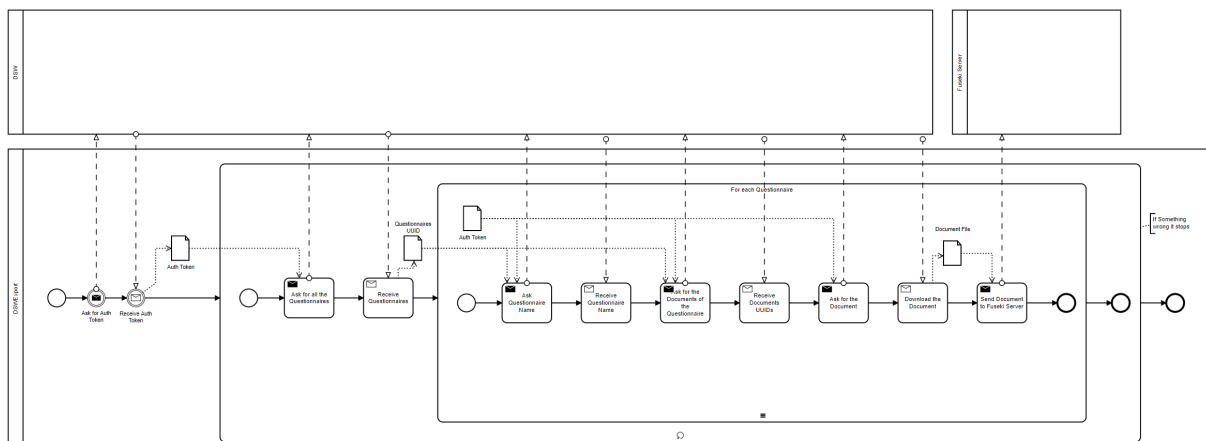


Figure 5.2: DSWEExport BPMN

The architecture of DSWEExport comprises three Packages:

- DSWEExport - 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.

5.3 Visualize the DMP

With the DMPs 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.

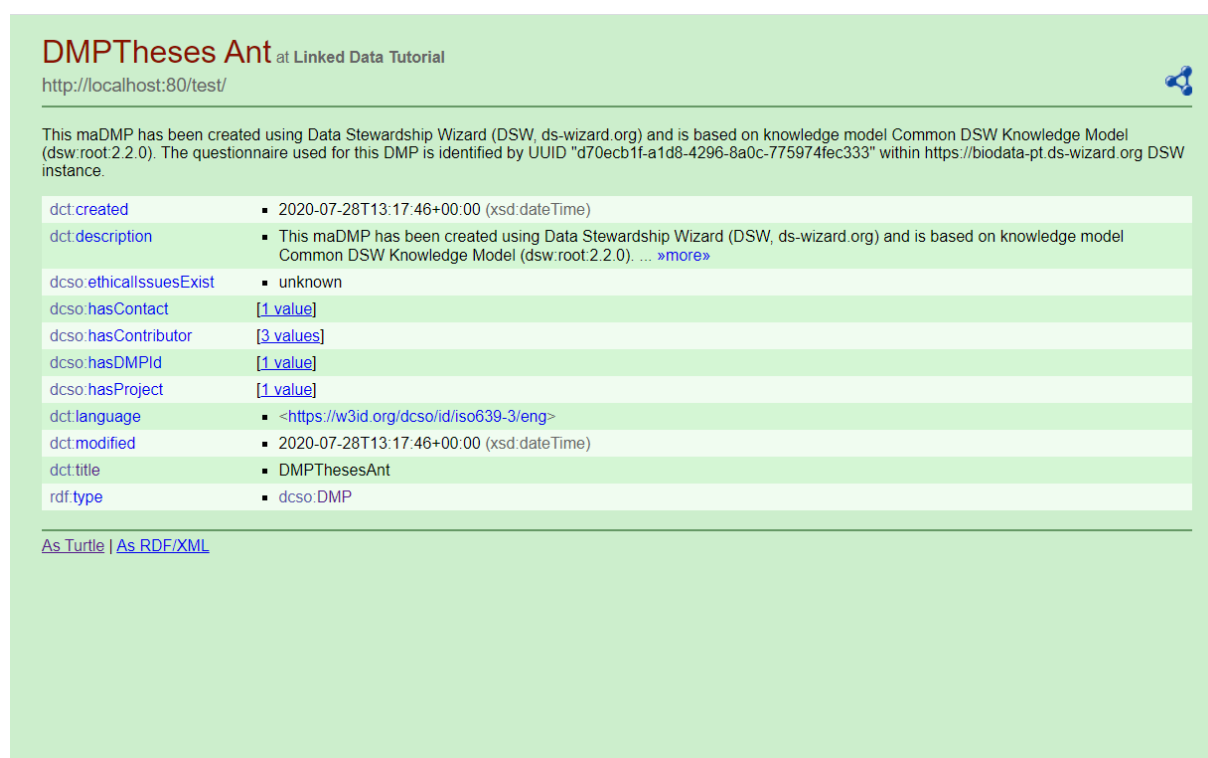
The first one to be tested was Pubby but I was having some troubles with it, as explained in Section 5.3.1. So some alternatives to Pubby were tested, the alternatives was LODDY ¹ and MadsHolten/sparql-visualizer ² but I could not get them working properly.

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.

5.3.1 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, as seen in the figure 5.3, but when I try to open a encapsulated value like hasContact it gives the error 404 not found, as seen in the figure 5.4, but the turtle output of the Pubby has that information, as seen in the figure 5.5.



The screenshot shows the Pubby interface for 'DMPTheses Ant' at 'Linked Data Tutorial'. The URL is 'http://localhost:80/test/'. A description states: 'This maDMP has been created using Data Stewardship Wizard (DSW, ds-wizard.org) and is based on knowledge model Common DSW Knowledge Model (dsw.root:2.2.0). The questionnaire used for this DMP is identified by UUID "d70ecb1f-a1d8-4296-8a0c-775974fec333" within https://biodata-pt.ds-wizard.org DSW instance.'

dct:created	2020-07-28T13:17:46+00:00 (xsd:dateTime)
dct:description	This maDMP has been created using Data Stewardship Wizard (DSW, ds-wizard.org) and is based on knowledge model Common DSW Knowledge Model (dsw.root:2.2.0). ... »more»
dcso:ethicalIssuesExist	unknown
dcso:hasContact	[1 value]
dcso:hasContributor	[3 values]
dcso:hasDMPId	[1 value]
dcso:hasProject	[1 value]
dct:language	< https://w3id.org/dcso/id/iso639-3/eng >
dct:modified	2020-07-28T13:17:46+00:00 (xsd:dateTime)
dct:title	DMPThesesAnt
rdf:type	dcso:DMP

At the bottom, there are links for 'As Turtle' and 'As RDF/XML'.

Figure 5.3: Pubby interface

¹LOBBY: <https://bitbucket.org/art-uniroma2/loddy/src/master/> [retrieved in 10/12/2020]

²MadsHolten: <https://github.com/MadsHolten/sparql-visualizer> [retrieved in 10/12/2020]

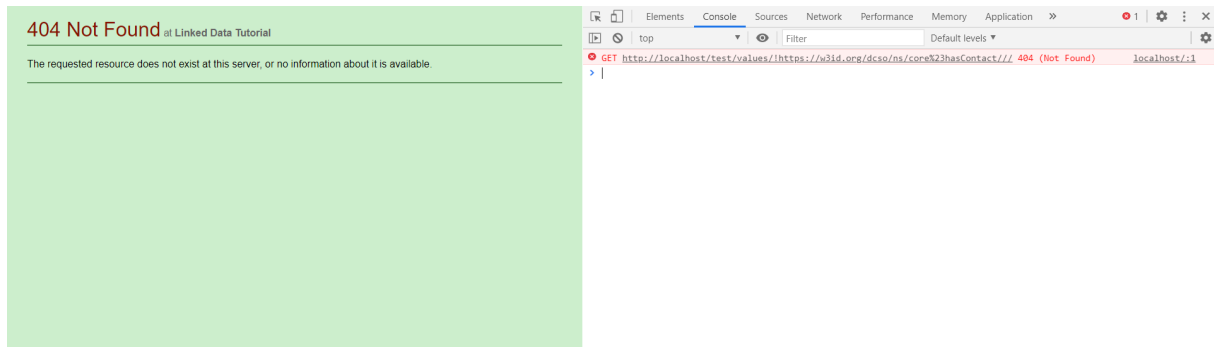


Figure 5.4: Pubby hasContact values

```

@prefix rdf: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .

<http://localhost:80/test/data/?output=ttl>
  rdfs:label "RDF description of DMPTheses Ant" ;
  foaf:primaryTopic <http://localhost:80/test/> .

<http://localhost:80/test/>
  a dcs:DMP ;
  dct:created "2020-07-28T13:17:46+00:00"^^xsd:dateTime ;
  dct:description "This maDMP has been created using Data Stewardship Wizard (DSW, ds-wizard.org) and is based on knowledge model Common DSW Knowledge Model (dsw:root:2.2.0). The questionnaire used for this DMP is identified by UUID \"d70ecb1f-aid8-4296-8a0c-775974fec333\" within https://biodata-pt.ds-wizard.org DSW instance." ;
  dct:language <https://w3id.org/dcs/1d/1so639-3/eng> ;
  dct:modified "2020-07-28T13:17:46+00:00"^^xsd:dateTime ;
  dct:title "DMPThesesAnt" ;
  dcs:ethicalIssuesExist "unknown" ;
  dcs:hasContact [ a dcs:Contact ;
                  rdfs:seeAlso <http://localhost:80/test/values.data/https://w3id.org/dcs/ns/core#23hasContact//> ;
                  foaf:mbox "antonio.terra@tecnico.ulisboa.pt" ;
                  foaf:name "António Terra" ;
                  dcs:hasContactId [ a dcs:ContactId ;
                                   dct:identifier "00000" ;
                                   dcs:identifier_type "orcid"
                                 ]
                ] ;
  dcs:hasContributor [ a dcs:Contributor ;
                      rdfs:seeAlso <http://localhost:80/test/values.data/https://w3id.org/dcs/ns/core#23hasContributor//> ;
                      foaf:mbox "jlb@tecnico.ulisboa.pt" ;
                      foaf:name "José Borbinha" ;
                      dcs:hasContributorId [ a dcs:ContributorId ;
                                             dct:identifier "22222" ;
                                             dcs:identifier_type "orcid"
                                           ] ;
                      dcs:role "supervisor"
                    ] ;
  dcs:hasContributor [ a dcs:Contributor ;
                      rdfs:seeAlso <http://localhost:80/test/values.data/https://w3id.org/dcs/ns/core#23hasContributor//> ;
                      foaf:mbox "antonio.terra@tecnico.ulisboa.pt" ;
                      foaf:name "António Terra" ;
                      dcs:hasContributorId [ a dcs:ContributorId ;
                                             dct:identifier "00000" ;
                                             dcs:identifier_type "orcid"
                                           ] ;
                      dcs:role "contact person"
                    ] ;
  dcs:hasContributor [ a dcs:Contributor ;
                      rdfs:seeAlso <http://localhost:80/test/values.data/https://w3id.org/dcs/ns/core#23hasContributor//> ;
                      foaf:mbox "antonio.terra@tecnico.ulisboa.pt" ;
                      foaf:name "António Terra" ;
                      dcs:hasContributorId [ a dcs:ContributorId ;
                                             dct:identifier "00000" ;
                                             dcs:identifier_type "orcid"
                                           ] ;
                      dcs:role "contact person"
                    ] ;

```

Figure 5.5: Pubby Turtle Output

5.3.2 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 achieved 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.

5.4 Workflow Demonstration

To begin the workflow a synthetic DMP will be created, a DMP of this theses, as seen in Figure 5.6.

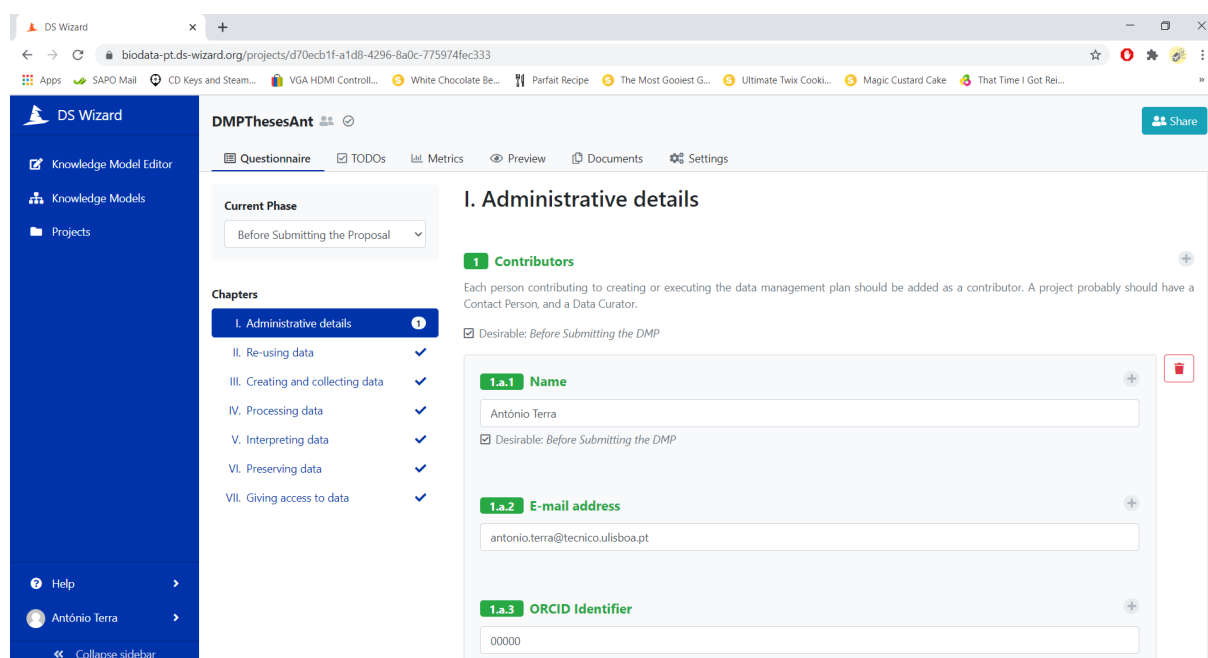


Figure 5.6: DSW questionnaire for synthetic DMP

There are real DMPs already present in the DSW instance of BioData.pt, as seen in Figure 5.7, this DMPs were collected during the "Ready for BioData Management?" programme, is explained in more detail in Section 4.3.

5.4.1 Convert Data Stewardship Wizard Document to a DMP Common Standard Ontology

To demonstrate the functionality of the application created to convert DSW Document in a DCSO I will start to gather a document from the BioData.pt DSW instance, I pick the Data Management Plan

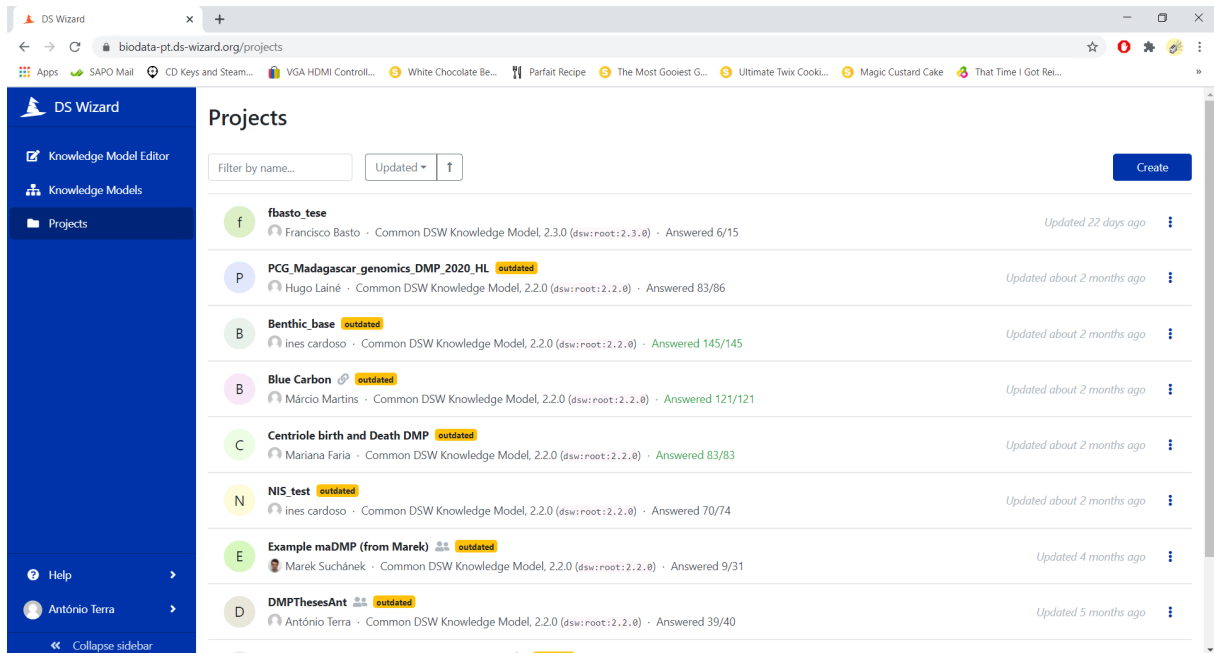


Figure 5.7: DSW list of DMPs

Unidade de Genómica as shown in Figure 5.8 because it uses the Knowledge Model from a old version of the DCSO, the one it was more updated when the application was created.

Then the downloaded Document was renamed to "genomaDSW" and moved to the folder resources in the application folder. The input of the application was altered to the new Document as seen in the Figure 5.9.

And run the application as shown in Figure 5.10, the messages in the terminal are informing that some of the answers are not compliant with the requirements of the DCSO.

The new File with DCSO and the entities form the DSW Document can be visualized in Protegé as shown in Figure 5.11.

5.4.2 DSWExport

To demonstrate the application DSWExport a Fuseki server will run in a remote machine as shown in Figure 5.12.

In the beginning the Fuseki sever is empty, does not have any Datasets, as shown in Figure 5.13.

Then the DSWExport service is executed as shown in Figure 5.14.

After DSWExport have been executed we can see that new Datasets are present in the Fuseki Server as shown in Figure 5.15.

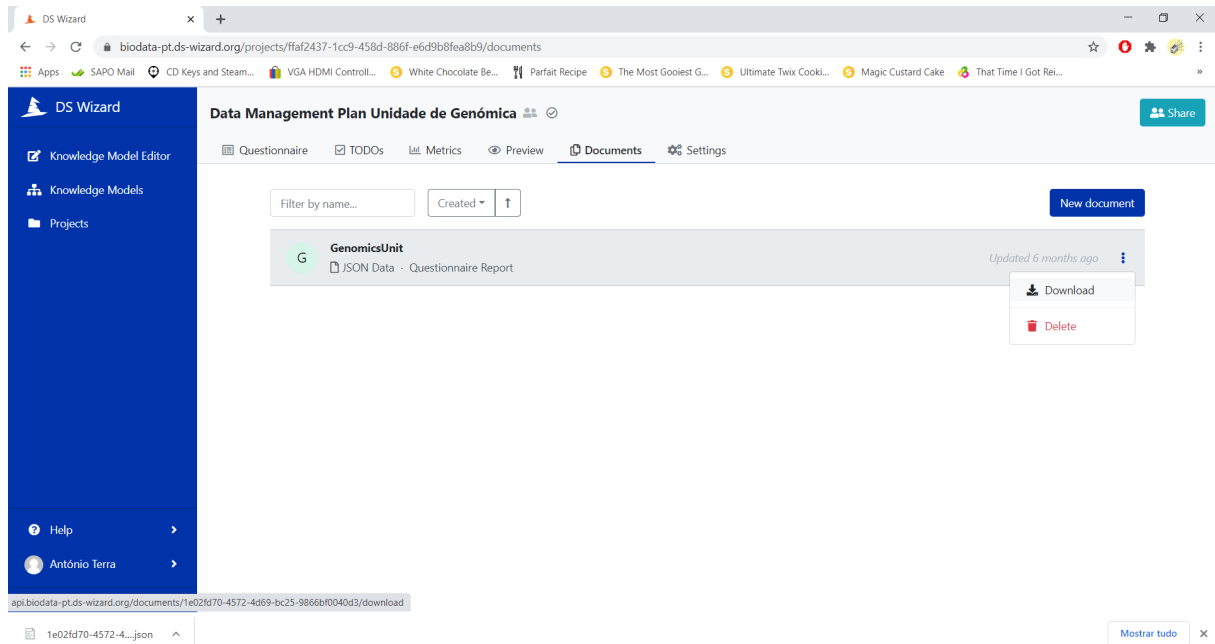


Figure 5.8: DSW Document to be used in the demonstration

5.4.3 Ontology Visualizer

To demonstrate the service Ontology Visualizer the same Fuseki Server is running, and the service is initialized as shown in Figure 5.16.

It can be seen that the Ontology Visualizer is running in the Figure 5.17 and is listing all the Datasets present in Fuseki Server.

If one of the Datasets is clicked, in this case the `Benthic_base` has clicked, it is listed all the content of that Dataset as shown if Figure 5.18.

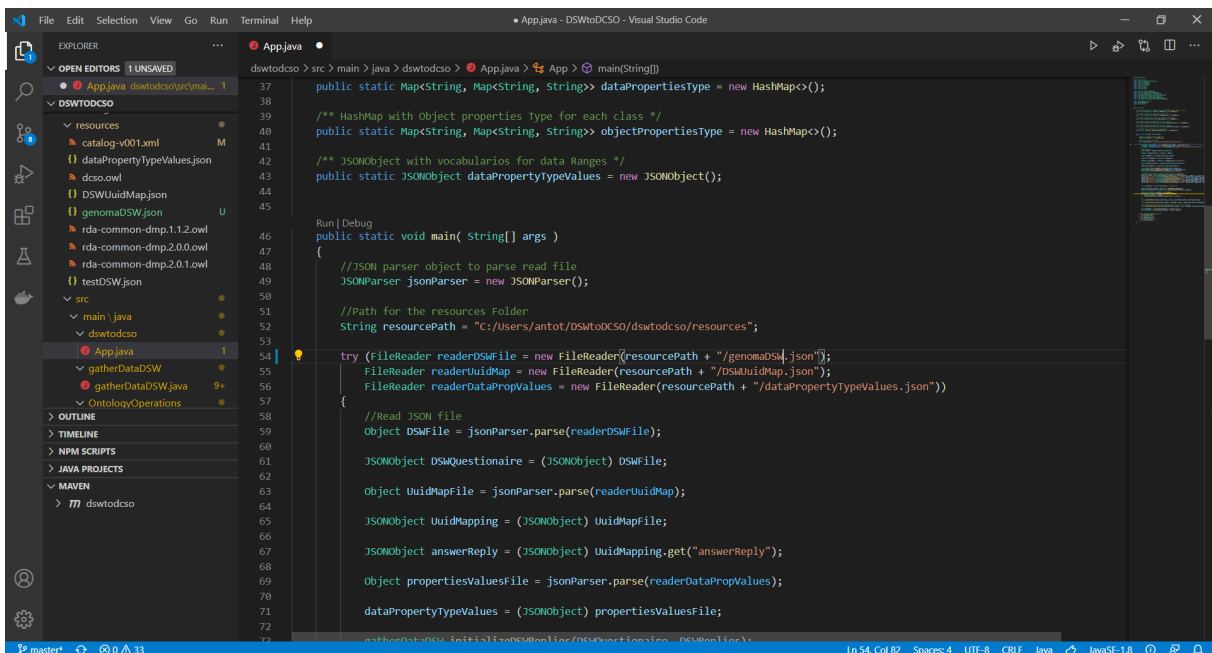


Figure 5.9: Input changed to the Document Downloaded

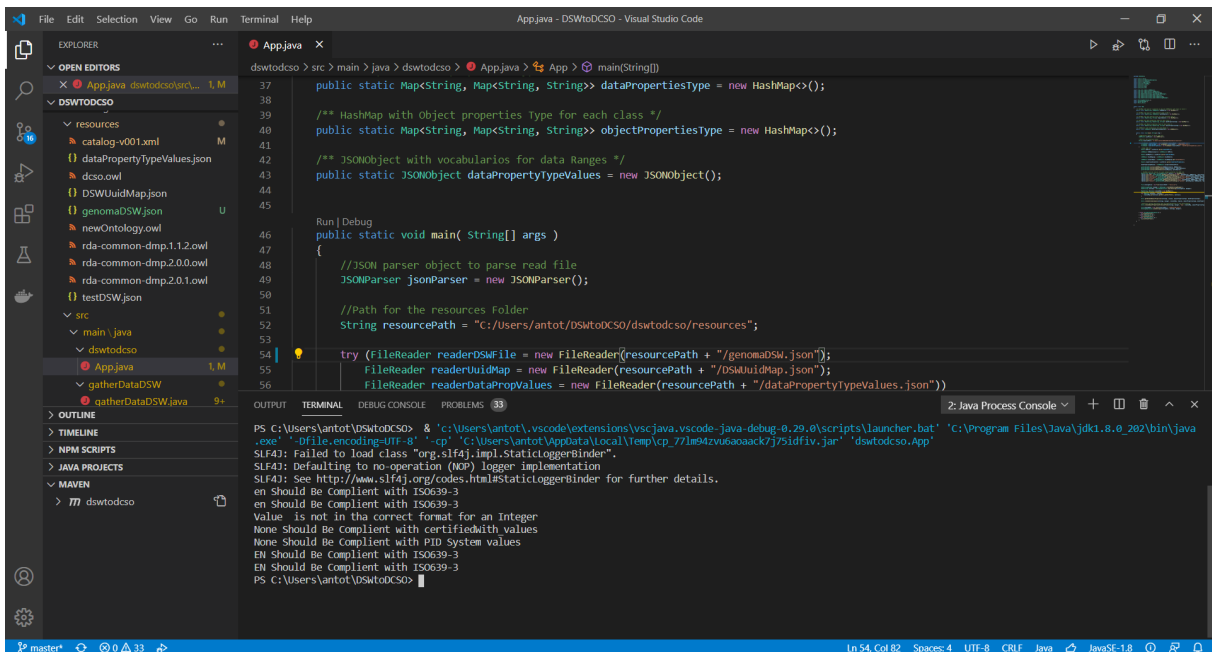


Figure 5.10: Execution of DSWtoDCSO

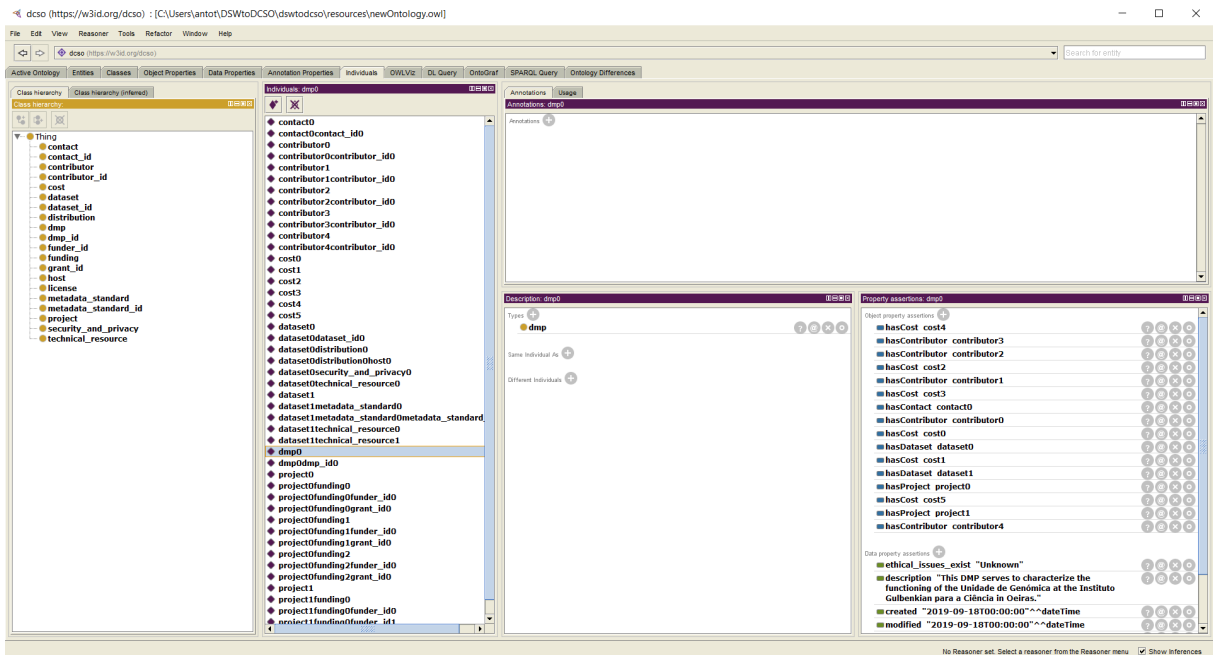


Figure 5.11: Visualization of new Ontology File

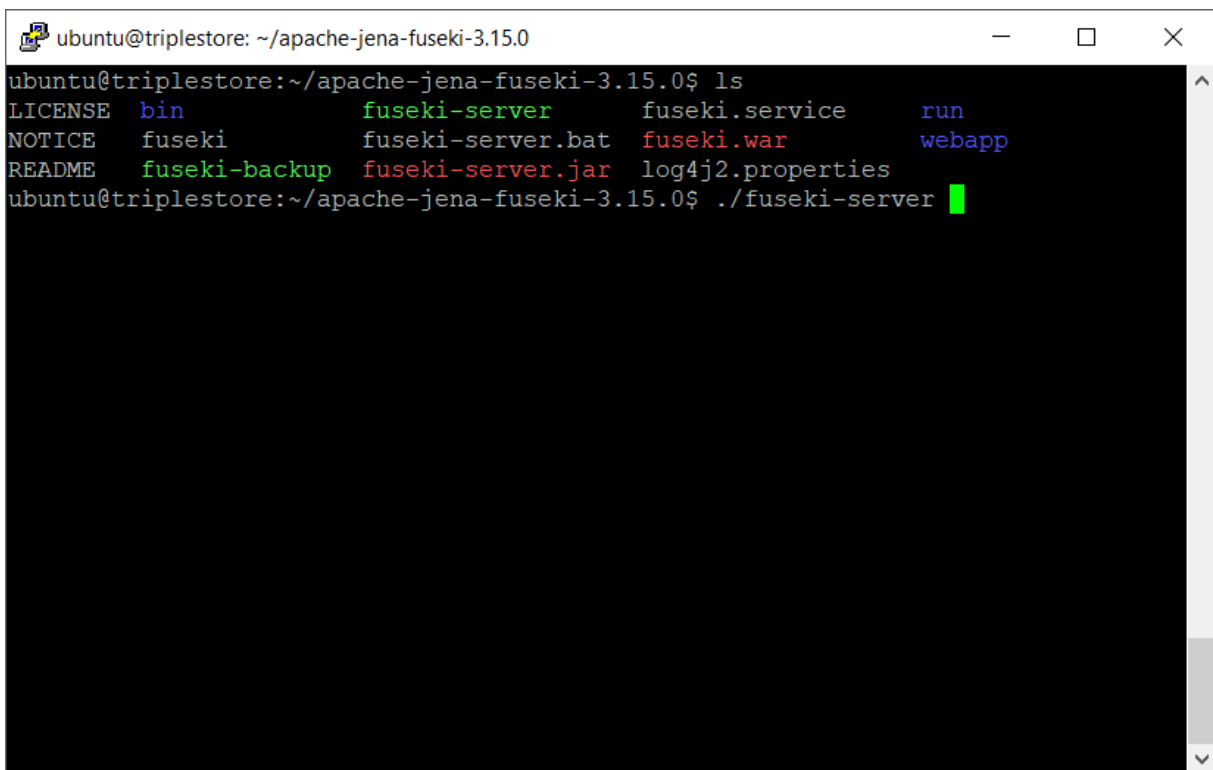


Figure 5.12: Start Fuseki Server

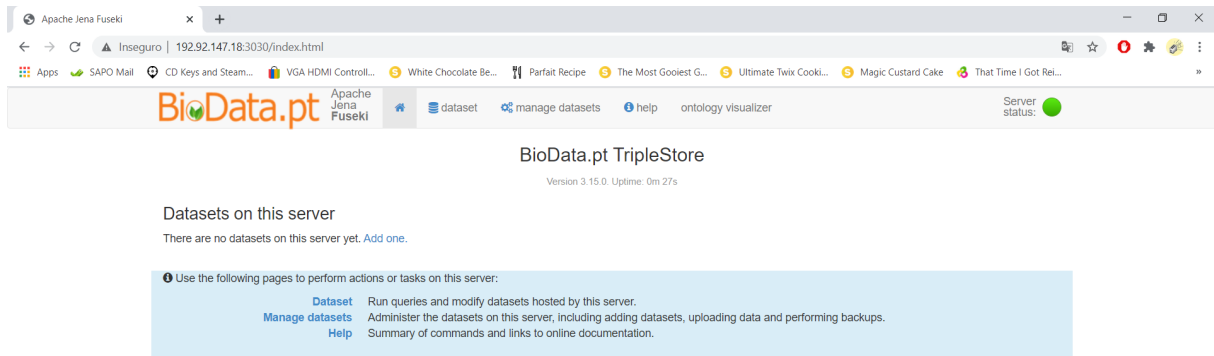


Figure 5.13: Fuseki is empty

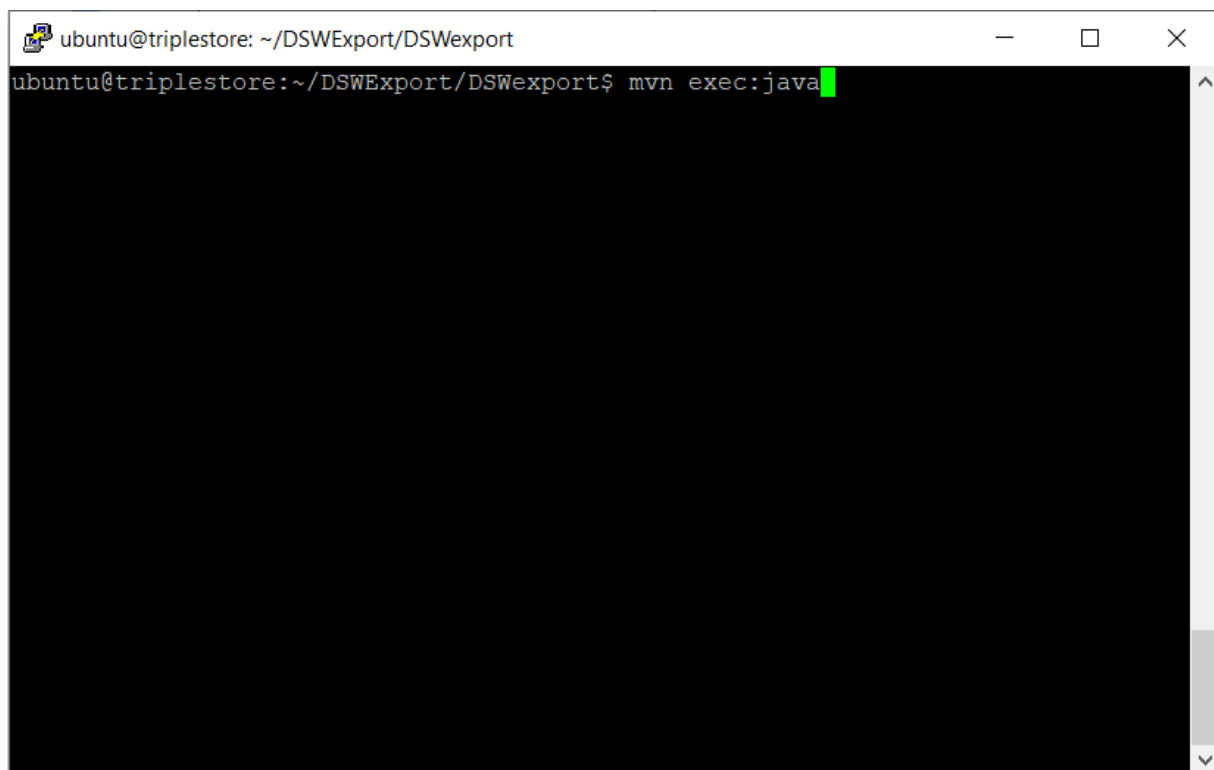


Figure 5.14: Execute DSWExport

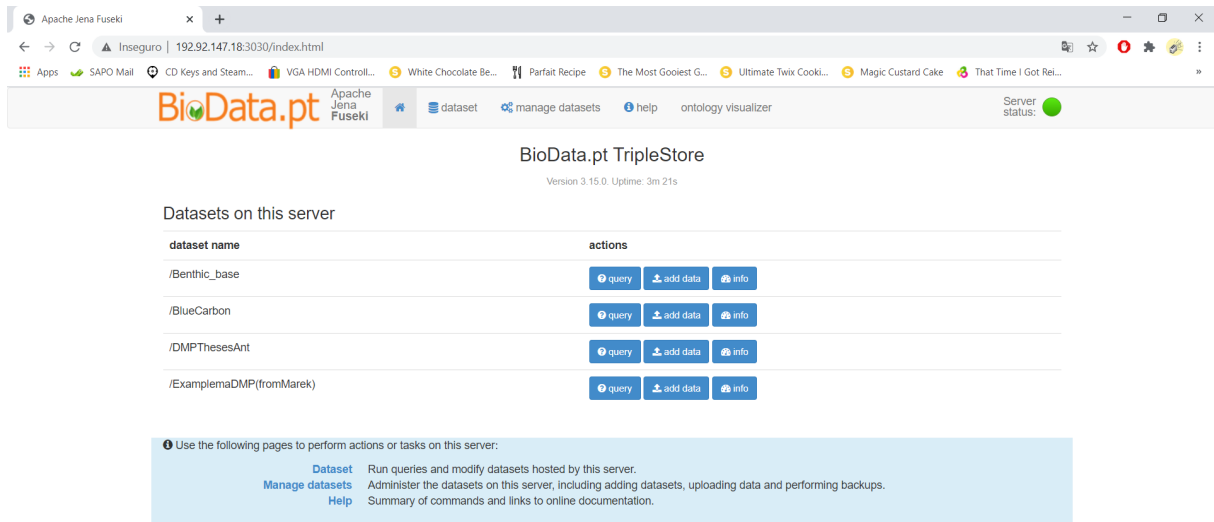


Figure 5.15: Fuseki Server after DSWEexport execution

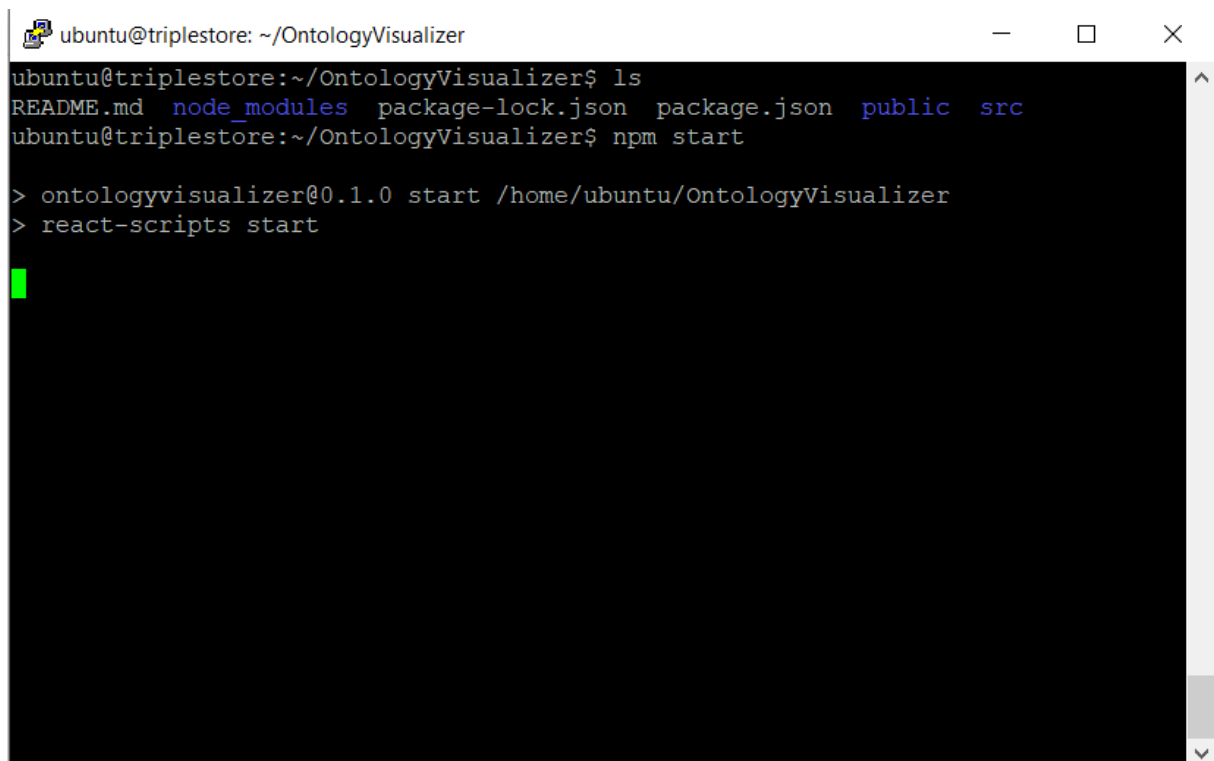


Figure 5.16: Start Ontology Visualizer

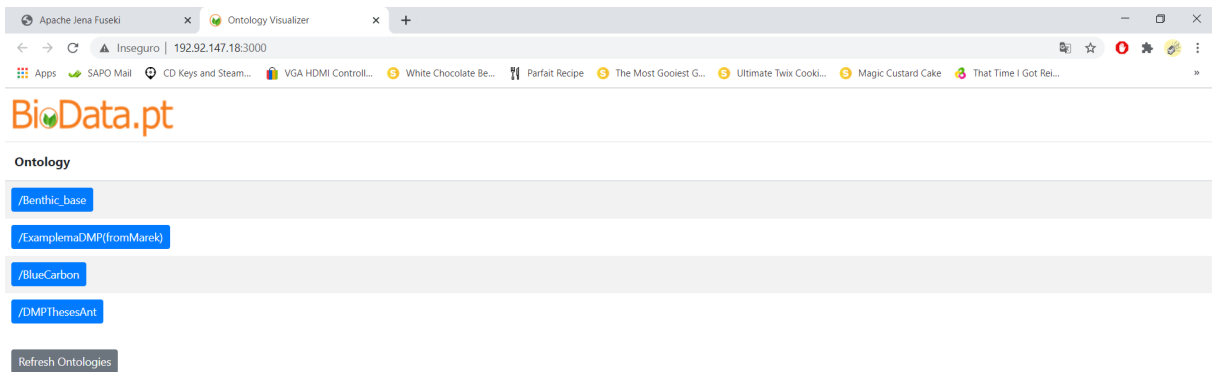


Figure 5.17: Ontology Visualizer

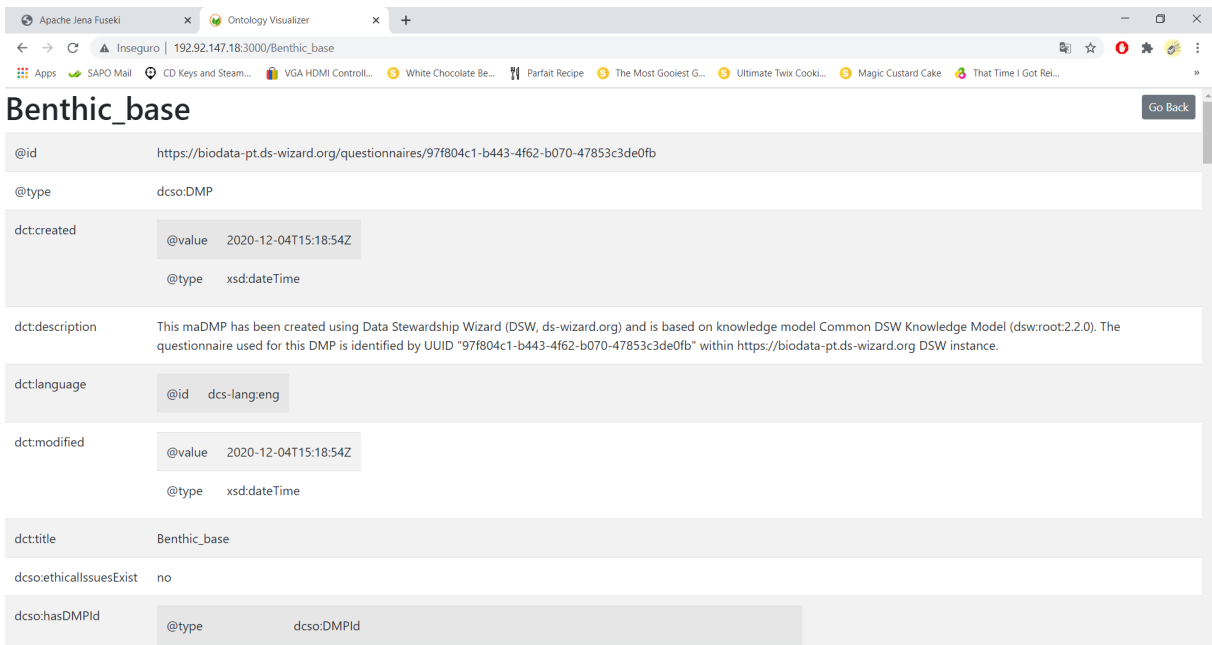


Figure 5.18: DMP example in Ontology Visualizer

6

Conclusion

Contents

6.1 Conclusions	43
6.2 System Limitations and Future Work	43

6.1 Conclusions

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 its preservation in a triplestore; (2) a creation of service to visualize maDMP;

To create the DMPs the BioData.pt DSW instance was used, both synthetic and real DMPs 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 DMPs 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.

6.2 System Limitations and Future Work

Given the time restrain the system have some limitations, such as the application DSWEExport 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 DSWEExport should export every file type supported 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.

Bibliography

- [1] A. Shoshani and D. Rotem, "Scientific data management. challenges, technology, and development," *Scientific Data Management: Challenges, Technology, and Deployment*, 01 2009.
- [2] A. Surkis and K. Read, "Research data management," *Journal of the Medical Library Association: JMLA*, vol. 103, no. 3, p. 154, 2015.
- [3] M. D. Wilkinson, M. Dumontier, I. J. Aalbersberg, G. Appleton, M. Axton, A. Baak, N. Blomberg, J.-W. Boiten, L. B. da Silva Santos, P. E. Bourne *et al.*, "The fair guiding principles for scientific data management and stewardship," *Scientific data*, vol. 3, 2016.
- [4] T. Miksa, J. Cardoso, and J. Borbinha, "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, 2018, pp. 2733–2742.
- [5] T. Miksa, A. Rauber, R. Ganguly, and P. Budroni, "Information integration for machine actionable data management plans," *International Journal of Digital Curation*, vol. 12, p. 22, 09 2017.
- [6] N. F. Noy, D. L. McGuinness *et al.*, "Ontology development 101: A guide to creating your first ontology," 2001.
- [7] J. R. G. Pulido, M. Garcia-Ruiz, R. Herrera, M. Cabello, S. Legrand, and D. Elliman, "Ontology languages for the semantic web: A never completely updated review," *Knowledge-Based Systems*, vol. 19, pp. 489–497, 11 2006.
- [8] D. L. McGuinness, F. Van Harmelen *et al.*, "Owl web ontology language overview," *W3C recommendation*, vol. 10, no. 10, p. 2004, 2004.
- [9] A. Whyte and J. Tedds, "Making the case for research data management," 2011.
- [10] OECD, "Making open science a reality," no. 25, 2015. [Online]. Available: <https://www.oecd-ilibrary.org/content/paper/5jrs2f963zs1-en>

- [11] M. Boeckhout, G. Zielhuis, and A. Bredenoord, "The fair guiding principles for data stewardship: fair enough?" *European Journal of Human Genetics*, vol. 26, pp. 931–936, 2018.
- [12] C. Strasser, "Research data management," *National Information Standards Organization*, 2015.
- [13] W. K. Michener, "Ten simple rules for creating a good data management plan," *PLoS Comput Biol*, vol. 11, 10/2015 2015.
- [14] T. Miksa, S. Simms, D. Mietchen, and S. Jones, "Ten simple rules for machine-actionable data management plans (preprint)," Feb. 2018. [Online]. Available: <https://doi.org/10.5281/zenodo.1172673>
- [15] R. Studer, V. Benjamins, and D. Fensel, "Knowledge engineering: Principles and methods," *Data & Knowledge Engineering*, vol. 25, no. 1, pp. 161 – 197, 1998. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0169023X97000566>
- [16] X. Su and L. Ilebrikke, "A comparative study of ontology languages and tools," in *International Conference on Advanced Information Systems Engineering*. Springer, 2002, pp. 761–765.
- [17] D. Kalibatiene and O. Vasilecas, "Survey on ontology languages," in *Perspectives in Business Informatics Research*, J. Grabis and M. Kirikova, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 124–141.
- [18] D. Beckett, T. Berners-Lee, E. Prud'hommeaux, and G. Carothers, "Rdf 1.1 turtle," *World Wide Web Consortium*, 2014.
- [19] J. Pérez, M. Arenas, and C. Gutierrez, "Semantics and complexity of sparql," *Lecture Notes in Computer Science The Semantic Web - ISWC 2006*, p. 30–43, 2006.
- [20] C. Buil-Aranda, A. Hogan, J. Umbrich, and P.-Y. Vandenbussche, "Sparql web-querying infrastructure: Ready for action?" in *The Semantic Web – ISWC 2013*, H. Alani, L. Kagal, A. Fokoue, P. Groth, C. Biemann, J. X. Parreira, L. Aroyo, N. Noy, C. Welty, and K. Janowicz, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 277–293.
- [21] D. Tomaszuk, "Document-oriented triplestore based on rdf/json," *Studies in Logic, Grammar and Rhetoric*, vol. 22, no. 35, 2010.
- [22] M. Horridge and S. Bechhofer, "The owl api: a java api for working with owl 2 ontologies," in *Proceedings of the 6th International Conference on OWL: Experiences and Directions-Volume 529*. CEUR-WS.org, 2009, pp. 49–58.

- [23] R. Hooft, M. Kuzak, M. Suchánek, and R. Pergl, ““Data Stewardship Wizard”: bringing together Researchers, Data Stewards, and Data Experts around Data Management Planning,” Apr. 2019. [Online]. Available: <https://doi.org/10.5281/zenodo.2649894>
- [24] J. B. Pereira-Leal, I. A. Abreu, C. S. Alabaça, M. H. Almeida, P. Almeida, T. Almeida, M. I. Amorim, S. Araújo, H. Azevedo, A. Badia *et al.*, “A comprehensive assessment of the transcriptome of cork oak (*quercus suber*) through est sequencing,” *BMC genomics*, vol. 15, no. 1, p. 371, 2014.
- [25] N. D. Mendes, P. T. Monteiro, C. Vaz, and I. Chaves, “Towards a plant experimental assay ontology,” *DILS 2014*, p. 41, 2014.
- [26] A. Francisco, C. Vaz, P. Monteiro, J. Melo-Cristino, M. Ramirez, and J. Carriço, “Phyloviz: Phylogenetic inference and data visualization for sequence based typing methods,” *BMC bioinformatics*, vol. 13, p. 87, 05 2012.
- [27] P. T. Monteiro, J. Oliveira, P. Pais, M. Antunes, M. Palma, M. Cavalheiro, M. Galocha, C. P. Godinho, L. C. Martins, N. Bourbon, M. N. Mota, R. A. Ribeiro, R. Viana, I. Sá-Correia, and M. C. Teixeira, “YEASTRACT+: a portal for cross-species comparative genomics of transcription regulation in yeasts,” *Nucleic Acids Research*, vol. 48, no. D1, pp. D642–D649, 10 2019. [Online]. Available: <https://doi.org/10.1093/nar/gkz859>

