Semantic Knowledge Graphs for Distributed Data Spaces: The Public Procurement Pilot Experience

Cecile Guasch[1], Giorgia Lodi[2(✉)], and Sander Van Dooren[1]

[1] European Commission, DG DIGIT, Brussels, Belgium
{cecile.guasch,Sander.VAN-DOOREN}@ext.ec.europa.eu
[2] Institute of Cognitive Sciences and Technologies of the Italian National Research Council (ISTC-CNR), Rome, Italy
giorgia.lodi@cnr.it

Abstract. This paper presents the experience gained in the context of a European pilot project funded by the ISA2 programme. It aims at constructing a semantic knowledge graph that establishes a distributed data space for public procurement. We describe the results obtained, the follow up actions and the main lessons learnt from the construction of the knowledge graph. This latter requires to support different data governance scenarios: some partners control, with their own tools, the building process of their portion of the knowledge graph. Other partners participate in the pilot by providing only their open CSV/XML/JSON datasets, in which case transformations are required. These are performed on the infrastructure made available by the European Big Data Test Infrastructure (BDTI). The paper introduces the design and implementation of the knowledge graph construction process within such a BDTI infrastructure. By instantiating an OWL ontology created for this purpose, we are able to provide a declarative description of the whole workflow required to transform input data into RDF output data, which form the knowledge graph. The declarative description is therefore used to provide instructions to a workflow engine we use (Apache Airflow) for knowledge graph construction purposes.

Keywords: Knowledge graph · Data space · Linked (Open) data · Data transformation

1 Introduction

The importance of Public Procurement in the economy of the EU is well documented. Over 250.000 public authorities in the EU spend around 14% of GDP,

Supported by (formerly) ISA2 programme. We thank all the European partners that contributed to this work: AGID, ANAC, Consip, IMPIC, DFO and DG DIGIT.

U. Sattler et al. (Eds.): ISWC 2022, LNCS 13489, pp. 753–769, 2022.
https://doi.org/10.1007/978-3-031-19433-7_43

around 2 trillion euros per year[1]. Therefore, it is important to make the best use of the data it generates. Traditionally, public procurement has been mainly document-based. However, with the increasing use of digital technologies and digital negotiation instruments, public procurement has faced a variety of new interoperability challenges. These are related to insufficient sharing and re-use of data, overall lack of quality for the available data, inability to match related data from numerous and heterogeneous databases and systems. To start facing some of these challenges, the Publications Office published an OWL ontology named ePO - eProcurement Ontology[2], aligned with the latest related EU directives and regulations. ePO describes the main objects of public procurement and their relationships.

In order to test and exploit this ontology, the European Commission has implemented a pilot project whose aim is to lay the foundations for the creation of a European public procurement data space. In this data space, a semantic knowledge graph, i.e., a knowledge graph constructed using semantic web standards such as RDF and OWL (henceforth referred to as 'KG'), is exploited for the integration of data between different public procurement actors. The KG consists of public procurement data modelled through the aforementioned ePO ontology.

In the light of this scenario, the main contributions of this paper are:

- a distributed architecture that exploits semantic web technologies for the EU public procurement data space, where different governance scenarios are possible;
- a novel declarative approach for creating and managing KGs. This approach consists of defining an OWL ontology we present, whose instances are declarative descriptions used by a workflow engine. The workflow engine orchestrates tasks based on these declarative descriptions, aiming at transforming input datasets into the desired representation. Overall, this contributes to the creation of the KG of the data space, reducing possible manual interventions and making it maintainable and sustainable over time;
- a workflow process that, using this OWL ontology, is able to orchestrate the tasks to be performed to produce RDF datasets, compliant with a reference domain ontology;
- an open data based approach for ETL - Extract Transform and Load, by which the catalogue of transformed federated datasets is built in, thus reducing the maintenance efforts and increasing overall consistency;
- a number of lesson learnt for future developments of the EU Public Procurement data space.

The rest of this paper is structured as follows. Section 2 presents an overview of related work. Section 3 describes the pilot experience and its configuration. Section 4 introduces the solution we designed and implemented for the realisation of the pilot. Section 5 discusses the main lesson learnt and Sect. 6 the uptake. Finally, Sect. 7 concludes the paper with future work.

[1] https://ec.europa.eu/growth/single-market/public-procurement_en.
[2] https://github.com/OP-TED/ePO.

2 Related Work

We present different works that we have analysed because they are similar to the overall work we propose. These are divided into: i) similar approaches in the use of semantic standards in the procurement domain; ii) similar works on the use of ontologies as declarative descriptions to govern workflow systems.

In the procurement domain, semantic technologies have been used in different projects. A recent one is The Buy for You Platform [24] that applies an approach that is similar to the one used in our pilot. It exploits KGs based on ontologies, proposing an infrastructure with rest APIs for easy access to data. The ontologies form a network and deal with two types of data: procurement-related data (e.g., contract, award, plan, tender) on one hand [25] and company data on the other hand (e.g., registered organisation, address, site). For the procurement-related data, it uses a data specification that is emerging in the contract management context named OCDS - Open Contracting Data Specification, entirely based on JSON and JSON-based rest APIs.

Other past attempts to model public procurement have been done with the LOTED2 [13], PPROC [21] and PublicContract[3] OWL ontologies; however, they seem focused on some specific elements of the procurement, only: LOTED2 on legal notices, PPROC and PublicContract on public contracts.

As for the use of ontologies for guiding the KG production, in [11], the authors pose a set of research challenges, also mentioning the use of "*declarative descriptions of workflows*" as a possible technique that is appearing, as we proposed in our pilot.

In [5], the authors introduce TITAN, a system that uses the BIGOWL ontology for describing workflows and entities that contain software components of the system. TITAN proposes a similar approach to ours, but more general. In contrast, we focus on describing specific activities on the creation of KGs in specific contexts, and for this we extend ontologies used in the public sector to document datasets in catalogues.

In [15] and [16], the LinkedPipes ETL tool is introduced and described. Its aim is to support the whole process of data publication, especially the lifting of internal data in relational databases or Excel, CSV, XML or JSON files to Linked Open Data, with a successive cataloguing activity. The data transformation pipelines are stored in the system as RDF but no specific OWL ontology is used to govern the pipeline, as in our case.

In [23], the authors provide a holistic approach and architecture to populate a commercial KG based on heterogeneous data sources. Although the approach is similar to ours, and enables the automation of the creation of the KG, the use of an OWL ontology that describes the workflow in a declarative way is not treated as in our case. The authors use the PROV-O ontology to keep track of the source information, but do not exploit it to provide the necessary instructions for a workflow system as we propose.

[3] https://w3id.org/italia/onto/PublicContract.

3 The Public Procurement Pilot

An attempt to get public procurement data at EU level has been done by establishing an European system named TED (Tender Electronic Daily) that mandates Member States to publish all the notices of their national tenders above the regulatory thresholds. While the latest reform of the regulation intends to get more and better public procurement data, it does not address some problems that were detected with the system [2]: i) fragmentation and complexity of procurement systems in Member States; ii) lack of compatibility between TED and Member State systems; iii) publication of mainly documents (notices) rather than data. This prevents the adoption of an effective data-driven approach to public procurement depriving the stakeholders of the possible savings and improvements that such a paradigm can bring, even in terms of transparency, corruption fight and governance of public procurement.

In 2020, Italy requested the ISA2 programme to develop, maintain and promote an infrastructure to gather, process, analyse and publish public procurement data based on the earlier cited ePO ontology. One key requirement is to work on reusable open source tools that can be implemented in the national (or regional) eProcurement infrastructures to carry out successive data analysis. In essence, the idea is to lay the foundations for creating a data ecosystem. Within it, public procurement data and data products can be seamlessly exchanged among stakeholders, allowing for their reuse to build advanced applications and services.

The pilot was launched after gathering strategic input from the Analytics subgroup of the expert group on eProcurement, who expressed the following guiding principles: i) to allow all data sources to be included in a reusable way, once they become relevant for supporting the policy objectives; ii) to make data timely accessible, traceable and comparable; iii) to reuse as much as possible data, data products and tools.

In the light of these considerations, the objectives of the pilot are: i) to explore the harmonisation of the public procurement data landscape thanks to the use of the ePO ontology built for such purpose; ii) to pilot a federated solution, paving the way towards a data space instead of a centralized data warehouse; iii) to explore the construction of quality processes and use of tools that involve the data owners and data providers at various levels: EU, national and local. To simulate the heterogeneity of the European public procurement landscape, the pilot selected several national data providers: ANAC the Italian National Anti-Corruption Authority collecting all Italian procurement data, IMPIC, the Portuguese public authority collecting all Portuguese procurement data, DFO the Norwegian Public and Financial Management Agency collecting all the Norwegian procurement data. In the Public Procurement data provisioning landscape, CONSIP, the Italian Central Purchasing Body, a primary data owner, is also involved to explore how the existing organisation of data provisioning mandated by law may be complemented by voluntary adhesion of data owners to the federated data space. The pilot also involves the Publications Office, owner of the earlier cited TED system and EU data provider. The Institute of

Cognitive Science and Technologies (ISTC) of the Italian National Council of Research (CNR) contributes in the pilot from a technological transfer perspective, supporting in the technical work related to the use of semantic technologies. Directorate General DIGIT of the EU Commission coordinates the pilot.

The heterogeneity and complexity to be dealt with in the construction of the resulting KG led the participants to automate the transformation processes, from the very beginning, in such a way as to reduce as much as possible any manual interventions.

The pilot aims at analysing the number of received tenders since 2017 until the latest available data, using contract award notice information. Therefore, in the KG, we did not instantiate all the ePO ontology (version 2.0.1) elements; rather, we mainly used the following classes: `Procedure`, `Lot`, `Technique`, `Purpose`, `StatisticalInformation`, `AwardDecision`, `ContractAwardNotice`, `Organisation`, `Role`.

4 Proposed Solution for the Pilot Implementation

As shown in Fig. 1, we designed a distributed architecture coherent with the pilot objectives and guiding principles. Multiple data sources are used, with different source data models, reflecting the diversity of the landscape. Two partners, Consip and Publications Office contribute with linked open datasets already in compliance with the ePO ontology, produced through their internal processes and infrastructures. The ISTC-CNR partner supported Consip in their KG construction processes, providing the required mapping rules from the original data to the ePO-based RDF target datasets. The rest of the partners from Italy, Portugal and Norway contribute with many open datasets available in a variety of data formats and structures (see Fig. 1). This requires data transformations that have been carried out using the European Big Data Test Infrastructure (BDTI) (see Sect. 4.1).

Within the BDTI, a transformation process is managed by a workflow management system whose tasks are governed by the instances of the OWL transformation ontology we developed for such a purpose (Fig. 1).

4.1 The Big Data Test Infrastructure

The Big Data Test Infrastructure (BDTI[4]) is a technical building block of the Digital Europe Programme of the European Commission that can be used, on a per-request basis, to support public administrations in their prototype analytic and Big Data solutions. Instead of setting up a testing environment for these solutions, the use of such an infrastructure allows public administrations to concentrate on the core business, insights and value they can obtain from their data.

[4] https://ec-europa.github.io/bdti-infrastructure/.

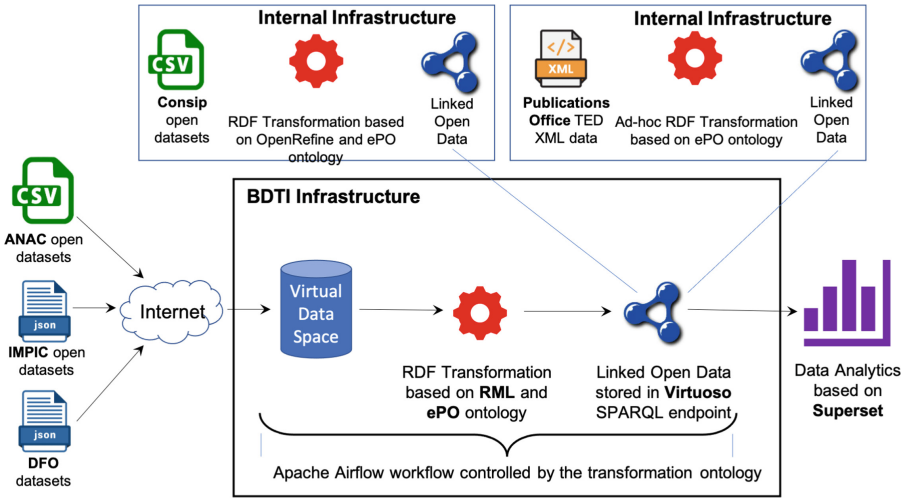

Fig. 1. Pilot architectural scenario

The infrastructure was assessed as particularly useful to support all those public sector partners in the pilot that do not participate with their own internal tools but only by providing open datasets already available in their data catalogues in different formats (e.g., JSON, CSV).

In particular, the BDTI was used to: (i) manually save datasets from ANAC, IMPIC and DFO in the BDTI cloud storage space; (ii) transform the content of the datasets into a KG according to the RDF standard and the ePO ontology earlier mentioned; (iii) publish the data in the SPARQL Virtuoso endpoint instance of the BDTI; and (iv) publish the metadata of transformed data in the SPARQL Virtuoso endpoint instance of the BDTI, thus forming a catalogue of federated transformed data sources.

4.2 Data Transformation Process in the BDTI

To carry out all these activities, we designed and implemented a process that, starting from datasets located in the cloud storage space of the BDTI, is capable of transforming the data into a KG by leveraging the RDF Mapping Language (RML) [12], using a set of its functions for data manipulation purposes (e.g., `array-join` for defining URIs[5], `controls_if` for verifying specific values). In the RML mapping rules, we also managed the creation of links (i.e., `owl:sameAs`) to other linked open datasets available in the Web of Data such as controlled

[5] We used the same URI schema for all those partners using the BDTI. The schema followed the '10 persistent rules for URIs' - https://joinup.ec.europa.eu/collection/semantic-interoperability-community-semic/document/10-rules-persistent-uris, where the domain part depends on the specific EU country.

vocabularies[6] published by the Publications Office and recommended in the ePO ontology. The RML mapping rules[7], expressed in R2RML [9] syntax, were saved in the cloud storage space of the BDTI and executed using the RML mapper[8] through instructions configured in a workflow management system.

In order to make this process manageable and sustainable over time, thus minimising any possible manual interventions, we designed an OWL ontology that describes all the activities and resources required by a workflow engine, used successively to orchestrate the stages of the building process. In essence, the RDF triples, instances of the OWL ontology we introduce in this paper, can be thought of as *declarative descriptions for a workflow system*. In the implementation of our pilot, we adopted Apache Airflow (see below) as workflow engine. We argue that one of the strengths of this approach is that the update of transformed datasets can be done reducing any manual interventions by querying the specific metadata of the input datasets (e.g., last modification date), while the monitoring of the construction of the KG is ensured by querying the transformation metadata. Finally, a further unforeseen result is that the declaration of transformations contributes to the creation of a catalogue of federated transformed data sources, ensuring by design their findability.

Transformation Ontology. The OWL ontology that controls the transformation process is illustrated in Fig. 2.

Ontology Modelling Approach. It is grounded on two foundational ontologies for metadata description; namely, DCAT-AP - European Application Profile for Data Catalogue Vocabulary [10], which extends the DCAT Web Recommendation [4] in order to describe datasets available in data catalogues, and PROV-O - Provenance Ontology [17], another Web Recommendation which allows one to represent all provenance information related to activities and entities. Our ontology imports PROV-O and extends it with a minimum set of classes and properties (the bottom level in Fig. 2) that represent the specific transformation activities and resources to be done and used in the KG construction process. Moreover, we extend DCAT-AP, based on DCAT version 2, by defining a data distribution concept used to support the core elements of the ontology (see below). In general, we favoured the approach of maintaining the control on our semantics and extend existing ontologies according to our requirements. In essence, we applied an indirect re-use of existing ontologies [22].

The resulting ontology is simple, with elements that can be clearly understood in contexts such as the public sector, as the use of DCAT-AP is becoming increasingly popular due to European and national requirements for federated data catalogues.

Competency Questions. The ontology has been developed using the methodology available in the literature called eXtreme Design [6,7] (e.g., definition of CQs,

[6] https://op.europa.eu/en/web/eu-vocabularies/authority-tables.
[7] https://git.fpfis.eu/public-datateam/eprocurement/-/tree/develop/rml-mappings.
[8] https://github.com/RMLio/rmlmapper-java.

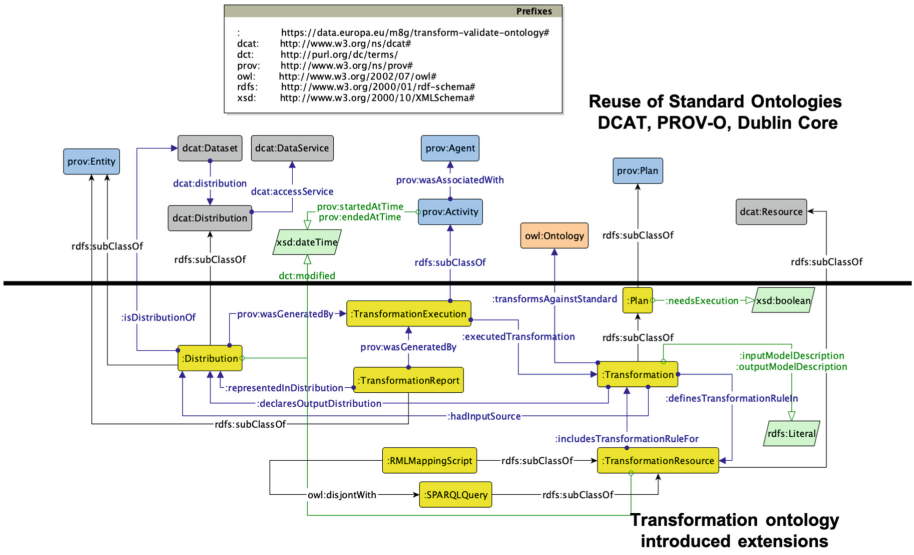

Fig. 2. Graffoo diagram of the transformation ontology

reuse of ontology design patterns). Therefore, we started from the elicitation of specific requirements translating them into so-called Competency Questions (CQs) that represent the de-facto ontological commitments. A non exhaustive list of CQs for the transformation process modelled in the ontology is provided in Table 1.

Ontology Description. A transformation (the class `:Transformation`) is a specific type of PROV-O plan (thus represented as subclass of `prov:Plan`), and it is defined as a planned set of operations to be executed by one or more agents; it aims at transforming a given input dataset distribution into an output dataset distribution.

To identify a dataset distribution, which is a representation of a dataset used to distribute it according to different serializations or formats, we extend the same concept as the one defined in DCAT so as to link it to the core elements of the proposed ontology. For instance, we added an inverse property from our `:Distribution` concept to the `dcat:Dataset` class and an OWL restriction that represents the connection of the distribution of a dataset to the execution of a transformation plan. This extension is represented by the class `:Distribution` (bottom part of Fig. 2); it inherits all the properties of the main `dcat:Distribution` (e.g., `dct:modified`, `dcat:accessURL`, etc.), including the relationship with the class `dcat:DataService`.

Table 1. Competency questions of the OWL ontology.

ID	Competency question
CQ1	Which is the input distribution to be used for the transformation?
CQ2	Which are the standards used in the transformation?
CQ3	Which is the transformation resource to be used in a transformation plan?
CQ4	What is the output dataset distribution generated by a transformation?
CQ5	Who executed the transformation activity?
CQ6	When the transformation resource of the transformation plan has been updated?
CQ7	Which are the output distributions generated by the execution of a transformation plan?

A transformation plan defines transformation rules within specific types of transformation resources (the class `:TransformationResource` intended as a subclass of `dcat:Resource`).

During our pilot, we identified two types of transformation resources; namely RML mapping rules files (the class `:RMLMappingScript` which is currently the de-facto standard for the construction of KGs, and SPARQL query. This latter class `:SPARQLQuery` allows us to represent alternative approaches with respect to the use of mapping languages like RML. Tools such as SPARQL Generate [18] or SPARQL Anything [8] can be captured using the `:SPARQLQuery` class where a SPARQL query is used to specify mapping rules. We believe that these transformation resources are sufficient to model well-established mechanisms for transforming different dataset formats (e.g., XML, JSON, CSV) into RDF, thus making the ontology applicable in domains other than our own, where RML mapping rules only are used.

A `:TransformationExecution` activity (a subclass of `prov:Activity`), executed by some Agent (`prov:Agent`), is defined. It generates a dataset distribution (`:Distribution`), executes (the `:executeTransformation` property) a transformation plan and produces a report (the class `:TransformationReport`). This activity is started and ended at some time (since `:TransformationExecution` is a subclass of `prov:Activity`, it inherits the properties `prov:startedAtTime` and `prov:endedAtTime` both typed literals `xsd:dateTime`). The produced report is a `prov:Entity` representing any return message that gives information on the success or otherwise of the transformation operation.

Transformation System. In order to execute the data transformation process at scheduled times and based on the activities and resources identified by the instances of the transformation OWL ontology, it was necessary to select a workflow/task runner engine. Apache Airflow [1] was selected as it is the most suitable solution for the purpose that meets the following criteria: i) Open source, as to lower the barriers for adoption of the paradigm; ii) scalable; iii) tasks can

be defined in code, so that the instance data of the ontology can be used to define the tasks. Apache Airflow fits these criteria as it is Open source software that allows for the scheduled execution of workflow tasks on a cluster of workers. Airflow provides the framework for workflow definition and scheduling, but the actual task execution is delegated to a Celery cluster. The Celery cluster is a distributed job queue: jobs get added to the queue, and are executed asynchronously on the worker nodes. This allows one to scale the process, as additional worker nodes can be added.

Inside Airflow, workflows consist of tasks which can depend on one another. Each task can be executed by a cluster node, once all its dependencies have successfully been fulfilled; the upstream tasks have been executed successfully. This model allows tasks to be performed in parallel as much as possible, limited only by the tasks dependencies and the availability of cluster capacity to execute the task. Since tasks can be scheduled on any node in the Celery cluster, data cannot be stored on disk at the node. Instead, an external system should be used, such as a database, object store or network file system, which must be moved to and from the node when needed. Moreover, although not implemented in the pilot, the model allows transformations to depend on multiple input distributions, which in turn could be the result of another transformation. As transformations are executed once one of their input distributions has changed (this is detected by the date of last update, i.e., the `dct:modified` property of the class `:Distribution`), a more complex logic should be considered to determine the order of scheduling if several input distributions of the transformation have a high update frequency.

Workflow Definition. In our pilot, the (extended) catalogue is the only place where the state of the workflow process is persisted. This guarantees a clear separation between the business processes whose output is recorded in the catalogue and the operational side, in the form of code executed by the engine. The workflow representation in Apache Airflow (tasks and their dependencies) is created through the execution of a Python program, that takes the instance data of the ontology as its input. The following instance data in Listing 1.1 is an example of how a ETL transformation can be defined.

Listing 1.1. Turtle instance data for transformation

```
@prefix dcat: <http://www.w3.org/ns/dcat#> .
@prefix etl: <https://data.europa.eu/a4g/transform-validate-
    ontology#> .
@prefix eproc: <http://eprocurement-placeholder/> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .

# Datasets
eproc:example_input_dataset a dcat:Dataset;
    dcat:distribution eproc:example_input_distribution .

eproc:example_transformed_dataset a dcat:Dataset;
    dcat:distribution eproc:example_output_dist .
```

```
# Distributions
eproc:example_input_distribution a etl:Distribution ;
    dcat:accessURL eproc:input.csv .

eproc:example_output_distribution a etl:Distribution .
    dcat:accessURL eproc:output.ttl .

# Transformation
eproc:example_transformation a etl:Transformation ;
    rdfs:label "Example data transformation" ;
    etl:hadInputSource eproc:example_input_distribution ;
    etl:declaresOutputDistribution eproc:
        example_output_distribution ;

    etl:definesTransformationRuleIn eproc:
        example_rml_transformation_script .

# Transformation Resource
eproc:example_rml_transformation_script a etl:RMLMappingScript ;
    rdfs:label "RML mapping rules used to transform the input
        distribution into the output distribution." ;
    etl:accessURL eproc:rml-transformation-rules.ttl ;
```

By using the instance data, it is possible to automatically generate the workflow in Apache Airflow. The basis of the process is that each instance of the `:Transformation` class in the catalogue (eproc:example_transformation in Listing 1.1) is turned into a workflow object.

Listing 1.2. Apache Airflow code for transformation data

```
    g = Graph()
    # Parse turtle file into in-memory graph
    g.parse("catalogue.ttl", format='text/turtle')
    # Use catalogue graph to create entity model
    catalogue = EntityRepository(g)
    transformations = catalogue.getTransformations()
    workflow_creator = DagTransform()
    for transformation in transformations:
        # Create workflow object from the transformation instance.
        workflow = workflow_creator.transformationToDag(
            transformation)
```

In its most basic setup, each workflow contains a single Transformation task, which performs the execution of the transformation script. Additional tasks can be defined, e.g. to load the transformed data into a target database. To access the data catalogue in a developer friendly way, and separate the ontology/data concerns from workflow's business logic, a rudimentary Object RDF Mapper (ORM)[9] is developed and used.

[9] ORM code on GitLab.

Workflow Execution. The tasks are planned by the Airflow scheduler, and executed by the Celery cluster. When the transformation task is executed, the `:TransformationExecution` class is instantiated. In Airflow, the equivalent of a `:TransformationExecution` is an Airflow DAG run. The main steps then can be summarised as follows.

- **Extract.** The file referenced by the `dcat:accessUrl` property of the input distribution, referenced through the `:hadInputSource` property of the ontology defined for the `:Transformation` class instance, gets downloaded to the Celery worker node. This file is the input data to the transformation process. Also, the `:RMLMappingScript` (in case of a RML transformation), also referenced from the `:Transformation` class instance, is downloaded to the node. In our pilot, both are stored in the AWS S3 objectstore. In the future, the system can be extended to support a wider variety of transformation systems. A plugin would subclass the `:TransformationResource` of the ontology and the Airflow code to support the transformation engine.
- **Transform.** After moving the downloaded `:Distribution` into the working directory of the Airflow runner, the transformations must be executed. In our pilot, this is done via the external executable process RMLMapper, passing it the file name of the input `:Distribution` and the RML mapping rules file(s) as parameters. The result of the transformation is stored in a temporary file on the Celery node.
- **Load.** The transformation result is written back to the `dcat:accessUrl` of the output distribution. In our pilot, this is the S3 object store. It is worth noting that this approach differs from a traditional ETL process, where the Load stage loads the data into the target database. In our case, the data merely gets stored as a file. If further representation of the data (for instance in a triplestore) is needed, an instance of the `dcat:DataService` class (see Fig. 2) must be added, linked to the output distribution. This will result in an additional workflow task to be added to the workflow to materialise the data into the database.

5 Lesson Learnt

From the pilot project experience we can draw a number of lessons learnt, useful for anyone, in different domains, when leveraging semantic technologies and KGs as means for the definition of a data space. These, related with each other, are summarised as follows.

RDF Declarative Approach to Data Transformation. The instantiation of our transformation ontology, as a declarative description of jobs to be executed by a workflow engine, allows us to make the process of building the KG sustainable and maintainable over time, as manual human interventions are greatly reduced. We argue that this approach is particularly effective in the scenarios we faced, where large numbers of data distributions consisting of even more than 100 data files for 4 years of procurement data for Italy, only, must be managed and transformed.

Lightweight Transformation Ontology. The benefits of extending DCAT-AP and PROV-O, well-known standards of the Semantic Web, to manage the transformation operations on the data are: i) helping in monitoring which data sources have been analysed and then transformed; ii) guaranteeing the discoverability of the transformed data sources results as the metadata of the output datasets is added to the output data catalogue at the time of the declaration of the transformation; iii) allowing for monitoring the transformation operations, thus understanding the status of the overall construction process; iv) allowing for traceability of the operations performed between the input data source and the output data source, discoverable through the data catalogue; v) allowing for automating the refresh of output datasets when input datasets have changed or when the transformation code has been revised.

Use of EU Commodities. Most of the pilot participants did not own infrastructures for managing KG. The use of commodities like the BDTI becomes crucial when supporting the data space establishment.

Fostering DCAT-AP in Europe. The use of DCAT-AP for datasets findability is increasing in Europe. However, this is not yet common practice in all EU countries. Due to the role of DCAT we described in this paper, promoting its adoption is crucial. In addition, adopting the solution we propose naturally contributes to increasing the reach of DCAT-AP.

Define a Common Language in the Data Space. In a data space, one key point is that actors 'speak the same language'. Data transformation towards a shared semantic layer, like the ePO ontology, has to happen as soon as possible in the data management process so as to build additional artefacts on a standardised and high quality set of datasets.

Define Streamlined ETL Processes. In a data space, another key point is that data is of good quality. Our generic approach ensures that the risks of degrading data quality through transformation are minimized. This is guaranteed thanks to the separation of concerns between the transformation scripts and automation of the process.

Issues When Working with Current Available Open Data. While working with existing open datasets seems desirable as a set of available resources that can be easily re-used, the pilot identified a drawback in this scenario: open data is often treated as a process apart from the main internal data management processes (processes on data that is not publicly available). This practice inevitably introduces delays between data changes in internal systems (e.g. a data warehouse) and the publication of data under the open data paradigm. In addition, it reduces the potential richness of the data as not all that available is publicly published. In essence, the mere use of these open sources may hinder easier and more timely data management than would be possible with a direct access to the data stored in internal systems. In our experience, some input open datasets

required a first data manipulation for allowing RML processes to run smoothly. This was particularly the case with Portuguese JSON files: the lot identifiers were simply incremental numbers without including the relevant procedure context. Due to some limitations in navigating JSON files in RML, this scenario prevented us from constructing persistent URIs for the lots. Therefore, a manipulation of the data to include the identifiers of the parent procedure in the lot identifier was done in Python. Finally, when linking some datasets from Italy to TED open datasets, we discovered entity duplication issues in TED. This happened when the same entity was used in different phases of the procurement (at contract notice and contract award notice times). The Publications Office is carrying out a work to ensure entity deduplication. These issues did not occur for the datasets we produced within the BDTI.

6 Uptake

The pilot experience led to follow-up actions described below. Firstly, Consip decided to publish online for anyone its produced portion of KG. Therefore, they enriched their open data catalogue[10] with a specific section named "Linked Open Data"[11] where the results of the work carried out in the pilot can be queried and re-used.

Secondly, the proposed RDF declarative approach to data transformation is used in a European funded project named WHOW - Water Health Open Knowledge[12]. In WHOW, open datasets located in data catalogues and documented using DCAT-AP are to be transformed in linked open data and the use of such an approach allows the project to meet its objectives in a sustainable and maintainable manner [19].

Finally, the future Public Procurement Data Space (PPDS) that the European Commission is currently designing and implementing will leverage the main results and digital artefacts presented in this paper. In particular, the PPDS is considering the transformation ontology as a key asset to support the transformation process through the use of a workflow engine. The plan also foresees to extend this approach for automating data extraction from data catalogues in Europe, validating the data according to specific business rules. The plan is not yet publicly available for anyone; however, from a high level overview of the public procurement data strategy[13], the main principles here described can be found.

[10] https://dati.consip.it/.
[11] https://dati.consip.it/linked_opendata.
[12] https://whowproject.eu/.
[13] https://vkazprodwordpressstacc01.blob.core.windows.net/wordpress/2021/07/PP-Data-strategy.pdf.

7 Conclusions and Future Work

This paper shows that the construction of a European public procurement data space based on semantic web standards and technologies and reusable open software solutions is feasible and effective in ensuring interoperability. It focuses on a distributed architecture capable of dealing with different data governance scenarios, where RDF transformations are performed and orchestrated via instances of an OWL ontology that describes the tasks of a workflow system.

Future Work. There is currently an on-going work for officially assigning to the presented ontology an URI under the European Core Vocabularies namespace, according to the URI policies adopted by the EU institutions and bodies[14]. This will also enable content negotiation mechanisms for the proposed ontology. We are planning to implement the workflow that allows us to validate the transformation against specific procurement business rules. In this sense, we have already considered the use of the ontology to control the execution of different types of validation, through existing validation engines (e.g., the SHACL validator already provided in the BDTI[15]).

Moreover, we are planning to extend the transformation ontology in order to represent data quality metrics. These can be used for example to create a transformation and validation monitoring dashboard that developers can leverage in assessing the overall effectiveness of the KG construction process. The Data Quality Vocabulary [3] can be taken into account as an additional modelling part of the proposed transformation ontology.

Finally, further investigation can be required to understand how the workflow engine can be made more flexible through ontology-code plugins, following the approach of the function ontology [20]. A plugin would consist of a function definition and an implementation in code. For example an 'FTP Distribution' plugin would allow for transparent access of distributions accessible over FTP. A micro-kernel architecture would allow one to add plugins to the workflow engine in a modular way.

Supplemental Material Availability: The source code and RML mapping rules that have been produced for the knowledge graph production process in the BDTI can be found in the following GitLab space: https://git.fpfis.eu/public-datateam/eprocurement

The transformation ontology is open for the re-use by anyone and it is available for the download on the gitlab repository of the European pilot project[16]. Moreover, we setup a github repository[17] to let users navigate it via HTML[18] by means of tools such as Widoco [14].

[14] https://data.europa.eu/URI.html.
[15] https://www.itb.ec.europa.eu/shacl/any/upload.
[16] https://git.fpfis.eu/public-datateam/eprocurement/-/blob/develop/transform-validate-ontology.ttl.
[17] https://github.com/transformationvalidation/transformationontology.
[18] https://transformationvalidation.github.io/transformationontology/.

References

1. Apache Airflow (2022). https://airflow.apache.org/
2. Ackermann, R., Sanz, M., Sanz, A., Milicevic, V.: Gaps and errors in the ted database (2019). https://www.europarl.europa.eu/cmsdata/161426/CONT_Gaps
3. Alberton, R., Isaac, A.: Data on the Web Best Practices: Data Quality Vocabulary - W3C Working Group Note, December 2016. https://www.w3.org/TR/vocab-dqv/
4. Albertoni, R., Browning, D., Cox, S., Beltran, A.G., Perego, A., Winstanley, P.: Data Catalog Vocabulary (DCAT) - Version 2–W3C Recommendation https://www.w3.org/TR/vocab-dcat-2/ (February 2020)
5. Benítez-Hidalgo, A., et al.: TITAN: a knowledge-based platform for big data workflow management. Knowl.-Based Syst. **232**, 107489 (2021). https://doi.org/10.1016/j.knosys.2021.107489
6. Blomqvist, E., Hammar, K., Presutti, V.: Engineering ontologies with patterns - the eXtreme design methodology. In: Hitzler, P., Gangemi, A., Janowicz, K., Krisnadhi, A., Presutti, V. (eds.) Ontology Engineering with Ontology Design Patterns - Foundations and Applications, Studies on the Semantic Web, vol. 25. IOS Press (2016). https://doi.org/10.3233/978-1-61499-676-7-23
7. Blomqvist, E., Presutti, V., Daga, E., Gangemi, A.: Experimenting with eXtreme design. In: Cimiano, P., Pinto, H.S. (eds.) EKAW 2010. LNCS (LNAI), vol. 6317, pp. 120–134. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-16438-5_9
8. Daga, E., Asprino, L., Mulholland, P., Gangemi, A.: Facade-x: an opinionated approach to SPARQL anything. In: Alam, M., Groth, P., de Boer, V., Pellegrini, T., Pandit, H.J. (eds.) Volume 53: Further with Knowledge Graphs, vol. 53, pp. 58–73. IOS Press (2021). http://oro.open.ac.uk/78973/
9. Das, S., Sundara, S., Cyganiak, R.: R2RML: RDB to RDF Mapping Language - W3C Recommendation, September 2012. https://www.w3.org/TR/r2rml/
10. DIGIT: European Commission: Discover the new DCAT-AP release 2.0.1 - Joinup, June 2020. https://joinup.ec.europa.eu/collection/semantic-interoperability-community-semic/news/dcat-ap-release-201
11. Dimou, A., Chaves-Fraga, D.: Declarative description of knowledge graphs construction automation: status and challenges. In: To appear in Proceedings of Third International Workshop on Knowledge Graph Construction, KGCW 2022, Greece, May 2022
12. Dimou, A., Vander Sande, M., Colpaert, P., Verborgh, R., Mannens, E., Van de Walle, R.: RML: a generic language for integrated RDF mappings of heterogeneous data. In: Proceedings of the 7th Workshop on Linked Data on the Web, April 2014. http://events.linkeddata.org/ldow2014/papers/ldow2014_paper_01.pdf
13. Distinto, I., d'Aquin, M., Motta, E.: LOTED2: an ontology of European public procurement notices. Semant. Web **7**(3), 267–293 (2016)
14. Garijo, D.: WIDOCO: a wizard for documenting ontologies. In: d'Amato, C., et al. (eds.) ISWC 2017. LNCS, vol. 10588, pp. 94–102. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-68204-4_9, http://dgarijo.com/papers/widoco-iswc2017.pdf
15. Klímek, J., Škoda, P.: Linkedpipes ETL in use: practical publication and consumption of linked data. In: Proceedings of the 19th International Conference on Information Integration and Web-based Applications and Services, pp. 441–445 (2017)

16. Klímek, J., Skoda, P.: Linkedpipes DCAT-AP viewer: a native DCAT-AP data catalog. In: International Semantic Web Conference (P&D/Industry/BlueSky) (2018)
17. Lebo, T., Sahoo, S., McGuinness, D.: PROV-O: The PROV Ontology - W3C Recommendation, April 2013. https://www.w3.org/TR/prov-o/
18. Lefrançois, M., Zimmermann, A., Bakerally, N.: A SPARQL extension for generating RDF from heterogeneous formats. In: Proceedings of Extended Semantic Web Conference (ESWC 2017), Portoroz, Slovenia, May 2017. http://www.maxime-lefrancois.info/docs/LefrancoisZimmermannBakerally-ESWC2017-Generate.pdf
19. Lippolis, A.S., et al.: Linked open data process design is finalised, June 2022. https://doi.org/10.5281/zenodo.6685819, https://doi.org/10.5281/zenodo.6685819, Deliverable n. 3.2 Activity title: Knowledge Graph Definition - Task 3.3 Linked Open Data production process design
20. Meester, B.D., Dimou, A., Verborgh, R., Mannens, E.: An ontology to semantically declare and describe functions. In: ESWC (2016)
21. Muñoz-Soro, J.F., Esteban, G., Corcho, O., Serón, F.: PPROC, an ontology for transparency in public procurement. Semant. Web **7**(3), 295–309 (2016)
22. Presutti, V., Lodi, G., Nuzzolese, A., Gangemi, A., Peroni, S., Asprino, L.: The role of ontology design patterns in linked data projects. In: Comyn-Wattiau, I., Tanaka, K., Song, I.-Y., Yamamoto, S., Saeki, M. (eds.) ER 2016. LNCS, vol. 9974, pp. 113–121. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46397-1_9
23. Simsek, U., Umbrich, J., Fensel, D.: Towards a knowledge graph lifecycle: a pipeline for the population of a commercial knowledge graph. In: Proceedings of Conference on Digital Curation Technologies (Qurator). CEUR-WS, Berlin (2020). http://ceur-ws.org/Vol-2535/paper10.pdf
24. Soylu, A., et al.: Theybuyforyou platform and knowledge graph: expanding horizons in public procurement with open linked data. Semant. Web **13** (2021). https://doi.org/10.3233/SW-210442
25. Soylu, A., et al.: Towards an ontology for public procurement based on the open contracting data standard. In: Pappas, I.O., Mikalef, P., Dwivedi, Y.K., Jaccheri, L., Krogstie, J., Mäntymäki, M. (eds.) I3E 2019. LNCS, vol. 11701, pp. 230–237. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-29374-1_19