Towards Findable, Accessible, Interoperable and Reusable (FAIR) Data Repositories: Improving a Data Repository to Behave as a FAIR Data Point

Repositórios para dados localizáveis, acessíveis, interoperáveis e reutilizáveis (FAIR): adaptando um repositório de dados para se comportar como um FAIR Data Point

João Luiz Rebelo Moreira[*]

Luiz Olavo Bonino[**]

Luís Ferreira Pires[***]

Marten van Sinderen[****]

Patricia Corrêa Henning[*****]

RESUMO

É necessário um esforço significativo para encontrar, entender e reutilizar dados da pesquisa. Para endereçar esse problema, os princípios de dados Localizáveis, Acessíveis, Reutilizáveis e Interoperáveis (FAIR em inglês) foram criados, e descrevem um conjunto mínimo de requisitos para gerenciamento e administração de dados, considerados a base tecnológica para a Nuvem Europeia de Ciência Aberta. O *FAIR Data Point* (FDP) utiliza dados ligados (LD) para expor dados e metadados aderentes aos princípios de dados FAIR, especificando um conjunto de metadados padronizados que um repositório de dados deve implementar. Os proprietários de dados

ABSTRACT

Significant effort is required to find, make sense and reuse research data. To tackle this problem, the Findable, Accessible, Reusable and Interoperable (FAIR) data principles describe a minimal set of requirements for data management and stewardship, considered as the technological basis for the European Open Science Cloud. The FAIR data point (FDP) leverages linked data (LD) to expose data and metadata adhering to the FAIR data principles, specifying a set of standardized metadata that a data repository should implement. Data owners can expose datasets, and data users can reuse datasets through RESTful services, enabling interoperability in a

[*] Doutor em Ciência da Computação pela Universidade de Twente (Holanda). Pesquisador de pós-doutorado em ciência da computação pelas universidades VU Amsterdam e Twente. Endereço: De Boelelaan 1105, 1081 HV Amsterdam. Telefone: 020 598 9898. E-mail: j.luizrebelomoreira@utwente.nl.

[**] Doutor em Ciência da Computação pela Universidade de Twente (Holanda). Diretor de tecnologia do escritório GO FAIR. Endereço: Rijnsburgerweg 10, 2333 AA Leiden. E-mail: luiz.bonino@go-fair.org.

[**] Doutor em Ciência da Computação pela Universidade de Twente (Holanda). Professor associado da Universidade de Twente. Endereço: P.O. Box 217, 7500 AE Enschede. E-mail: l.ferreirapires@utwente.nl.

[**] Doutor em Ciência da Computação pela Universidade de Twente (Holanda). Professor associado da Universidade de Twente e chefe do grupo SCS. Endereço: P.O. Box 217, 7500 AE Enschede. E-mail: m.j.vansinderen@utwente.nl.

[**] Doutora em Informação e Comunicação em Saúde pelo Instituto de Comunicação e Informação Científica e Tecnológica em Saúde (ICICT / FIOCRUZ). Professora Associada da Universidade Federal do Estado do Rio de Janeiro (UNIRIO). Endereço: Avenida Pasteur, 296, Urca - Brasil. E-mail: henningpatricia@gmail.com.

podem expor conjuntos de dados e os usuários de dados podem reutilizar conjuntos de dados por meio de serviços RESTful, permitindo a interoperabilidade em escala na web. Os repositórios de dados e o software subjacente apenas recentemente começaram a oferecer suporte à LD, e seus metadados estão disponíveis apenas como pares de valores-chave. Uma questão em aberto neste contexto é como permitir que um software de repositório de dados existente seja compatível com a especificação do FDP, ou seja, como adicionar descrições semânticas aos repositórios de dados para garantir a interoperabilidade semântica entre dados de diferentes repositórios. Este artigo descreve uma solução não invasiva e não intrusiva de proxy semântico que permite que um software de repositório de dados, o serviço EUDAT B2share, se comporte como um FDP, permitindo a interoperabilidade semântica por meio de traduções semânticas. A solução inclui uma metodologia para o mapeamento de metadados com base em transformações endógenas de modelos léxicos para modelos semânticos. Mostramos como os metadados nos pares de valores-chave de um repositório de uso geral podem ser compatíveis com a tecnologia LD sem alterar o software do repositório. A validação da solução inclui testes funcionais das camadas de metadados do FDP e uma análise de desempenho do impacto do proxy semântico na troca de dados. Os resultados mostram que o B2share pode ser compatível com as especificações do FDP, tendo impacto reduzido no desempenho da troca de dados. Portanto, a validação mostra que a solução é viável e adequada para transformar um software de repositório de dados de uso geral em um FDP.

Palavras-chave: Dados FAIR; Reusabilidade de Dados; Software de Repositório de Dados; *FAIR Data Point*.

web scale. Data repositories and their underlying software only recently started supporting LD, and their metadata are only available as key-value pairs. An open question in this context is how to enable an existing data repository software to be compliant with the FDP specification, i.e., how to add semantic descriptions to data repositories to ensure the semantic interoperability among data from different repositories? This paper describes a semantic proxy solution to enable a data repository software, the EUDAT B2share service to behave as an FDP in a non-invasive and non-intrusive way, enabling the semantic interoperability through semantic translations. Our solution describes a methodology for metadata mapping based on endogenous model-driven transformations from lexicon to semantic models. We show how metadata in key-value pairs from a general-purpose repository can be made compliant with LD technology without changing the repository software. The solution validation includes functional tests of the FDP metadata layers and a performance analysis of the impact of the semantic proxy on data exchange. The results show that B2share can be compliant to FDP specifications with a reduced impact on the data exchange performance. Therefore, the validation shows that the solution is feasible and adequate to transform a general-purpose data repository software in an FDP.

Keywords: FAIR Data; Data Reusability; Data Repository Software; FAIR Data Point.

INTRODUCTION

In the context of interoperability and aggregation of research data for knowledge reuse, an open topic is how to "facilitate the paradigm shift from document-based to

knowledge graph-based information exchange in science and technology" (Auer 2017). The basis for reusing data is an appropriate way to store and keep data safe over years and to access the data, i.e., by verifying the user's identity and granting authorization. For research data storage, the so-called data repositories play that role. Data repositories are usually designed to store and manage the outputs of research after the data are produced, playing an important role to allow the reusability of these data, in which interoperability is a first- class citizen. Data repositories ensure that data authorship is maintained, while the responsibility of keeping data safe and intact over years is transferred to the repository. Usually, the Dublin Core set of metadata is supported by the most popular data repository software and services, which is targeted at making data citable. For example, the EUDAT B2share is one of the most advanced data repository software (EUDAT 2018a), which can be deployed in local environments (e.g., universities) and provides data repository services through user-friendly, secure, robust and reliable web services (EUDAT 2018b). B2share enables community-driven metadata management, i.e., via domain tailored metadata, assigning Persistent Identifiers to ensure long-lasting access and references.

When research data are stored spread across different data repositories, reusing data demands a significant effort to find, make sense and really incorporating the data in the research. The Findable, Accessible, Reusable and Interoperable (FAIR) data principles describe a minimal set of requirements for data management to deal with the reusability problem, being adopted by several funding programs (e.g., H2020), and is considered as the technological foundation for the Open Science movement. The FAIR data point (FDP) (DTL 2017) is a specification for the structure of a data repository software and its minimum metadata required to adhere to the FAIR data principles. The FDP is leveraged by semantic technologies. Data owners can expose datasets and data users can reuse datasets through the RESTful architecture of FDP, enabling interoperability of data repositories in a web scale. Linked data allows scientists to combine public metadata expressed as linked data triples in an ad-hoc manner without the need to first harvest metadata entries and store them locally.

Exposing new and existing datasets following the FDP specifications facilitates the interpretation and combination of heterogeneous and challenging types of research data. In this context, an open question is how to enable an existing data repository software to behave as an FDP. The aim of the EUDAT-based FAIR pilot project was to address this problem, demonstrating how to make B2share compliant to the FDP specification, i.e., how to enable B2share to capture metadata as an FDP. In this paper, we introduce the approach used in this project, which involves a semantic proxy that translates data retrieved from B2share, through a RESTful API, to linked data compliant to the FDP specification. Our approach enables a non-invasive and non-intrusive solution, which includes a methodology to map the metadata from the data repository software to the metadata levels of FDP based on the research on semantic translations implemented as model-driven engineering transformations.

The paper is structured as follows: Section 2 introduces some background on data reusability, discussing the role of data repositories for data reuse, the EUDAT B2share data repository software and the FDP specification, and defines the problem addressed in this paper. Section 3 presents our solution, describing the architecture, methodology and implementation of the semantic proxy. Section 4 presents the validation of our solution, and discusses the validations results. Finally, Section 5 concludes this paper by discussing our contributions, lessons learned and future work.

REUSABILITY OF RESEARCH DATA

Data repositories are usually designed to store and manage the outputs of research after the data are produced and the results are published. Therefore, they play a major role on allowing the reusability of research data, for which interoperability is required.

Role of data repositories

The most popular protocol for metadata harvesting is the Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH), which is based on the Dublin Core and sets a basis for research data interoperability, while the Open Archival Information System (ISO OAIS) is a well-known organization that promotes metadata interchange and dissemination. In a comparison study in which OAIS experts participated (Amorim et al. 2017), several data repositories were inventoried and six were selected to be validated according to OAI-PMH: EUDAT B2share, CKAN, Zenodo, Dspace, Eprints and Figshare. Only CKAN is not natively compliant to OAI-PMH, but all of them provide APIs for exposing metadata records. This study concluded that selecting a data repository software is a challenging task and should be based on technical choices. EUDAT, CKAN and Dspace presented open-source licenses that allow them to be customized and deployed in local servers instead of relying on external storage. In particular, EUDAT services provide a way to integrate with CKAN and Dspace.

Some of these data repositories have support for linked data to improve data interoperability. For example, RDF for CKAN is provided via plugins that allow it to expose and consume metadata according to the Data Catalog Vocabulary (DCAT) (CKAN 2018). There are similar solutions for Dspace and Eprints. EUDAT implemented a linked data service pilot for data semantic annotation with the support of the EUDAT semantics workgroup, where the annotation module was integrated to B2share service and linked to domain-specific vocabularies according to a domain-specific template.

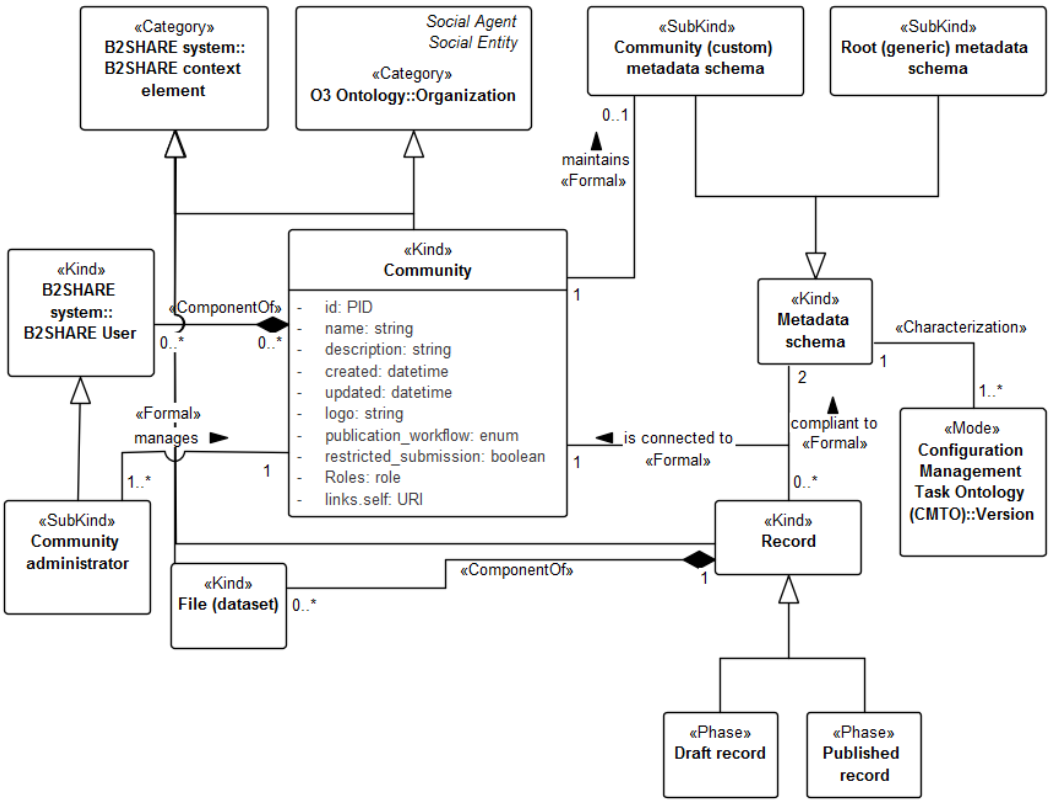
EUDAT B2share service

EUDAT is a European initiative that devises technology and services to support the management of scientific data throughout the whole research process and beyond. It provides services for replicating (B2safe), publishing (B2share), sharing and versioning (B2drop) and searching (B2find) data. Furthermore, there are services for ad-hoc data transfers (B2stage), authentication and authorization (B2access) and for labeling data with persistent identifiers (B2handle).

B2share is the data repository service of EUDAT and provides the functionality to archive and publish the so-called long-tail data. Data are uploaded in deposits (records), which can be mapped to the concept of a dataset. A deposit is always annotated with some metadata, which are collected from the user at the time of upload to B2share. In B2share, a central element is the (scientific) community, which has the roles of creating and maintaining metadata schemas and curating the community datasets, enabling flexibility for domain-specific metadata. An end user of B2share, typically a researcher, can be part of one or more communities. A community can maintain metadata schemas but is always compliant to a generic metadata schema that is configured by the system administrator, having as default

the EUDAT common metadata schema based on a superset of the Dublin Core elements. A community administrator has permission grants needed to change the metadata schema definitions and record curation tasks. A record in B2share is a "container" that adheres to both generic and the community metadata schemas, and to which any user can upload data files. A record submitted within a community must conform to the schemas in order to be accepted, i.e., published with an assigned Persistent Identifiers (PID), so that it becomes publicly accessible and cannot be changed. The other possible state of a record is "draft", i.e., immediately after its creation it is only accessible by its owner and can be freely modified (metadata can be changed and files can be uploaded).

Figure 1: Ontology-driven conceptual model of the B2share RESTful API



The community metadata schema is defined through a JavaScript Object Notation (JSON) schema and can be retrieved through the B2share Representational state transfer application program interface (RESTful API), which is a common syntactic approach that enables the seamless integration among software systems in the *Web* through the HTTP protocol (EUDAT 2018d). A REST architecture serializing data as JSON is one of the most common architecture for an API. We performed an ontological analysis to understand the B2share RESTful API, with the support of the Unified Foundational Ontology (UFO) and its OntoUML language (Guizzardi et al. 2015). Figure 1 illustrates the OntoUML conceptual model we defined as a result of our analysis, which helped us communicate with B2share experts and understand how the data are stored in the system.

FAIR Data Point

The Findable, Accessible, Interoperable, and Reusable (FAIR) data principles have rapidly been adopted by several institutions worldwide (Wilkinson, Verborgh, et al. 2017), leveraged by the GO FAIR initiative (GO-FAIR 2018). GO FAIR aims to implement a bottom-up approach for the implementation of the European Open Science Cloud (EOSC), which represents a vision to strengthen EU's competitiveness in digital technologies, enabling users to benefit from data-driven science. The FAIR principles are organized in four categories (FAIR), where each one represents a requirement based on best practices from data management and stewardship, applied to both data and metadata. The FAIR level of a data repository is termed FAIRness, which can be measured according to specific metrics (Wilkinson, Sansone, et al. 2017).

GO FAIR provides a set of technology specifications and implementations called Data FAIRport. Within the Data FAIRport infrastructure, the FAIR Data Point (FDP) component specifies how to access data and metadata. The FDP provides metadata at five complementary layers: FDP Metadata (level 1), Data Catalog Metadata (level 2), Dataset Metadata (level 3), Distribution Metadata (level 4) and Data Record Metadata (level 5). The FDP implementation is optimized for machine discoverability and processing. (Meta)data access starts with the FDP URL, which resolves to metadata that describes the FDP and supports further exploration of the FDP by iterating through the different metadata levels . For example, the metadata include a link to the next FDP level (catalogs), which in turn links to the Dataset-level descriptions. The FDP metadata specification is based on existing standards and vocabularies, such as OAI-PMH, Dublin Core, DCAT and LDP. Therefore, the FDP specification is not "yet another standard", but rather aims to combine existing standards to meet the FAIR objectives. The FDP specification also gives a recommendation for the minimal metadata needed towards dataset findability (F) and it specifies (how) to declare a data license in order to be reusable (R).

Finally, the dataset metadata contain links to one or multiple distribution and data record metadata. However, those links are maintained by the data owners and not by the FDP itself. Hence, data access and immutability cannot be guaranteed as it is guaranteed in a repository. These metadata need to be stored, and once this happens they have to be exposed also according to the FDP specification.

Problem definition

Exposing new and existing datasets following the FDP specifications facilitates the interpretation and combination of heterogeneous and challenging types of research data. The goal of the EUDAT-based FAIR project for data interoperability (EUDAT 2018c) was to implement and deploy a FDP using a combination of existing semantic web standards and the EUDAT service for data repository (B2share). Therefore, the design problem addressed in this paper is how to enable an existing data repository software to behave as an FDP, more specifically, how to make B2share compliant to the FDP specification, i.e., how to enable B2share to capture metadata as an FDP. In this context, three requirements were established during the project inception:

(R1). The time frames of B2share testing and this work overlapped, which constrained changes in the B2share code. Moreover, since B2share provides a way to access data (a RESTful API), a requirement emerged to develop a non-invasive and non-intrusive solution, i.e., a solution that is totally decoupled of B2share. This is an important

constraint since our solution could not directly access the database where the data were stored and obliged the solution to deal only with the API interfaces.

(R2). Since the FDP relies on metadata implemented as RDF from existing standards, and enabling semantic interoperability of B2share data repository software is a requirement, B2share RESTful API needed to be empowered by providing semantic annotated data.

(R3). FDP specifies metadata that a data repository software needs to comply and some of these metadata are supported by B2share. So, we had to identify mappings between B2share and FDP to align B2share terminology according to the FDP metadata layers. The FDP level 5 (data record) is out of the scope of this work since the data record metadata are domain-specific and, therefore, a solution cannot be generally applicable for this level, requiring a domain-driven approach.

REUSABILITY OF RESEARCH DATA

Our solution to address the three requirements is inspired on our research on model-driven engineering of ontology alignments implemented as semantic translations (Moreira et al. 2017). Semantic translation is the "process of changing the underlying semantics of a piece of knowledge. Given some information described semantically, in terms of a source ontology, it is transformed into information described in terms of a target ontology" (Ganzha et al. 2017).

Solution overview

Our solution implements semantic translations from B2share data representations to the FDP metadata. Here we enlarge the scope of the definition of "semantic translation" because B2share metamodel is not described as an ontology, e.g. implemented as RDF, but its model can be considered as a "piece of knowledge". This approach addresses requirement R3 because the semantic translations developed were derived from the mappings between the FDP metadata layers and the B2share data model. The high-level mappings are described in Table 1.

Table 1 – FAIR data point levels and their equivalences to B2share system

FDP level	**B2share**	**Description**
FAIR Data Point (L1)	B2share system	A FDP is a data repository specification based on re3data.org, which defines B2share as a data repository software (re3data.org 2018).
Group/ catalog (L2)	Community	A catalog (FDP) is a way to categorize, i.e., to create a taxonomy of, datasets. The basic categories (facets) of B2share are the Communities.
Dataset (L3)	Record	A dataset (FDP) is comprised of parts of data that can be manipulated as a unit, which roughly corresponds to a record in B2share.
Distribution (L4)	File	A distribution (FDP) is a resource that gives access to the data, similarly to files.

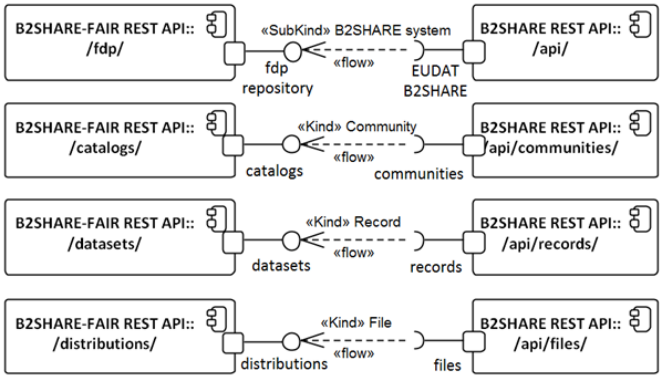
In software engineering, a common approach to implement data transformations based on mappings is the proxy pattern. A proxy is a programming design pattern in which a software can act as a wrapper that is called by the client to access the real intended service. It is used to simply forward the original object from the service to the client or can provide additional logic to the object (Fowler 2002). Therefore, our solution adopted this pattern to access data from B2share, execute the mappings and deliver the data to the client according to the FDP specifications. To access the B2share data we used a RESTful API, which is the common approach to address requirement R1, i.e., access data from an information system in a non-invasive and non-intrusive way.

To address requirement R2 we improved the proxy with semantic technologies, turning it into a semantic proxy that gets data from B2share and delivers the data to the clients as RDF triples. The solution applies horizontal and endogenous model-driven transformations, mapping the B2share lexicon model, which is serialized as key-value pairs in JSON to an RDF model that is delivered as JSON-LD (a way to serialize RDF (W3C 2018)). The FDP metadata specification already prescribes the use of RDF, making this solution appropriate. Therefore, the B2share-FAIR solution is a proxy that implements semantic translations from data collected from the B2share RESTful API (source) at runtime on each client call to FDP compliant data. The proxy is the implementation of the B2share-FAIR REST/JSON-LD API.

Behavioral model: API communication

The B2share-FAIR REST/JSON-LD API was structured according to (1) the FDP metadata levels; and (2) the B2share RESTful API resources available, i.e., the services associated to their respective endpoints. Figure 2 illustrates the proxy endpoints (on the left), where each endpoint issues a synchronous GET request to the B2share endpoint according to the high-level mappings (Table 1). For example, when a client requests information about the B2share FDP data repository (*/fdp/* endpoint) , the proxy forwards the request to the equivalent B2share RESTful API (*/api/* endpoint), receives the data, executes the required transformations and responds the transformed data to the client.

Figure 2: B2share-FAIR RESTful API as a proxy of B2share REST endpoints



A requirement to implement each proxy endpoint is to map the metadata fields from B2share model onto the FDP specification at design time, which is a challenging task. For example, how to know that the Community "name" metadata should be mapped to the Catalog *dct:title* or to *dct:publisher*? To support the design of these mappings we applied our MDE methodology for the development of semantic translations

(Moreira et al. 2017). Our methodology follows a common software engineering approach with specification and implementation phases during the design time of the ontology alignment. The specification describes in natural language the possible mappings and the involved rules, linking the original ontology to the generated ontology.

This methodology is leveraged by the practice of ontological analysis with the Unified Foundational Ontology (UFO), providing a well-grounded set of categories that improve the understanding of the analyzed domain and, ultimately, improves the semantic interoperability of a model (Guizzardi et al. 2015). For example, when analyzing the identity of dct:title metadata as the name property of a catalog, it is clear that this metadata should be mapped to the Community "name" property. Therefore, for each FDP level, the procedure underlying the methodology adopted in our solution is:

(1) Pre-analysis of all B2share metadata and all metadata of the equivalent FDP level.

(2) For each metadata field of the FDP level (ordering by the required ones):

(a) Understand the FDP metadata description.

(b) Search the similar concept in B2share and create the mapping. Here the mapping can be either straightforward, as the case of *B2share.Community.name* with *FDP.Catalog.dct:title*; or multiple, which may require the definition of formation rules; or the mapping does not exist. For the second and third cases it is required to annotate either the rule or the non-existing term.

(3) Check if there are required FDP metadata that could not be mapped, which represents a crucial gap of the data source. This is the worst-case scenario, which makes it necessary to either fix a value for the metadata or to change the data source, which violates requirement R1.

(4) Check each metadata field of the B2share entity that was not mapped:

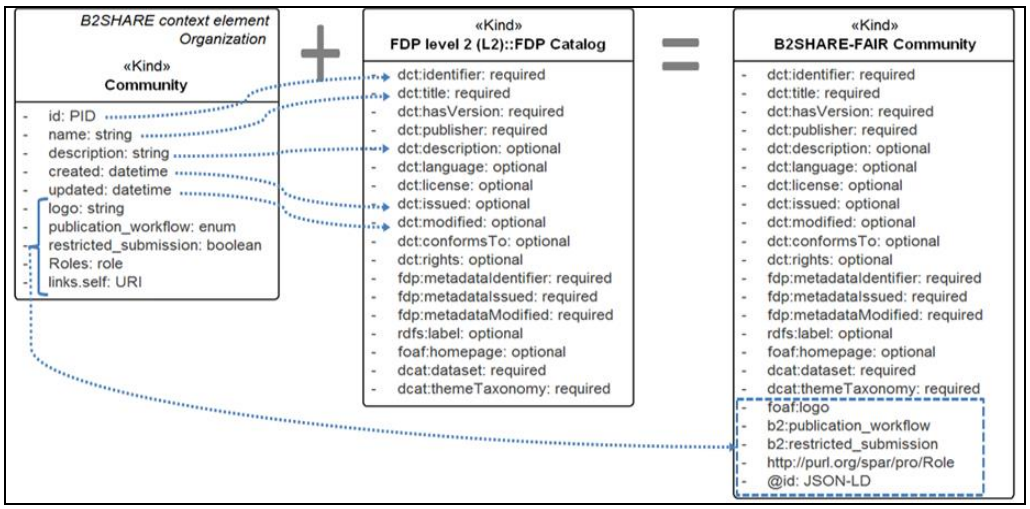
(a) Search the similar concept in common and, preferably, standardized ontologies.

(i) If it is found, then create the mapping.

(ii) If it is not found, i.e., the concept is not present in other ontologies (e.g., it is a domain-specific concept), it is required to represent this concept in an ontology. An important issue here is the need of only creating the elements in the ontology that are not found in the FDP specification and neither in other ontologies.

Figure 3 illustrates the mappings created between Community (B2share) and Catalog (FDP level 2) according to this methodology. From the "id" to the "update" property of Community (on the left) the mappings were straightforward. Some of the B2share metadata could not be mapped to FDP (step 4), so for "logo" and "Roles" metadata, existing ontologies were reused, as FOAF and the Publishing Role Ontology (PRO), respectively. However, the specific metadata of B2share (*publication_workflow* and *restricted_submission*) required to be represented in an ontology. Thus, we created the B2share ontology, where *publication_workflow* is an object property and *restricted_submission* is a data property of Community. Fortunately, we found no required FDP metadata that could not be mapped onto B2share metadata (worst scenario in step 3). Our mappings were specified at design time for all four levels and discussed (harmonized) with the B2share and FDP experts.

Figure 3: Mappings from B2share Community to Catalog (FDP metadata level 2)



Implementation

All components of the solution, including the B2share ontology, the proxy code and test data are available in (Moreira 2018). The solution was implemented in Python 3, a non-functional requirement, with support of PyLD (Digital Bazaar 2017), the JSON-LD library for RDF data serialization. Along with jsonld.js (the JavaScript library), PyLD is one of the most popular JSON-LD library implementations, which has a strong support from the community.

Figure 4: Translation method for Catalog (FDP metadata level 2) from Community

```
# B2SHARE: Community
# Level 2: Catalog metadata layer
def translate_catalog(community):

    AdminRole = {
    }
    MemberRole = {
    }

    context = {
        # ontologies used in FDP according to spec
        "rdf" : "http://www.w3.org/1999/02/22-rdf-syntax-ns#",
        "rdfs" : "http://www.w3.org/2000/01/rdf-schema#",
        "dcat" : "http://www.w3.org/ns/dcat#",
        "xsd" : "http://www.w3.org/2001/XMLSchema#",
        "owl" : "http://www.w3.org/2002/07/owl#",
        "dct" : "http://purl.org/dc/terms/",
        "lang" : "http://id.loc.gov/vocabulary/iso639-1/",
        "fdp" : "http://rdf.biosemantics.org/ontologies/fdp-o#",
        "foaf" : "http://xmlns.com/foaf/",
        # B2SHARE otology (internal terms)
        "b2" : "https://b2share.eudat.eu/ontology/b2share/",
        # Other ontologies (reused)
        "pro" : "http://purl.org/spar/pro/" # Publishing Roles
    }

    doc = {
        "@id": community.links.selflink,
        "@type": "dcat:Catalog",
        "http://purl.org/dc/terms/identifier": community.identifier,
        "http://purl.org/dc/terms/title": community.name,
        "http://purl.org/dc/terms/description": community.description,
        "http://purl.org/dc/terms/issued": community.created,
        "http://purl.org/dc/terms/modified": community.updated,
        "http://xmlns.com/foaf/logo" : community.logo,
        "https://b2share.eudat.eu/ontology/b2share/publication_workflow" : community.publication_workflow,
        "https://b2share.eudat.eu/ontology/b2share/restricted_submission" : community.restricted_submission,
        "https://b2share.eudat.eu/ontology/b2share/AdminRole" : AdminRole,
        "https://b2share.eudat.eu/ontology/b2share/MemberRole" : MemberRole
    }
    compacted = jsonld.compact(doc, context)
    return compacted
```

The core part of the implementation is the semantic translations module (*translators.py*) with the methods for each translation. For example, Figure 4 illustrates the translation method from a Community to a Catalog, showing all ontologies from the FDP specification in the context element, as well as B2share ontology and PRO, used to deal with the role-based access control of B2share. In addition, the properties are instantiated according to the mappings from the community element.

VALIDATION AND DISCUSSION

To validate our mappings we assessed their soundness by exhaustively revisiting them with specialists of B2share and FDP. The RESTful API functional validation was performed on the B2share test environment[1] executing the translations for each level through client calls on the endpoints of B2share-FAIR RESTful API. The first level (data repository) translated the information from the B2share data repository software (the test instance) to the FDP level 1, illustrated in Figure 5(a). For the second level (catalog), each community registered in B2share was translated, either through the list given by /catalogs/ or by requesting each */catalogs/[ComunityID]*. Figure 5(b) shows the result of the translation of Aalto University community as a catalog. The validation of FDP level 3 (dataset) considered both the direct access to a B2share register through */dataset/[RegisterID]* and the B2share dataset search capability, by forwarding the client query string. Finally, FDP level 4 (distribution) was validated through a pre-selection of a set of existing files from B2share. For each translation method, an equivalent assertion method was developed to verify whether a mapping rule has been broken after executing.

Another important aspect of the validation has been to measure the impact on the performance of the service calls, since a performance overhead is expected when a proxy solution is applied. The total time transaction, i.e., total time taken from the client request to the response, was measured in two situations: (1) without the solution, to serve as a baseline for comparison; and (2) with the solution. Each FDP level was tested, and scripts were created to compute the total time transaction for each situation (1 and 2), comparing them in each execution. To statistically measure this variable and avoid noise caused by network fluctuations, a number of synchronous calls (10) were executed 10, 50, 100 times and an anomaly detection algorithm removed the noisy measurements (above the mean plus standard deviation). FDP level 1 has one test case, level 2 has two test cases, level 3 has three test cases and level 4 one test case. Level 1 has one test case because here is only one way to request the /api/ endpoint. Level 2 has two test cases because there are two ways to use the /communities/ endpoint, either listing all communities or by a community identifier. Level 3 has three test cases because /record/ can be used either by accessing a list of records by community identifier (all records of a community) or by accessing a list of records as the result of a search (forwarding the querystring) or by accessing one specific record (through the record identifier). Finally, level 4 has only one way to access the list of files of a record: by the record identifier. In each test case we varied the input data, such as e.g., the IDs used and querystrings for searching datasets. All scripts are open and available[2].

[1] https://trng-B2share.eudat.eu/api/

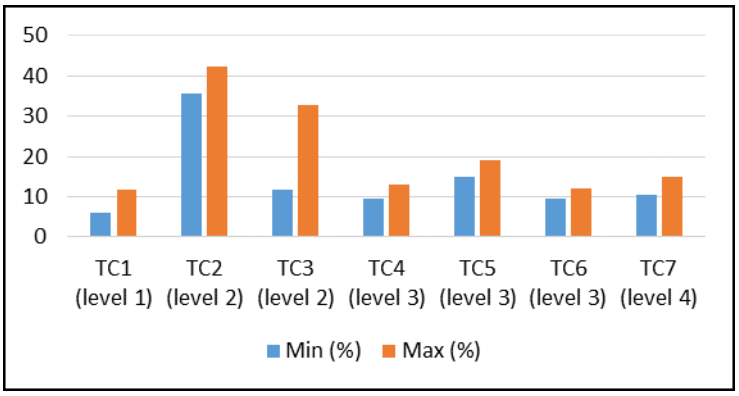
[2] https://github.com/jonimoreira/B2SHARE-FAIR/tree/master/src/proxy/tests

Figure 5: (a) B2share service translated to an FDP data repository. (b) B2share community (Aalto) translated to an FDP catalog

The environment where the proxy was deployed for the validation is a local virtual machine running Ubuntu 16.04 with 4GB RAM, 1 CPU, 16-bit, accessing the B2share test environment. The results of the execution of each test case are illustrated in Figure 6, which presents the B2share-FAIR minimum and maximum overhead difference in terms of percentage of the total time transaction between the two situations (with and without the solution). For example, TC1 refers to tests with FDP level 1, having the minimum difference computed as 6.13%, i.e., on average B2share-FAIR is 6.13% slower than directly accessing the B2share endpoint. TC1, TC4, TC5, TC6 and TC7 presented acceptable differences, impacting less than 20% (TC5 maximum difference). However, TC2 and TC3 presented a huge difference, with the worst-case of almost 43% slower (TC2 maximum difference). This difference comes from the programming approach for accessing indexed Python object arrays mapped from the JSON object (input). In contrast with the other levels in which objects were retrieved

by iterating arrays, in level 2 we used our own index-based solution (e.g., community.roles["MemberRole"].name), which presented poor performance. Besides this issue, a threat to this validation, which probably affected the results negatively, is that both the proxy and the tests scripts were deployed in the same test environment and executed together at the same time. Therefore, the differences are expected to be smaller in a distributed production environment and, thus, we argue that our solution is appropriate for non-critical performance constraints. A possible solution to address critical performance constraints is to implement caching mechanisms in the proxy server.

Figure 6: Performance validation (overhead) of the semantic proxy



The validation showed the appropriateness and usefulness of our solution, demonstrating how to make B2share compliant to the FDP specification, enabling B2share to capture metadata as an FDP. Requirement R1 was addressed and no changes in B2share code and/or (meta)data insertion were necessary. The RESTful APIs accomplished their role of enabling the integration of B2share with the semantic proxy in a decoupled manner. Requirement R2 was addressed by the solution through the semantic translations approach. The solution demonstrated how data stored in B2share could be annotated with metadata from ontologies. Furthermore, the implementation of the RESTful API providing data as JSON-LD syntax could enable the serialization of these ontologies. Requirement R3 was addressed by the identification of the mappings between B2share and FDP metadata levels and their implementation as semantic translations, validated both by the functional and performance tests. The solution validation showed that the automatic translations could produce the desired results in terms of compliance with the FDP specification.

CONCLUSIONS

In this paper, we introduced an approach to facilitate the shift from document-based to linked data exchange in research. Our methodology enabled an existing document-based data repository software, the EUDAT B2share, to behave as an FDP through semantic mappings. Our solution follows the proxy pattern, a wrapper, and implements semantic translations, transforming research data retrieved from B2share (as JSON) to FDP compliant RDF (as JSON-LD). This approach addressed the requirement of a decoupled solution from B2share by using its RESTful API to access the data, i.e., no changes in B2share code or data insertion were necessary. In addition, we demonstrated how data stored in B2share can be exposed in RDF triples, enabling semantic interoperability of the data repository software. The performance validation of the semantic proxy showed that it has a small impact on the total

transaction time of client requests when compared to direct access to B2share. Therefore, our solution shoed to be a feasible approach to facilitate an existing data repository software to expose the data stored as linked data.

Important lessons learned include the need of a strict methodology to align the metadata used in the data repository to the FDP metadata layers. This is a challenging task, especially in B2share, which enables the community and system administrators to add new metadata "on the fly" (at runtime). Moreover, we observed that the practice of ontological analysis adds a significant value when analyzing how a data repository software is structured. A similar approach could be used as a standard for the FDP level 5 (data record), by applying ontological analysis over existing domain-specific metadata and ontology-driven conceptual modelling for the change and development of ontologies. Recurrent questions arise in this context: how to find and assess existing ontologies regarding a given domain? When more than one ontology is available, how to measure the best ontology? Although there are several researches in this topic, there is a gap on bridging the exist research with the data FAIRport.

In our solution, MDE played an important role to implement the semantic translations as endogenous model-driven transformations. The serialization of RDF as JSON-LD and its support with programming libraries seem to be an adequate implementation decision when dealing with input data serialized as JSON. Future work includes more fine-grained tests with health data generated in our research on cardiac early warning systems (Moreira et al. 2018), giving emphasis to the accessibility aspect of private and sensitive data according to data management plans compliant to the EU legislation (95/46/EC and 2016/679 directives). Furthermore, we plan to experiment the RDF Mapping Language (RML)[3] for the implementation level of our approach. We also intend to apply our approach in data repositories that resemble B2share (functionally and technologically), such as Dataverse, CKAN and Zenodo. Finally, we plan to adapt our methodology to use the FAIR metrics to measure the FAIRness of the data repository software.

Artigo recebido em 09/07/2019 e aprovado em 15/10/2019

REFERÊNCIAS

AMORIM, R. C.; CASTRO, J. A.; SILVA, J. R.; RIBEIRO, C. A comparison of research data management platforms: architecture, flexible metadata and interoperability. *Universal Access in the Information Society,* n.16, p. 851-862, 2017.

AUER, S. Interoperability and aggregation of research data. *In:* SEMANTICS CONFERENCE, 2017. *Proceedings* [...]. 2017.

CKAN. *Linked data and RDF features for CKAN.* 2017. Disponível em: http://docs.ckan.org/en/latest/maintaining/linked-data-and-rdf.html. Acesso em: 05 maio 2018.

DIGITAL BAZAAR. *Python library for JSON-LD.* 2017. Disponível em: https://github.com/digitalbazaar/pyld. Acesso em: 05 maio 2018.

[3] http://rml.io/

DTL. *FAIR data point specification.* 2017. Disponível em: https://github.com/DTL-FAIRData/FAIRDataPoint/wiki/FAIR-Data-Point-Specification. Acesso em: 05 maio 2018.

EUDAT. *B2share:* Github page. 2018a. Disponível em: https://github.com/EUDAT-B2SHARE/b2share. Acesso em: 05 maio 2018.

EUDAT. *B2share:* store and publish your research data. 2018b. Disponível em: https://b2share.eudat.eu/. Acesso em: 05 maio 2018.

EUDAT. *An EUDAT-based FAIR data approach for data interoperability.* 2018c. Disponível em: https://eudat.eu/communities/an-eudat-based-fair-data-approach-for-data-interoperability. Acesso em: 05 maio 2018.

EUDAT. *EUDAT B2share REST API.* 2018d. Disponível em: https://B2share.eudat.eu/help/api. Acesso em: 05 maio 2018.

FOWLER, M. Patterns of enterprise application architecture. [S.l.]: Book, 2002.

GANZHA, M.; PAPRZYCKI, M.; PAWŁOWSKI, W.; SZMEJA, P.; WASIELEWSKA, K. Streaming semantic translations. *In:* INTERNATIONAL CONFERENCE ON SYSTEM THEORY, CONTROL AND COMPUTING (ICSTCC), 21., 2017. *Proceedings* [...]. 2017.

GO-FAIR. *GO FAIR:* a bottom-up international approach. 2018. Disponível em: https://www.go-fair.org/. Acesso em: 05 maio 2018.

GUIZZARDI, G.; WAGNER, G.; ALMEIDA, J. P. A.; GUIZZARDI, R. S. S. Towards ontological foundations for conceptual modeling: the unified foundational ontology (UFO) story. *Applied Ontology*, n. 10, p. 259-71, 2015.

MOREIRA, J. L. R. *Enabling B2SHARE to behave as a FAIR data point:* a proof-of-concept. 2017. Disponível em: https://github.com/jonimoreira/B2SHARE-FAIR. Acesso em: 05 maio 2018.

MOREIRA, J. L. R.; DANIELE, L. M.; FERREIRA PIRES, L. F.; SINDEREN, M. V.; WASIELEWSKA, K.; SZMEJA, P.; PAWLOWSKI, W.; GANZHA, M.; PAPRZYCKI, M. Towards IoT platforms' integration: semantic translations between W3C SSN and ETSI SAREF. *In:* SEMANTICS WORKSHOP SIS-IOT, 2017. *Proceedings* [...]. 2017.

MOREIRA, J. L. R.; FERREIRA, L. F. P.; SINDEREN, M. V.; WIERINGA, R.; SINGH, P.; COSTA, P. D.; LLOP, M. Improving the semantic interoperability of IoT early warning systems: the Port of Valencia use case. *In:* ENTERPRISE INTEROPERABILITY IX: I-ESA 2018. *Proceedings* [...]. [*S.l.*]: Springer, 2018.

RE3DATA.ORG. *B2share data repository entry in re3data.org.* 2018. Disponível em: https://www.re3data.org/repository/r3d100011394. Acesso em: 05 maio 2018.

W3C. JSON for Linked Data (JSON-LD) latest version (1.1). 2018.

WILKINSON, M. D.; SANSONE, S. A.; SCHULTES, E.; DOORN, P.; BONINO, L. O.; DUMONTIER, M. A design framework and exemplar metrics for FAIRness. *BioRxiv, 2017.*

WILKINSON, M. D.; VERBORGH, R.; BONINO, L. O.; CLARK, T.; SWERTZ, M. A.; KELPIN, F. D. L.; GRAY, A. J. G.; SCHULTES, E. A.; VAN MULLIGEN, E. M.; CICCARESE, P.; KUZNIAR, A.; GAVAI, A.; THOMPSON, M.; KALIYAPERUMAL, R.; BOLLEMAN, J. T.; DUMONTIER, M. Interoperability and FAIRness through a novel combination of web technologies. *PeerJ Computer Science*, n. 3, p. e110, 2017.