information

Article

Systematic Construction of Knowledge Graphs for Research-Performing Organizations

David Chaves-Fraga [1,*], Oscar Corcho [1], Francisco Yedro [1], Roberto Moreno [2], Juan Olías [2] and Alejandro De La Azuela [2]

[1] Ontology Engineering Group, Universidad Politécnica de Madrid, 28660 Boadilla del Monte, Spain
[2] Universitas XXI, 28043 Madrid, Spain
* Correspondence: david.chaves@upm.es

Abstract: Research-Performing Organizations (e.g., research centers, universities) usually accumulate a wealth of data related to their researchers, the generated scientific results and research outputs, and publicly and privately-funded projects that support their activities, etc. Even though the types of data handled may look similar across organizations, it is common to see that each institution has developed its own data model to provide support for many of their administrative activities (project reporting, curriculum management, personnel management, etc.). This creates obstacles to the integration and linking of knowledge across organizations, as well as difficulties when researchers move from one institution to another. In this paper, we take advantage of the ontology network created by the Spanish HERCULES initiative to facilitate the construction of knowledge graphs from existing information systems, such as the one managed by the company Universitas XXI, which provides support to more than 100 Spanish-speaking research-performing organizations worldwide. Our effort is not just focused on following the modeling choices from that ontology, but also on demonstrating how the use of standard declarative mapping rules (i.e., R2RML) guarantees a systematic and sustainable workflow for constructing and maintaining a KG. We also present several real-world use cases in which the proposed workflow is adopted together with a set of lessons learned and general recommendations that may also apply to other domains. The next steps include researching in the automation of the creation of the mapping rules, the enrichment of the KG with external sources, and its exploitation though distributed environments.

Keywords: knowledge graph; research-performing organizations; declarative mapping rules

check for updates

Citation: Chaves-Fraga, D.; Corcho, O.; Yedro, F.; Moreno, R.; Olías, J.; De La Azuela, A. Systematic Construction of Knowledge Graphs for Research-Performing Organizations. *Information* **2022**, *13*, 562. https://doi.org/10.3390/info13120562

Academic Editor: Ryutaro Ichise

Received: 14 October 2022
Accepted: 24 November 2022
Published: 30 November 2022

1. Introduction

Research-performing organizations such as universities and research centers collect and accumulate a large amount of data related to their activities (e.g., scientific results, project outputs, academic courses, etc.). Although there are some common information models (e.g., EuroCRIS [1]), in this domain it is common practice for each of these organizations to develop their own information system to support all their activities. This fact has a negative impact on integrating and exploiting knowledge across institutions and also makes it difficult for researchers to manage their data when they move from one organization to another. The effectiveness of semantic web technologies and knowledge graphs [2] for complex data management tasks has already been demonstrated in several domains [3–5], by companies (e.g., Google [6], Amazon [7]) and public communities (e.g., DBpedia [8], Wikidata [9]).

HERCULES is a multi-anual project, promoted by the main Spanish association of universities (CRUE) [10], that aims to build a semantic layer to harmonize the knowledge and data of the information systems of Spanish research-performing organizations. The main objective is to address data interoperability problems across organizations in terms of schemes and formats, ensuring efficient exploitation of combined knowledge.

HERCULES covers the description of PhD courses, preparation and tracking of research projects, and management of research outputs (e.g., papers, datasets, etc.) among others.

The project has already developed an ontology [11] that represents this domain, the Hercules Ontology Network (ROH). Its authors follow a bottom-up approach, taking the data model from the University of Murcia [12], as the main input while considering existing ontologies in this domain. Now, the rest of Spanish universities are encouraged to transform their data into a compliant knowledge graph with the aforementioned vocabulary, but the project neither specifies how to do it nor provides any technological support for that process.

In this paper, we describe the process followed in the creation of knowledge graphs compliant with the HERCULES ontology. As a real-use case, we are supported by the information system provided by the company Universitas XXI, which provides support to more than 100 research-producing organizations around the world (mostly in Spain and Latin America). In addition to following the modeling choices from the ontology, we develop a sustainable and systematic workflow to construct knowledge graphs using standard semantic web technologies (i.e., R2RML mapping rules). As a result of this work, and based on our previous experiences in similar contexts, we present several real use cases where research-performing organizations have already included this workflow in their information systems, together with a set of lessons learned and general recommendations that may also apply to other domains.

The remainder of this paper is structured as follows: Section 2 provides an overview of the state of the art in KG construction and the use of mapping rules in real-world scenarios. Section 3 describes in detail the ROH ontology used in our work. Section 4 presents the sustainable workflow implemented to construct knowledge graphs with research data extracted from Spanish university databases supported by the HERCULES information systems. Section 5 describes several use cases where our workflow has already been adopted, and Section 6 provides a set of lessons learned and recommendations. Section 7 outlines the main conclusions of the article and future work.

2. Related Work

In this section, we describe different (semantic) approaches that represent research and scientific data. We also present current solutions that allow the construction of knowledge graphs from a declarative perspective (i.e., using mapping rules) and related real use cases where this approach has been followed.

Many ontologies and vocabularies were defined to represent the scientific data domain. The most representative ontology for our use case is the VIVO ontology [13], which is focused on describing the academic domain. It represents the relations between researchers and academic assets such as projects, courses, papers, etc. The ontology is developed under the support of the VIVO project, which also allows creating academic portals using semantic web technologies. Another relevant ontology is the Bibliographic ontology (BIBO) [14], which aims to represent citations and bibliographic references, and is widely reused in other ontologies (e.g., VIVO). SPAR ontologies [15] are a family of ontologies that aim to provide full coverage for publishing and referencing scientific publications. Although there are many vocabularies [16,17] that model the research domain, the Hercules Ontology Network [11] is based mainly on VIVO and BIBO ontologies.

There are many domains and real use cases where declarative mapping rules are used to construct knowledge graphs. Most of them are supported by the W3C standard, R2RML [18], or its well-known extension for data beyond relational databases, RML [19]. In [20] the authors present a framework for integrating Bosch company manufacturing data as a virtual knowledge graph. They propose the use of Ontop [21] as the engine that translates SPARQL queries into SQL through OBDA mappings (Ontop mapping rules are equivalent to R2RML mapping rules [22]). Although it provides a general framework for creating virtual knowledge graphs from relational databases, the mapping rules are quite simple (around 53 rules), and details on how they have been generated or their

sustainability are not described in depth. Another successful use case in which mapping rules were used to construct knowledge graphs is reported in [23], where the European Union Agency for Railways (ERA) uses semantic web technologies to describe the European railway infrastructure. The knowledge graph is created from 28 heterogeneous data sources (a relational database and CSV files), and the mappings are declared using (YARR)RML [24]. The construction follows an ETL-based approach and uses the RMLMapper processor [25]. The authors mention that the generated mapping rules are "*the central resource for the ERA KG generation process*", and that they can be adapted, modified, or extended for new versions, but they do not specify any methodology or workflow to be followed. There are many other use cases [26,27] where mapping rules were used to construct (virtual) knowledge graphs, but their main focus is usually on optimizing the data integration process.

Despite the enormous effort of the community to provide a suite of mapping languages (e.g., the W3C recommendation R2RML [18] for RDB or its main extension RML [19]) and their corresponding optimized parsers (e.g., Ontop [21], Morph-KGC [28], SDM-RDFizer [29]), we realized that most of the current in-production KGs constructed from (semi)structured data sources [30–32] follow a scripting-based approach. In several use cases (e.g., Bio2RDF [33]), it has been demonstrated that this procedure negatively affects important features of the created KGs, such as maintenance, reproducibility, transparency, and data provenance. Hence, our suspicion is that there is a gap, not in the technological side (languages and engines), but in the methodology of the creation of the declarative mapping rules. In this paper, we present a sustainable methodology for declaring a knowledge graph construction process and its applications on a real industry-based example, showing the benefits of following a declarative approach together with a set of lessons learned.

3. Research-Performing Organizations in Spain: The Hercules Project

The HERCULES project was born to create a semantic layer on top of the research-performing organizations information systems to harmonize them and enhance data interchange among institutions and with the Ministry of Science and Education. In this section, we present an overview of the Hercules Ontology Network (ROH) [11], which has already been developed within the project, and which Spanish universities are encourage to be aligned with.

The Hercules Ontology Network

Figure 1 shows a graphical representation of the ROH classes and their main relations. We can observe that it models different entities, each of them represented by a color in Figure 1. For example, the entity research object is composed of a main class `roh:ResearchObject` and a set of subclasses such as `vivo:Dataset`, `bibo:Patent`, and `bibo:Document`. We provide a detailed explanation of each entity in the following points:

- **Project** entities describes information related to any business or science activity, which is carefully planned with milestones, work packages, risks, etc. Each project can be classified into different categories (private, national, european, etc.) depending on how it is funded, and it may also be part of another project.
- **Person** entities focus on representing information about researchers, based mainly on the `foaf:Person` class. The ontology extends this class to incorporate specific research information. For example, it includes data properties such as `roh:scopusID`, `roh:orcid`, and `vivo:researchId` and object properties such as `roh:hasRole`, `roh:hasCV`, and `roh:hasKnowledgeArea`.
- **Organization** entities include the general description of research-performing organizations (e.g., universities, research centers, etc.). Similarly to the person entity, an organization entity mainly extends `foaf:Organization`, including specific data and objects properties about this kind of institutions. In addition, it includes a subclass to represent data from organizations that are allowed to emit academic accreditations.
- **Funding** entities represent the information associated with the funding of a project or an organization. The entity presents a general class (`roh:Funding`) and a set of subclasses

(e.g., `roh:Grant`, `roh:Loan`). Each funding is divided into several `roh:FundingAmount`. Furthermore, the entity defines three other classes that are related between them: a `roh:FundingProgram` comes from a `roh:FundingSource`, which is supported by a `vivo:FundingOrganization`.

- **Research Object** entities aim to follow good open science practices, providing support to semantically represent all the results from projects (e.g., deliverables, reports, datasets), academic courses (e.g., Ph.D. or Master thesis), and other common research outputs (e.g., scientific papers, patents, etc.). It imports two main classes from the Information Artifact Ontology (OBO-IAO) [34] to represent in more detail any kind of research document, software repositories such as GitHub, Zenodo, or BitBucket, experimental protocols, or pieces of software.
- **Activity** entities focus on representing information about the actions or participations in events that researchers usually carry out during their career. In addition to the general class `roh:Activity`, the ontology includes a set of more specific classes such as `bibo:Conference`, `vivo:Internship`, and `vivo:InvitedTalk`.

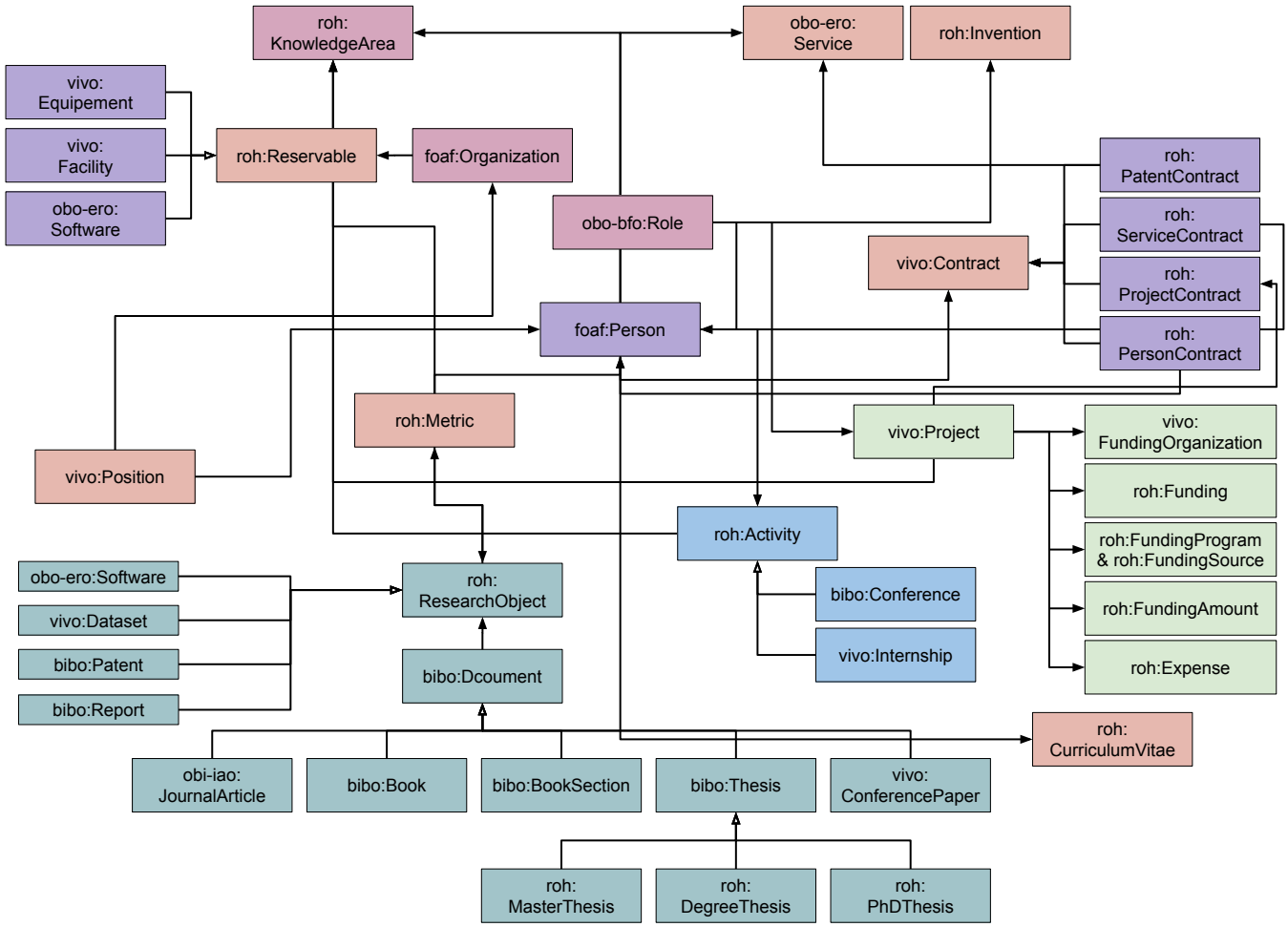

Figure 1. Overview of the ROH ontology with its main classes [11], adapted from the official ontology documentation. Arrows with a filled tip denote an inheritance relationship, while arrows with a nonfilled tip indicate an object property relationship. Dashed arrows denote temporal and geographical constraints. Each color represents classes that pertain to the same entity. Although this is not a standard notation for ontology diagrams, we have to use this diagram, as was the way the documentation was provided.

There are other entities and concepts defined in the ontology, such as metrics or curriculum vitae, but it is out of the scope for this paper to provide a full overview of all of them.

Apart from having a general overview of the entities and concepts that the ontology models, the main input that the knowledge engineer needs to construct the mappings is the documentation of the ontology. Currently, ROH documentation is generated following good practices in ontology publication: it uses Widoco [35], and it is openly available through a permanent identifier [36].

However, in the first versions of the ontology, when the knowledge graph construction process was carried out, the documentation was provided in a PDF file with tables. An example of how the documentation about the `roh:ResearchResult` class and its subclasses is provided is shown in Figure 2. This ad-hoc way of describing the ontology increases the difficulties of knowledge engineers to ensure the correctness of the mapping rules, hence compromising the quality of the constructed knowledge graph. For example, the documentation does not provide information about the meaning of the colors used in Figures 1 and 2, so the knowledge engineer must spend time to understand their meaning. We suspect that, at least in Figure 2, the colors are used to refer to an extension or reuse of classes already defined in previous ontologies (e.g., light blue is used for the VIVO ontology [37]). To mitigate this risk, instead of relying on the documentation, we decided to use the source of the ontology (i.e., the OWL file) to generate a preliminary set of mapping template rules with a semi-automatic mapping generation tool, as we will explain in more detail in Section 4.

Prefix Class	Class		Prefix	Object Property	Range Class	Prefix	Datatype Property	Range
roh	ResearchResult		roh	hasKnowledgeArea	skos:Concept	roh	identifier	
			roh	researchResultHasPart	roh: ExperimentalProtocol or roh:Repository or bibo: Document or ero: ERO_0000071 or vivo:Project	roh	title	
			roh	seqOfAuthors	rdf:Seq	roh	needsEthicalValidation	xsd:boolean
			roh	hasSucessor	roh:ResearchResult	vivo	abbreviation	rdfs:Literal
			roh	correspondingAuthor	foaf:Person	vivo	description	rdfs:Literal
			roh	spends	roh:ResearchObjectExprense	vivo	freeTextKeyword	
			roh	hasLicense	vivo:License			
			vivo	dateTimeInterval	vivo:DateTimeInterval			
			bibo	authorList	rdf:Seq			
	roh	ResearchObject	roh	correspondingOrganization	foaf:Organization	roh	language	
			roh	hasPartOfResearchResult	roh:ResearchResult			
			roh	producedBy	roh:Project			
			vivo	relates				

Figure 2. Excerpt from the ontology documentation for the `roh:ResearchResult` class and its subclasses, adapted from the official ontology documentation [11].

4. Sustainable Workflow for Constructing KGs

In this section, we describe in detail the process for creating a set of R2RML mapping rules compliant with the ROH ontology. We decide to choose R2RML, as it is the standard W3C recommendation to create mapping rules from relational databases. The input sources used are associated with the Universitas XXI database. However, the workflow presented here is domain independent; thus it can be applied or adapted to any other domain or database instance. We aim to create a sustainable and reasonable framework for constructing knowledge graphs in complex data scenarios.

Figure 3 presents the main steps of the workflow: (i) automatic generation of mapping templates; (ii) divide and conquer approach for systematic mapping; (iii) mapping template refinement; (iv) systematic filling of the rules; (v) validation of the mapping with domain experts in the loop; and (vi) RDF generation. In our use case, this procedure took us 6 persons-month to complete. In the following sections, we explain each step in more detail.

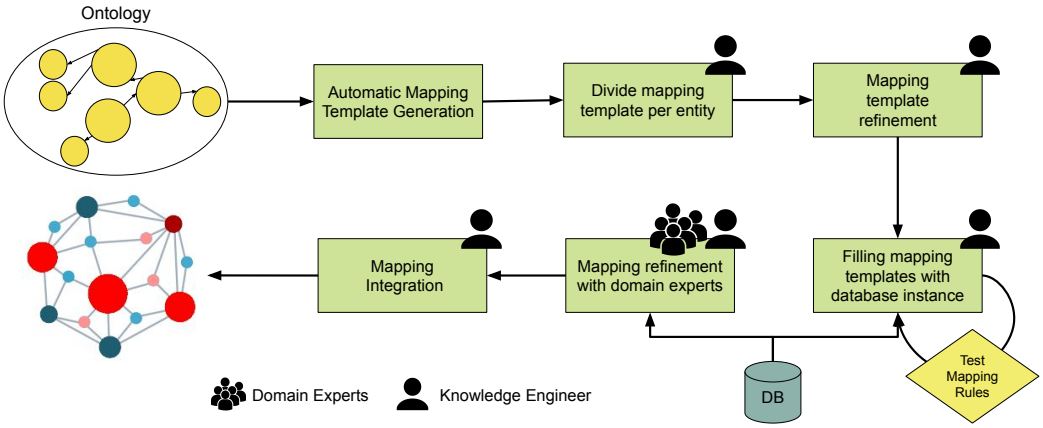

Figure 3. Workflow followed for declarative construction of knowledge graphs for research performing organizations.

4.1. Generation and Refinement of Mapping Templates

The first step of our workflow is to generate a set of mapping templates using the defined ontology as input. The requirements imposed to ourselves are: (i) the output mapping templates have to use a human-friendly format, and (ii) reduce to the minimum the manual and unreproducible work. We decided to use our tool OWL2YARRRML [38], which takes an OWL ontology as input and outputs a template mapping (without filling in the references to the data source) in YARRRML [24]. We show an example of a generated template in Listing 1. This tool implements the following rules:

1. For each ontology class (e.g., `vivo:Project`) an empty mapping is generated, with a common structure: one empty source, one subject map with a potential URI created following good practices on resource name strategy (We define it in a `rr:template` property, using the ontology URI base followed by the name of the class and an empty reference.), one predicate object map that indicates the class and subclasses of the entity.
2. For each data property associated to a class (i.e., the domain of the property is the class), a tuple of predicate object maps is created. It is composed by a predicate, which is the actual value of the data property, and an empty reference in the object. In addition, if the range of the property is defined with a datatype, a third value is added with the corresponding value. For example, in Listing 1 the property `roh:projectStatus` has the class `vivo:Project` in its domain and the datatype `xsd:integer` in the range, hence the corresponding mapping tuple is created.
3. Similar to the previous step, for each object property and its domain class (If the property has several classes the process is repeated), a reference predicate object map is created where the parent triples map is the actual triples map which defines the rules for the class in the range of the property. The conditions of the join remain empty in this step. In our example, the property `roh:produces` has the class `vivo:Project` as domain and `roh:ResearchObject` as range, so the corresponding rule is created.

After obtaining the first version of the mapping template, we decide to perform two relevant steps to improve the quality of the template and also to help during its filling process. The first step is to split the mapping document into multiple ones per entity. After the generation, the complete mapping contains more than 20,000 lines (in YARRML syntax) and more than 300 triples map. Filling the complete document in one go does not seem to be a good idea due to the amount of rules, and can negatively impact on the quality of the generated knowledge graph. Hence, we decide to split the full document into several independent mappings, including the rules for each entity (i.e., a mapping template for project, a mapping template for activity, etc.). We finalize this step with 21 mapping entities. The second step is to refine each mapping template. Manually, the knowledge engineer reviews each entity mapping to check for potential inconsistencies. Examples of these

inconsistencies are: modifying the subject URI to use another web domain, changing the datatype of an object to be more general, or adding the `rr:termType rr:IRI` for generating URIs. Additionally, detailed comments on the mapping are also posted to help during the next steps and during maintenance. Examples of these comments include: mentioning potential SKOS concept lists (as they could demand transformation functions over the input data), potential datatypes of an object that are not defined in the ontology, or decisions taken in the mapping declaration (e.g., instantiate all classes and subclasses of an entity to improve the SPARQL queries).

Listing 1: Automatic YARRRML template.

```
mappings:
ProjectTM:
sources:
- table:
s: universitas-kg:project/$()
po:
- [a, vivo:Project]
- [a, obo-bfo:BFO_0000015]
- [vivo:identifier, $()] #datatype
- [vivo:abbreviation, $()]
- [roh:title, $()]
- [roh:projectStatus, $(), xsd:integer]
- [roh:knowledgeArea, $()] #SKOS
- p: vivo:relates #absence?
o:
- mapping: audit
condition:
function: equal
parameters:
- [str1, $()]
- [str2, $()]
- p: vivo:participates
o:
- mapping: activity
condition:
function: equal
parameters:
- [str1, $()]
- [str2, $()]
- p: roh:produces
o:
- mapping: researchObject
condition:
function: equal
parameters:
- [str1, $()]
- [str2, $()]
```

4.2. Systematic Filling of Mapping Rules

The second step is to fill in the mapping templates with reference from the input data source (a relational database in our case). In order to follow a systematic procedure, we first analyze for each entity what the potential tables of the database are that may contain relevant information for the mapping. To do that, before starting the mapping process, we perform a preliminary analysis of the database to ensure that the most relevant concepts of the ontology were also represented in the tables. Then, we rely on the documentation provided by the tables (description in natural language) and the defined table clusters by the database administrators in the information system. These steps decrease the search space the knowledge engineer has to inspect for filling the mapping rules. For example, in this case, it does not make sense to inspect a table from the cluster "expenses" to fill the template of the person entity.

Once we limit the corresponding tables for each entity, we start the actual mapping step. The first step is to find the table or tables that contain the relevant information for each mapping, we again rely on the natural language description of the table and its columns for performing it. Then, the empty references of the template are filled with the column references. However, in several cases, the information needed to fully fill the rules needs to be adapted. We identify two main cases where the data needs to be pre-processed before the

construction of the knowledge graph. The first case is when the data needed for an entity is distributed in several tables and has to be defined in only one for a correct mapping. The second case is when the data need to be transformed before doing the conversion to RDF (e.g., for a SKOS list). For those cases, we decide not to increase the mapping complexity (i.e., making a SQL view directly in the mapping using the `rr:sqlQuery` property or using functions inside the mapping rules [39]), but we directly define an SQL script that creates that view. We decide to follow this approach for several reasons: (i) the views can be managed by database experts from the company without knowledge about semantic web technologies; (ii) we believe that it is more maintainable to define all views together in one SQL script than distributing them across the mapping rules; (iii) the built-in SQL functions are usually enough for the requirements of the data transformations [40]; and (iv) the declaration of functions within the mapping rules is not standardized yet and most engines do not support this feature.

When each mapping template is complete (see Listing 2), we perform an initial testing validation process. First, we use the YARRRML translator tool [41] to translate our YARRRML mapping into turtle syntax-based R2RML rules. If the engine throws an error, this means that there is a syntax issue in the mapping document that has to be resolved. After this step, we are ready to load the mapping into an R2RML engine. In our case, we rely on the Universitas XXI—Oracle DBMS, enabling its semantic capabilities, as it includes a layer for R2RML mappings (Please, note that the use of Oracle RDBMS as the backend database does not allow us to provide any empirical evaluation in this paper). In this task, we check, on the one hand, if the mapping rules have any error regarding misspelling table or column names (which is very common), and, on the other hand, if the generated RDF is the expected one. This last step means that we perform an initial and simple validation where we check if the RDF is generated according to the mapping rules by running simple SPARQL queries that ask for all resources of each class and their main properties (see Listing 3 as an example). For each entity, we create an independent view on Oracle to test them systematically.

Listing 2: YARRRML mapping filled.

```
ProjectTM:
sources:
- table: view_of_table1
s: universitas-kg:project/$(c1)
po:
- [a, vivo:Project]
- [a, obo-bfo:BFO_0000015]
- [vivo:identifier, $(c1)]
- [vivo:abbreviation, $(c2)]
- [roh:title, $(c3)]
- [roh:projectStatus, $(c4), xsd:integer]
- [roh:knowledgeArea, skos:$(c5)]
- p: vivo:participates
o:
- mapping: activity
condition:
function: equal
parameters:
- [str1, $(c7)]
- [str2, $(c1)]
- p: roh:produces
o:
- mapping: researchObject
condition:
function: equal
parameters:
- [str1, $(c8)]
- [str2, $(c1)]
```

Listing 3: SPARQL query for extracting basic information from a researcher.

```
PREFIX bibo: <http://w3id.org/roh/mirror/bibo#>
PREFIX roh: <http://w3id.org/roh#>
PREFIX vivo: <http://w3id.org/roh/mirror/vivo#>

SELECT DISTINCT * WHERE {
?researcher a foaf:Person .
?researcher roh:hasAccreditation ?acreditation .
?researcher roh:hasKnowledgeArea ?knowledge_area .
?researcher foaf:name ?name .
?researcher roh:participates ?participation .
?participation a ?partic_type .
?researcher roh:title ?acticity_title
FILTER (?partic_type = bibo:Conference  ?partic_type = vivo:InvitedTalk)
}
```

4.3. Validating the Mappings with Experts

One of the most important steps during the construction of a knowledge graph following a declarative way is to ensure that the declared relationships between the ontology and the input sources are correct. For that reason, once the mappings are completed and tested, we involved a set of domain experts in the loop from the company. It is important to mention that in addition to an in depth knowledge of the domain, it is desirable that these experts also have good technical knowledge (e.g., SQL), and they are able to query and work with the relational database system efficiently. Together with the knowledge engineer, they review and validate the mapping rules for each entity. The following activities are performed:

- **Semantic validation of mapping relations.** The domain experts validate that the relationships declared in the mappings between the concepts and the properties of the ontology are semantically equivalent to the references (tables and columns in this case) of the input sources.
- **Validation of the SQL views.** During the mapping process, the knowledge engineer creates a set of SQL views to transform and prepare the data in the RDB to generate the desirable knowledge graph. Domain experts review and validate these SQL views to ensure their correctness and that the transformations are also semantically equivalent (for example, in the SKOS lists).
- **Identification of missing references.** The experts help the knowledge engineer in identifying and filling missing references from the database to the ontology properties. This activity can follow a bottom-up approach, reviewing database references, and finding the correspondence in the ontology, or in the other way around. The knowledge engineer decides which approach to follow depending on the size and knowledge coverage of both resources.

At the end of this step, we can confirm that each mapping entity is ready to construct the complete knowledge graph.

4.4. Mapping Integration and KG Construction

The final step is to integrate all mapping entities into one complete document and construct the final knowledge graph. This is currently carried out through a manual step because we need to be sure that the references to the mapping identifier within the join conditions are the same for all entities. However, this process can be automated by implementing an integration tool. We finalize loading the complete mapping document into the Oracle database, generating more than 4 million triples from a sample database instance of a research performing organization (Due privacy reasons we cannot give more information about the research performing organization we use for testing our approach.).

5. Use Case: Universitas XXI

Universitas XXI is a Spanish company that provides technological support to more than 100 research-performing organizations worldwide. It has developed an information system that can be adapted to the requirements of each supported institution, but it is critical for the company to be compliant with the ROH ontology and provide that service to

organizations. Providing its database instance, the help to define the standard methodology presented in this paper, and they include it to construct their knowledge graphs (one per each supported organization). Therefore, the knowledge graphs constructed from the Universitas XXI information system are currently exploited in several real use cases. In this section, we provide an overview of all of them.

5.1. Feed the HERCULES Central Node

The HERCULES project configures a centralized node where all knowledge graphs that comply with its ontology can be integrated. Thus, the virtual knowledge graph created by each organization during the previous steps has been materialized into a native RDF dataset. Although at the moment of writing the SHACL shapes with the data constraints (that are claimed to have been generated in the context of the project) have not been provided by the creators of the ontology, it is supposed that they will be available soon, and hence the generated knowledge graphs will go through an additional validation step. This step will verify that the data we generated with our solution are compliant with the provided ontology and can be integrated in the central node of the project. The current validation that we perform is through a set of SPARQL queries provided by the official repository of the ontology [42].

5.2. Publishing the Knowledge Graphs through REST APIs

To bring RDF data closer to developers, a common procedure is to provide them through REST APIs [43,44]. This step would facilitate not only the exploitation of the knowledge graphs from the Universitas XXI developers, but also from any IT department of the supported universities, without the need of having specific technical knowledge on semantic web technologies. R4R [45], which has already been tested and used in several EU and national projects, is our own tool able to easily expose the data through REST APIs.

5.3. Exploiting Integrated Knowledge within the Organizations

Taking advantage of easily querying the integrated knowledge graph through ORACLE DBMS, Universitas XXI has developed a set of new features in its information system. Due to the use of declarative mappings to construct the KG, the rules are already integrated into the workflow of each Oracle database instance (one per organization). This allows database managers and developers to implement these new features once and deploy them to those organizations that want the services. In addition to an SPARQL endpoint, which is deployed by default, they provide new user interfaces for organizations to easily access the integrated knowledge.

6. Lessons Learned

In this section, we present a set of lessons learned extracted from the development of this methodology where declarative mapping rules play a key role. These recommendations summarize our experience of using standard semantic web technologies to systematically construct a knowledge graph in a complex domain such as research, but we hope that they will serve the community as general recommendations for any other domain.

- **Simple but useful support tools.** The construction of a knowledge graph in complex domains requires tools that support the creation of the rules and the management of complex data management tasks. Although there are solutions that aim to automatically create semantic annotations [46], they usually need a target KG (e.g., DBpedia or Wikidata) to create the actual instances from the input sources. However, most of the use cases have to fit the input data to a domain ontology, what demands manual work from a knowledge engineering for creating the mapping rules. We notice that simple tools such as OWL2YARRRML [38], the use of YARRRML [24] syntax instead of the common turtle-based syntax for the rules, or the deployment of virtual SPARQL endpoints per resource facilitates the creation and management of the rules and also guarantees their high quality and correctness.

- **Domain experts with technical knowledge in the loop.** One of the most relevant tasks during the construction of a knowledge graph is to ensure the correctness of the mapping (i.e., a column/field/register from the input source means exactly the same as a property/class of the ontology). The developer of the mapping rules (aka the knowledge engineer) knows very well the ontology and its structure, but in complex environments having a complete overview of the input sources is complicated, as documentation is not always sufficient. Involving domain experts that go beyond the knowledge of domain and have technical skills (for querying the database, understanding the tables and relations, etc.) is one of the key aspects to be successful during the mapping process. They help the knowledge engineer to understand the meaning of tables and columns as well as complex relationships and modeling decisions of the database.
- **Divide and Conquer.** Mapping has been been understood as an engineering task, but we believe that it is actually one of the most relevant and complex steps for constructing high quality domain knowledge graphs. For example, in our use case, we have a complex data integration problem where the ontology and the database have been developed completely independently, modeling a rich domain as it is research in different ways. First, it is needed to identify which of the two inputs overlaps the other, i.e., if the ontology covers more knowledge than the database or vice versa, and then a divide and conquer process can be followed to ensure a systematic mapping task. In our case, the database is the resource that covers more knowledge than the ontology, so we decided to split the mapping process by each class of the ontology. In this manner, we follow a systematic mapping process, ensuring that all the classes and properties from the ontology will be mapped.
- **Delegate complex tasks to the DBMS.** During the knowledge graph construction process there are many cases where the input data needs to be transformed or modified for obtaining the desirable structure in the generated RDF. This is usually the case of the SKOS lists, where the URIs defined in the thesaurus do not have a 1-1 mapping to the actual values of the database. Hence, some data transformation functions have to be applied. There are approaches that allow the declarative description of functions within mapping rules [39,47], that are interesting when the input source is not loaded in a database (i.e., raw files, APIS, etc). However, in cases that the input sources are supported by a DBMS it is better to create views and apply the transformation functions using the capabilities of the databases. On the one hand, delegating its application, we ensure an efficient execution of the transformation functions while we still maintain them in a declarative form. On the other hand, the domain experts and database managers are able to understand, manage, change and execute the created views, without adding more complexity to the mapping documents.
- **Sustainable procedures.** Either the ontology or the input sources can suffer changes after finishing the construction of the knowledge graph. This has a direct impact to the mapping rules, as they need to be adapted to new versions of the involved artifacts. In complex environments where the mapping rules can be huge (e.g., in our case the mapping document is defined by more than 5000 rules following N-Triples syntax), changing a property, a reference to a column or a class might be a difficult task. Defining common procedures and sustainable workflows to address these potential problems is also part of the construction of the knowledge graphs. The domain experts and database managers have to understand the mapping rules, its syntax but also its semantics, to be able to make these changes without the support of the knowledge engineer. Solutions such as YARRRML [24] that define the mapping rules following the YAML syntax or Mapeathor [48] that does the same but using Excel sheets, are good examples of how to use the general technical know how that developers, engineers, or database managers have for declaring the rules in a more human-friendly form.

7. Conclusions and Future Work

In this paper, we have described a methodology for constructing a knowledge graph and integrating several information over research-performing organizations. As a real use case, the methodology is supported by the HERCULES project, which aims to use semantic web technologies to improve data sharing between the current 79 universities in Spain. Together with the company Universitas XXI, we demonstrate the power of standard and declarative mapping rules for constructing knowledge graphs from any of these organizations. Taking advantage of the ontology network created by HERCULES, we define a systematic procedure to define the mapping rules, supported by well-known tools and a sustainable workflow, that can be applied as well in other domains to construct knowledge graphs. The created resources are used to generate the corresponding knowledge graph of each research performing organization that is supported by Universitas XXI. Additionally, each knowledge graph will be exploited by a REST API to allow an easy access for the IT developers of the research-performing organizations. Several new features are being implemented for accessing the integrated data, that will also help researchers and other stakeholders to improve several tasks such as generation of curriculum vitae, quality metrics calculation, etc.

The future work of this project includes the exploration of new techniques that (semi)automatize the creation of mapping rules [46], for example, by performing entity recognition or word embedding approaches [49] to find relations between the ontology documentation and the input database. From an engineering perspective, we plan to improve our set of tools that support this methodology. One of the most promising ideas will be to implement an engine able to find the variations between different versions of mapping templates, to easily identify which classes and properties are new or have been deprecated in the ontology. From a research perspective, it will also be interesting to analyze and propose solutions for efficient federated query processing over the distributed knowledge graphs of each organization [50], as the centralized approach proposed nowadays may not be sustainable in the near future. Finally, we also plan to enrich the knowledge graph with external research resources such as ORCID or the OpenAIRE research graph [51].

Author Contributions: Conceptualization, D.C.-F., O.C. and F.Y.; methodology, D.C.-F. and F.Y.; software, D.C.-F.; validation, D.C.-F., O.C., F.Y., R.M., J.O. and A.D.L.A.; investigation, D.C.-F. and O.C.; resources, R.M., J.O. and A.D.L.A.; data curation, F.Y., J.O. and A.D.L.A.; writing—original draft preparation, D.C.-F.; writing—review and editing, O.C.; visualization, D.C.-F.; supervision, R.M. and O.C.; project administration, D.C.-F., O.C. and R.M.; funding acquisition, D.C.-F. All authors have read and agreed to the published version of the manuscript.

Funding: This work was partially funded by Universitas XXI. David Chaves-Fraga is supported by the Spanish Minister of Universities (Ministerio de Universidades) and by the NextGenerationEU funds through the Margarita Salas postdoctoral fellowship.

Data Availability Statement: Not applicable. Due to privacy reasons, we cannot provide more information about the research performing organization we use for testing our approach.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analysis, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

Abbreviations

The following abbreviations are used in this manuscript:

KG	Knowledge Graph
RDF	Resource Description Framework
RML	RDF Mapping Language
SHACL	Shapes Constraint Language
SQL	Structured Query Language
ROH	Hercules Ontology Network

References

1. Asserson, A.; Jeffery, K.G.; Lopatenko, A. *CERIF: Past, Present and Future: An Overview*; Technical Report; euroCRIS: Kassel, Germany, 2002.
2. Hogan, A.; Blomqvist, E.; Cochez, M.; d'Amato, C.; Melo, G.d.; Gutierrez, C.; Kirrane, S.; Gayo, J.E.L.; Navigli, R.; Neumaier, S.; et al. Knowledge Graphs. *Synth. Lect. Data, Semant. Knowl.* **2021**, *12*, 1–257.
3. Belleau, F.; Nolin, M.A.; Tourigny, N.; Rigault, P.; Morissette, J. Bio2RDF: Towards a mashup to build bioinformatics knowledge systems. *J. Biomed. Inform.* **2008**, *41*, 706–716. [CrossRef] [PubMed]
4. Jaradeh, M.Y.; Oelen, A.; Farfar, K.E.; Prinz, M.; D'Souza, J.; Kismihók, G.; Stocker, M.; Auer, S. Open research knowledge graph: Next generation infrastructure for semantic scholarly knowledge. In Proceedings of the 10th International Conference on Knowledge Capture, Marina Del Rey, CA, USA, 19–21 November 2019; pp. 243–246.
5. Scrocca, M.; Comerio, M.; Carenini, A.; Celino, I. Turning transport data to comply with EU standards while enabling a multimodal transport knowledge graph. In Proceedings of the International Semantic Web Conference, Athens, Greece, 2–6 November 2020; Springer: Berlin/Heidelberg, Germany, 2020; pp. 411–429.
6. Google Knowledge Graph. Available online: https://developers.google.com/knowledge-graph (accessed on 3 October 2022).
7. Amazon Knowledge Graph. Available online: https://www.amazon.science/tag/knowledge-graphs (accessed on 3 October 2022).
8. Auer, S.; Bizer, C.; Kobilarov, G.; Lehmann, J.; Cyganiak, R.; Ives, Z. DBpedia: A nucleus for a web of open data. In *The Semantic Web*; Springer: Berlin/Heidelberg, Germany, 2007; pp. 722–735.
9. Vrandečić, D.; Krötzsch, M. Wikidata: A free collaborative knowledgebase. *Commun. ACM* **2014**, *57*, 78–85. [CrossRef]
10. Spanish Association of Universities (CRUE). Available online: https://www.crue.org/ (accessed on 3 October 2022).
11. Emaldi, M.; Puerta, M.; Buján, D.; López-de Ipiña, D.; Azcona, E.R.; Gayo, J.E.L.; Sota, E.; Maturana, R.A. ROH: Towards a highly usable and flexible knowledge model for the academic and research domains. *Semantic Web* 2022, *under review*.
12. Hercules Project—University of Murcia. Available online: https://www.um.es/en/web/hercules/inicio (accessed on 3 September 2022).
13. Corson-Rikert, J.; Mitchell, S.; Lowe, B.; Rejack, N.; Ding, Y.; Guo, C. The VIVO ontology. *Synthesis Lectures on Semantic Web: Theory and Technology*; Morgan and Claypool Publishers: San Rafael, CA, USA, 2012; p. 3.
14. Bibliographic Ontology (BIBO). Available online: https://bibliontology.com/ (accessed on 3 September 2022).
15. Peroni, S.; Shotton, D. The SPAR ontologies. In Proceedings of the International Semantic Web Conference, Monterey, CA, USA, 8–12 October 2018; Springer: Berlin/Heidelberg, Germany, 2018; pp. 119–136.
16. Sure, Y.; Bloehdorn, S.; Haase, P.; Hartmann, J.; Oberle, D. The SWRC ontology–semantic web for research communities. In Proceedings of the Portuguese Conference on Artificial Intelligence, Covilhã, Portugal, 5–8 December 2005; Springer: Berlin/Heidelberg, Germany, 2005; pp. 218–231.
17. Jeffery, K.; Houssos, N.; Jörg, B.; Asserson, A. Research information management: The CERIF approach. *Int. J. Metadata Semant. Ontol.* **2014**, *9*, 5–14. [CrossRef]
18. Das, S.; Sundara, S.; Cyganiak, R. R2RML: RDB to RDF Mapping Language. W3C Recommendation, W3C. 2012. Available online: http://www.w3.org/TR/r2rml/ (accessed on 15 September 2022).
19. 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 Ldow, Seoul, Korea, 8 April 2014.
20. Kalaycı, E.G.; Grangel González, I.; Lösch, F.; Xiao, G.; Kharlamov, E.; Calvanese, D. Semantic integration of Bosch manufacturing data using virtual knowledge graphs. In Proceedings of the International Semantic Web Conference, Athens, Greece, 2–6 November 2020; Springer: Berlin/Heidelberg, Germany, 2020; pp. 464–481.
21. Calvanese, D.; Cogrel, B.; Komla-Ebri, S.; Kontchakov, R.; Lanti, D.; Rezk, M.; Rodriguez-Muro, M.; Xiao, G. Ontop: Answering SPARQL queries over relational databases. *Semant. Web* **2017**, *8*, 471–487. [CrossRef]
22. Xiao, G.; Lanti, D.; Kontchakov, R.; Komla-Ebri, S.; Güzel-Kalaycı, E.; Ding, L.; Corman, J.; Cogrel, B.; Calvanese, D.; Botoeva, E. The virtual knowledge graph system ontop. In Proceedings of the International Semantic Web Conference, Athens, Greece, 2–6 November 2020; Springer: Berlin/Heidelberg, Germany, 2020; pp. 259–277.
23. Rojas, J.A.; Aguado, M.; Vasilopoulou, P.; Velitchkov, I.; Assche, D.V.; Colpaert, P.; Verborgh, R. Leveraging Semantic Technologies for Digital Interoperability in the European Railway Domain. In Proceedings of the International Semantic Web Conference, Virtual, 24–28 October 2021; Springer: Berlin/Heidelberg, Germany, 2021; pp. 648–664.
24. Heyvaert, P.; De Meester, B.; Dimou, A.; Verborgh, R. Declarative rules for linked data generation at your fingertips! In Proceedings of the European Semantic Web Conference, Anissaras, Greece, 3–7 June 2018; Springer: Berlin/Heidelberg, Germany, 2018; pp. 213–217.
25. RMLMapper Implementation. Available online: https://github.com/RMLio/rmlmapper-java (accessed on 1 October 2022).
26. Xiao, G.; Ding, L.; Cogrel, B.; Calvanese, D. Virtual knowledge graphs: An overview of systems and use cases. *Data Intell.* **2019**, *1*, 201–223. [CrossRef]
27. Chaves-Fraga, D.; Priyatna, F.; Santana-Pérez, I.; Corcho, O. Virtual statistics knowledge graph generation from CSV files. In *Emerging Topics in Semantic Technologies*; IOS Press: Washington, DC, USA, 2018; pp. 235–244.
28. Arenas-Guerrero, J.; Chaves-Fraga, D.; Toledo, J.; Pérez, M.S.; Corcho, O. Morph-KGC: Scalable Knowledge Graph Materialization with Mapping Partitions. *Semant. Web J.* **2022**. [CrossRef]

29. Iglesias, E.; Jozashoori, S.; Chaves-Fraga, D.; Collarana, D.; Vidal, M.E. SDM-RDFizer: An RML interpreter for the efficient creation of RDF knowledge graphs. In Proceedings of the Proceedings of the 29th ACM International Conference on Information & Knowledge Management, Virtual, 19–23 October 2020; pp. 3039–3046.
30. Heling, L.; Bensmann, F.; Zapilko, B.; Acosta, M.; Sure-Vetter, Y. Building knowledge graphs from survey data: A use case in the social sciences (extended version). In Proceedings of the European Semantic Web Conference, Portorož, Slovenia, 2–6 June 2019; Springer: Berlin/Heidelberg, Germany, 2019; pp. 285–299.
31. Liu, Z.; Shi, M.; Janowicz, K.; Regalia, B.; Delbecque, S.; Mai, G.; Zhu, R.; Hitzler, P. LD Connect: A Linked Data Portal for IOS Press Scientometrics. In Proceedings of the European Semantic Web Conference, Hersonissos, Greece, 29 May–2 June 2022; Springer: Berlin/Heidelberg, Germany, 2022; pp. 323–337.
32. Shen, Y.; Chen, Z.; Cheng, G.; Qu, Y. CKGG: A Chinese knowledge graph for high-school geography education and beyond. In Proceedings of the International Semantic Web Conference, Virtual, 24–28 October 2021; Springer: Berlin/Heidelberg, Germany, 2021; pp. 429–445.
33. Iglesias-Molina, A.; Chaves-Fraga, D.; Priyatna, F.; Corcho, O. Enhancing the Maintainability of the Bio2RDF Project Using Declarative Mappings. In Proceedings of the SWAT4HCLS, Edinburgh, UK, 9–12 December 2019; pp. 1–10.
34. Information Artifact Ontology (OBO-IAO). Available online: https://obofoundry.org/ontology/iao.html (accessed on 3 September 2022).
35. Garijo, D. WIDOCO: A wizard for documenting ontologies. In Proceedings of the International Semantic Web Conference, Vienna, Austria, 21–25 October 2017; Springer: Berlin/Heidelberg, Germany, 2017; pp. 94–102.
36. Hercules Ontology Network (ROH). Available online: http://w3id.org/roh/ (accessed on 13 October 2022).
37. Börner, K.; Conlon, M.; Corson-Rikert, J.; Ding, Y. VIVO: A semantic approach to scholarly networking and discovery. *Synth. Lect. Semant. Web Theory Technol.* **2012**, *7*, 1–178.
38. Chaves-Fraga, D. oeg-upm/owl2yarrrml. 2022. Available online: https://doi.org/10.5281/zenodo.5603173 (accessed on 13 June 2022).
39. Meester, B.D.; Maroy, W.; Dimou, A.; Verborgh, R.; Mannens, E. Declarative Data Transformations for Linked Data Generation: The Case of DBpedia. In Proceedings of the European Semantic Web Conference, Portorož, Slovenia, 28 May–1 June 2019; Springer: Berlin/Heidelberg, Germany, 2017; pp. 33–48.
40. Chaves-Fraga, D.; Ruckhaus, E.; Priyatna, F.; Vidal, M.E.; Corcho, O. Enhancing virtual ontology based access over tabular data with Morph-CSV. *Semant. Web* **2021**, *12*, 869–902. [CrossRef]
41. Chaves, D.; LuisLopezPi.; Doña, D.; Guerrero, J.A.; Corcho, O. oeg-upm/yarrrml-translator. 2022. Available online: http://dx.doi.org/10.5281/zenodo.7024500 (accessed on 10 October 2022). [CrossRef]
42. Hercules Ontology Network Compentency Questions. Available online: https://github.com/HerculesCRUE/ROH/tree/main/validation-questions/sparql-query (accessed on 3 September 2022).
43. Espinoza-Arias, P.; Garijo, D.; Corcho, O. Crossing the chasm between ontology engineering and application development: A survey. *J. Web Semant.* **2021**, *70*, 100655. [CrossRef]
44. Meroño-Peñuela, A.; Lisena, P.; Martínez-Ortiz, C. Web Data APIs for Knowledge Graphs: Easing Access to Semantic Data for Application Developers. *Synth. Lect. Data Semant. Knowl.* **2021**, *12*, 1–118.
45. Badenes-Olmedo, C.; Espinoza-Arias, P.; Corcho, O. R4R: Template-based REST API Framework for RDF Knowledge Graphs. In Proceedings of the ISWC (Demos/Industry), Virtual, 24–28 October 2021.
46. Chaves-Fraga, D.; Dimou, A. Declarative Description of Knowledge Graphs Construction Automation: Status & Challenges. In Proceedings of the 3rd International Workshop on Knowledge Graph Construction, Crete, Greek, 30 May 2022.
47. Jozashoori, S.; Chaves-Fraga, D.; Iglesias, E.; Vidal, M.E.; Corcho, O. Funmap: Efficient Execution of Functional Mappings for Knowledge Graph Creation. In Proceedings of the International Semantic Web Conference, Athens, Greece, 2–6 November 2020; Springer: Berlin/Heidelberg, Germany, 2020; pp. 276–293.
48. Iglesias-Molina, A.; Pozo-Gilo, L.; Dona, D.; Ruckhaus, E.; Chaves-Fraga, D.; Corcho, O. Mapeathor: Simplifying the specification of declarative rules for knowledge graph construction. In Proceedings of the ISWC (Demos/Industry), Virtual, 1–6 November 2020.
49. Brunner, U.; Stockinger, K. Entity matching with transformer architectures-a step forward in data integration. In Proceedings of the International Conference on Extending Database Technology, Copenhagen, Denmark, 30 March–2 April 2020.
50. Heling, L.; Acosta, M. Federated SPARQL Query Processing over Heterogeneous Linked Data Fragments. In Proceedings of the ACM Web Conference 2022, Virtual, 25–29 April 2022; pp. 1047–1057.
51. Manghi, P.; Bardi, A.; Atzori, C.; Baglioni, M.; Manola, N.; Schirrwagen, J.; Principe, P.; Artini, M.; Becker, A.; De Bonis, M.; et al. The OpenAIRE research graph data model. *Zenodo* **2019**. [CrossRef]