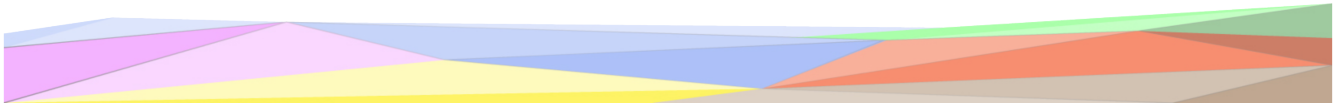
UNIVERSITY
OF TRENTO - Italy

Dipartimento di Ingegneria e Scienza dell'Informazione

– KnowDive Group –

KGE 2022 - Metadata Project Report

Document Data:	Reference Persons:
February 21, 2023	Pelagalli Camilla, Alessandro Salvatore Raho



Index:

Revision History:

Revision	Date	Author	Description of Changes
0.1	20.04.2020	Fausto Giunchiglia	Document created
0.2	13.11.2022	Pelagalli, Raho	Inception Phase added
0.3	13.11.2022	Pelagalli, Raho	Inception Backtracking(On Going) and Informal Modelling
0.4	07.12.2022	Pelagalli, Raho	Formal Modeling
1.0	20.02.2023	Pelagalli, Raho	Final Document

1 Introduction

1.1 The revealing nature of metadata

The necessity of a *meta language* for computer systems was addressed by Stuart McIntosh and David Griffel in the ADMINS (Administrative Data Methods For Information Naming Systems) progress report of the MIT Center for International Studies, written in 1967 [GM67]. They referred to an *object language* about *subject descriptions* of data and token codes for the data, that should fit the changing purposes of its users.

The potential recursion embodied in the current mainstream definition of metadata as "data about data" already arose as they underlined that they had "a language for pointing at actual data which is also in a language" [GM67].

In order to help human users and computer applications to understand digitized data, the concept of *standard* became a complementary aspect to metadata as boundaries in the *content*, *context* and *structure* of a data object, being it in physical or intellectual form, had to be set to reveal its nature and not to hide it because of the confusion generated by the language and semantic heterogeneity used to describe it.

Content is intended as what is intrinsic to an information object, its essence, not in relation with others. *Context* places the data object as a dependent part in a whole environment, primarily defined by the time and space categories: all the object extrinsic qualities have to be enlighten as metadata should take into account the relationships that the data object establishes with surrounding elements and the transformations that it undergoes as a consequence. *Structure* grounds the data object in the environment with a formal set of associations that enables supervised interaction between data objects over a network and between the user and the data they are searching [Bac16].

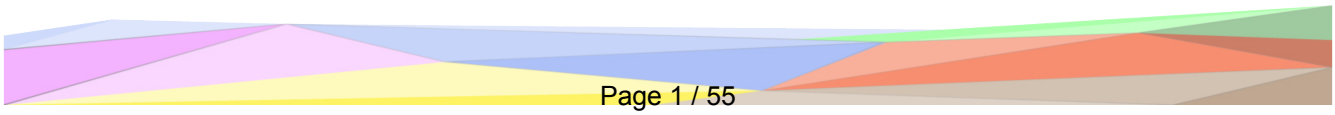
Data digitization does not imply access, but standardized metadata creation does, making a data object findable and thus potentially useful. Metadata creation should be a collaborative effort to provide end-users reliable and authoritative resources, persistent and preserved in all their informational layers, that could be reused and repurposed [Bac16] [Giu+22].

Jason Scott informally states that Metadata "you see, is really a love note – it might be to yourself, but in fact it's a love note to the person after you, or the machine after you, where you've saved someone that amount of time to find something by telling them what this thing is." (`http://ascii.textfiles.com/archives/3181`).

Describing to a user what a data object actually is without ambiguity would save them time that could be spent in the developing of tools that make a meaningful use of the provided data, paving the way of innovation in Data Science.

1.2 The exploratory nature of Knowledge Graphs

Metadata can be used to describe both data objects and the relationships among them. The resulting network of interactions can be explored through *Knowledge Graphs*, which have a bi-layered structure representing real world objects from both a knowledge and a data point of view, as entity types (*Etypes*) and entity representations (*entities*) respectively.



The real world properties used to denote an object can also be viewed from a KG as actual *property values* and mapped to the underlying data and object properties, where *data properties* consider the object singularly and *object properties* depict the intervening relationships between different objects of a real world domain, taking into account the possibility of self-loop interactions of an object with itself [Giu22].

Knowledge graphs leverage the full potential of graph theory and ER modeling, and are further enriched by the *iTelos* methodology, followed in this report to develop the KGE *Metadata project*. The final *Entity Type Graph*, or ETG, embeds three additional informational layers, ontology, teleology and language, to reproduce a digitized replica of a real world object and its surrounding microcosm [Giu+21].

The KGE Metadata project documentation plays an important role in order to enhance the reusabiltiy of the resources handled and produced during the process. A clear description of the resources and the process developed, provides a clear understanding of the KGE project, thus serving such an information to external readers in order to exploit that in new projects.

The current document aims to provide a detailed report of the KGE Metadata project developed following the iTelos methodology. The report is structured, to describe:

- Section 2: The project's purpose and the domain of interest and the resources involved (both schema and data resources) in the integration process.
- Section 3: The input resources considered by the KGE project.
- Section 4, 5, 6, 7: The integration process along the different iTelos phases, respectively.
- Section 8: How the result of the KGE process (the KG) can be exploited.
- Section 9: Conclusions and open issues summary.

2 Purpose and Domain of Interest (DoI)

Our project context is an **Open Data environment**, where anyone is free to use, re-use or redistribute resources - data or contents - while preserving provenance and openness.

Data Scientia is the concrete representation of this environment: an under-costruction web portal that allows users to explore different kind of datasets. The portal contains two main web pages for now:

- **Homepage**: the research group's Vision and Mission are outlined and the main initiatives are globally described and referenced, such as UKC, SCroLL, Digital University and Smart University (http://datascientia.disi.unitn.it/).
- **Liveschema**: a catalog of knowledge resources, related to single projects. The user can retrieve different resources singularly, by searching inside Catalogs or by browsing datasets directly. There are 7 Catalogs (DERI Vocabularies, FINnish Thesaurus and Ontology service, GitHub Repository, KnowDive Vocabularies, Linked Open Vocabulary, Research Vocabularies Australia and User defined Datasets) and 958 Datasets. The user can filter

datasets by Providers, Tags, Formats and Licenses. Available Services are: Dataset uploader, Knowledge embedder, FCA generator, Cue generator, Visualization generator and Query Catalog (http://liveschema.eu/).

Data Scientia's main goal is to gather all **projects** that result from Knowledge Graph Engineering (KGE) processes following the iTelos methodology ([**https://doi.org/10.48550/arxiv.2105.09418**]). Each project can contain multiple datasets, each belonging to a different informational layer. For each project, three main layers should be identified:

- **Language Layer**: any linguistic resource - property or dataset about languages and terms used - should be explicitly referenced to increase re-usability.
- **Knowledge Layer**: any Knowledge Base encoding information about KG schemas (etypes and properties). If detailed enough, this layer could be decomposed in its core elements: ontology and teleology, thus giving the user an unprecedented insight into the conceptual and relational data shaping behind the actual data values representation.
- **Data layer**: any dataset which consists of data in some format, instantiating the KG's structure (entities and attributes). If users are interested in expanding data resources, they can use the already existing schemas as meaningful anchors, without loosing data quality.

Data Scientia users should have access to both projects and their layers and decide what to re-use, modify or produce to meet the purpose they have in mind. Projects should be linked based on their similarities on a general and layered basis.

Metadata play a crucial role in this context because they describe datasets and represent the connections between them, making resources findable and addressable without any effort. For example, users looking for healthcare ontologies should be able to access them directly, without having to go through whole projects at the risk of stepping away from their purpose. Metadata reduces search time and space: they allow users to skip to what is essential to them, making the Data Scientia successful and meaningful.

Our project **purpose** can now be stated unambiguously:

*A KG supporting the **Data Scientia** web portal's users to find the most suitable **resource** for their needs, as well as all the resources **linked** by the one searched.*

To achieve the project objective, we have analyzed existing KGE Projects and Data Scientia resources and projects, which were our **data resources**. We have extracted the fundamental layered view of datasets from KGE Projects and we have outlined Data Scientia main entity types, along with their properties, by carefully examining the web portal and through a Data Scientia top level description and requirements given to us by the domain expert. The latter helped us understanding the preliminary boundaries of our **schema resources**, by stating the present structure of the portal with its shortcomings: the absence of linked resources and informational layers. To overcome them, we have designed a possible and plausible future structure for the Data Scientia web portal, by defining its Knowledge Graph.

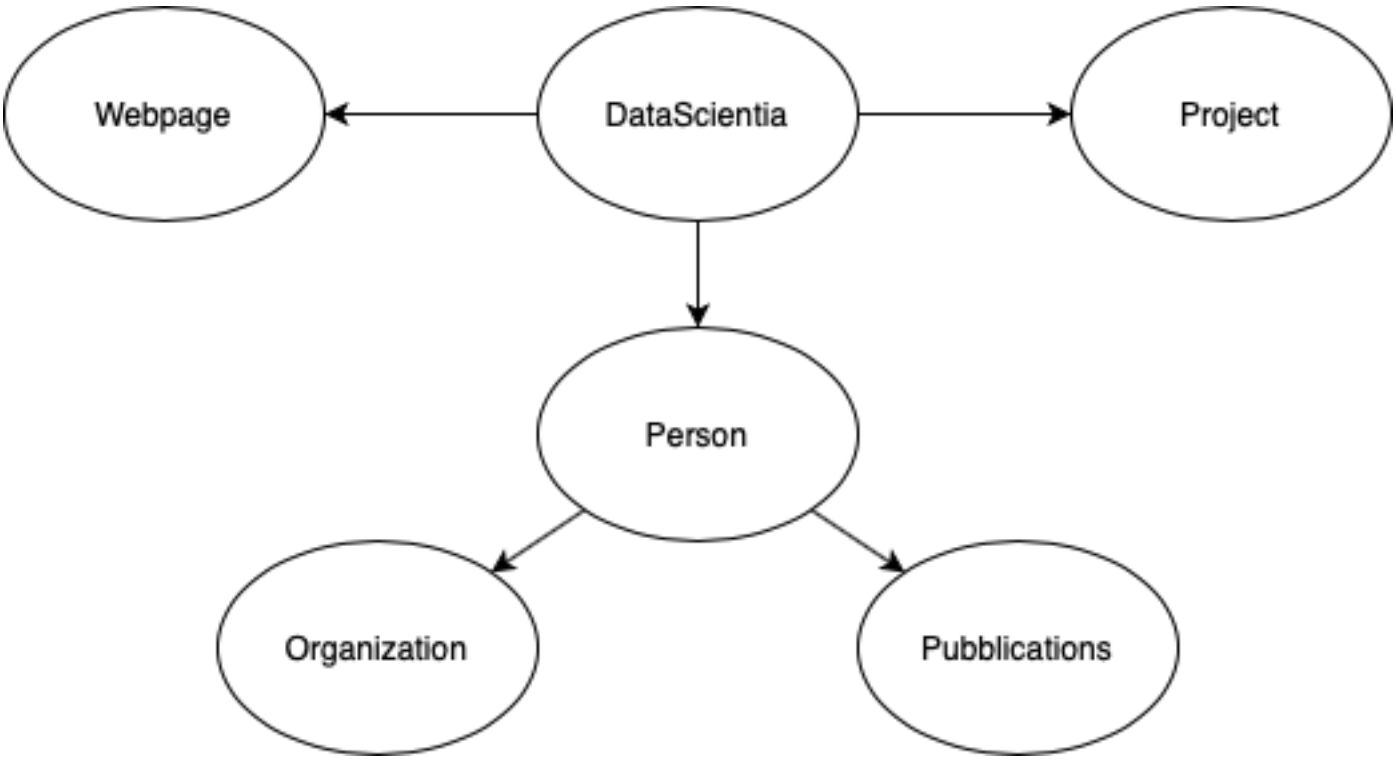


Figure 1: DataScientia Structure

3 Data Sources

As a starting point for our **Knowledge sources**, we have discussed Data Scientia inner structure with the domain expert. As shown in Figure 1, the resulting basic elements of the portal are the following:

- **Data Scientia**: the root portal, containing all the resources we need to manage, manipulate and integrate to meet the external user's needs and requirements.
- **WebPage**: the user-friendly interface(s) where resources are loaded and user can make queries and browse available services.
- **Person**: Data Scientia is a community and community members are assumed to interact directly with the web portal. A Person entity type is crucial to understand the boundaries between different roles of an individual with respect to the portal. The user should also have an easy way to find information about community members, partners, authors, researchers and so on. For example, these personas need to be described through the Organization they work for, or their publications. From now on we will name this group of people Data Scientia community.
- **Project**: Projects can be uploaded to - or downloaded from - the portal and they also need to be decomposable and recomposable as KGE processes, according to their informational layers.

Starting from this draft, since we currently do not have access to any metadata datasets, we have collected data resources from Data Scientia already existing web pages and thought about which kind of new web resources would be useful to add or integrate. We have read the documentation about existing projects, uploaded to both the Data Scientia portal and the liveschema catalog, along with KGE projects which followed the iTelos methodology. We have explored their components carefully to extract the fine-grained metadata used to create the property columns of the attached e-types labels. We have also focused on which metadata are used in DataScientia to depict people.

4 Purpose Formalisation

Giulia Pelagalli is a PhD student in civil engineering at University of Trento and she is doing a research study on Coronavirus disease (COVID-19) modes of transmission and spreading through public transportation. Giulia is searching for a specific data resource about urban and suburban transportation in Trentino. How can the Data Scientia web portal help her accomplish this task? From the catalog LiveData she can browse all available data resources. Being each dataset associated with its theme and description, she can easily understand if these resources are compliant with her needs. She can also know right away the dataset language, format and license without having to scan the dataset itself row by row. She can save time and space in her computer: she does not have to download the resource to inspect it: she has access to the most useful information about the specific dataset through its metadata. Giulia can filter data resources according to their geography: each dataset has its geographical metadata; if she finds a resource for Trentino, she may be interested in a dataset which provides the same information, but for a different region, e.g Marche, for the sake of comparison. Also, if she wants to go beyond the data resource, she can see if there is/are any knowledge and/or language resources attached to it. In this way, she can analyse the underlying schema that gives structure and meaning to data and check if the language resources can be re-used in her research study. Giulia can check if there are existing projects tackling her same domain, by performing a different kind of search in the web portal: she can look at complete projects with a purpose similar to hers: these projects contain all the informational layers and could be referenced by Giulia in her study; she could also browse them to see their structure, documentation and datasets, if present. Note that datasets are available in two different ways/views. First as datasets belonging to the same layer, through each corresponding catalog (LiveData, LiveKnowledge, LiveLanguage): she can use a bottom-up approach to combine these datasets into a Knowledge Graph for her study. Second, from a top-down point of view, datasets could be used as parts, and layers, of complex projects or KGs.

Alessandra Castricini is a final year student of Computer Science at University of Trento and she is currently working on her Bachelor's degree thesis under the supervision of professor Fausto Giunchiglia. She is a big data enthusiast, so she is performing an investigation of innovative and emerging tools to track students' routines and everyday life situations. Can Data Scientia provide her with any resource or existing literature about this topic?

From the catalog Live People she could browse a large amount of collected data in the field

of human behavior and social interactions. Live People data are kept in a private repository for privacy reasons: she could gain access since she is gathering data for study purposes. Each available dataset could be explored through its metadata: title, creation, layer and especially linked language, knowledge and data resources. The layer describing a dataset can be one or multiple: Data and/or Language and/or Knowledge, so Alessandra can immediately know if the dataset she is looking at contains actual data, schema representations or vocabularies. If the dataset deals only with one layer, for example students' routine etypes definition, she can check if there are related resources belonging to the data layer through the Data Resource metadata: there might be already existing datasets which contain diachronic data about the everyday life of university students over a period of time. Since Alessandra's thesis might refer to existing literature or researches as theoretical background, she might want to explore existing study projects which share her same purpose: Data Scientia would allow her to look at study projects characterized by the presence of the metadata documentation. The latter would redirect Alessandra to the scientific development and explanation of the project, helping her both with theory and bibliography construction: many useful papers and authors would be made accessible to her. Also, she could explore the Data Scientia web portal looking for people that work on the KnowDive research group, of which his thesis' supervisor is the founding member, to look at their publications; they might also be included in study projects in the author metadata.

Salek Mali is a data intermediary and works on the Google translate service. He is gathering as many language resources as possible to address syntactic and/or semantic heterogeneity across different languages. He does not have a specific purpose, he just wants to collect data about language, possibly with an open access and license. What resources can the Data Scientia web portal offer him?

From the catalog Live Language, Salek can inspect many different language resources. A language resource is a dataset containing words and terms of a specific language for a specific topic. He can filter resources based on their language or their area of interest.

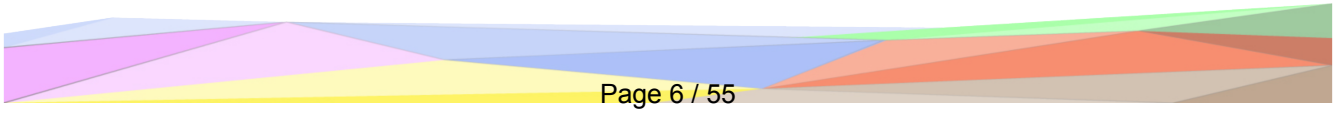
Eugene Jones is a Knowledge Engineer working for European Commission. He want to produce a European standard for the creation of resources about student activities and routines.

He can both explore Live People and Live Knowledge in order to find useful schema resources to describe at best how students in different European regions spend their time. More in details, Live People offers many works and researches from the past about people and their habits. Moreover it is possible to find both a schema resource (Knowledge layer) often paired with factual data usable with that specific schema. Meanwhile Live Knowledge is a catalog containing only schemas that can be reused for new needs by the Data Scientia users. In both cases it will be possible to filter resources among their theme.

Alfredo DalPra is an highschool teacher and in it's free time he provides to some curios students some introductory lectures about Data Analysis and Management. He is looking for simple dataset (not purpose-related) that can be used as small exercise for his evening class.

Thanks to the Live Data Catalog, Alfredo is able to have a look to just the data with the needed distribution format (csv) that can be easily read and understood by students.

Carlotta Neri is a KGE student. She have to build a knowledge graph for the course project based on the purpose she was given: healthcare in Trento. For this, she need to find as many datasets medicine-related as she can in order to understand what are the resource available to create the structure of a new reusable model.



From Data Scientia she will be able to find all different type of data related to Trento and his healthcare infrastructures aiming to learn what are the available information. Starting form the found data, she can also find related schema and languages that can be taken into consideration when considering the specific purpose. She will moreover be able to follow the guided built-in iTelos metodology structure on the web site being able to build step-by-step her project.

5 Inception

Clarified our purpose and needs for this project, we can now start the Inception Phase.

As said in 3, there aren't already available datasets and tables able to give a detailed meta-data description of the e-types we identified in the previous phase. Due to this lack of starting information, we decided to build our own data. This was for us the chance to explore in depth the definition of each single object we needed, analyzing for each of them their relevance in terms of the purpose we have to accomplish. We formalized many different architectures refining each time all our inner structure obtaining what, according also to the domain expert, is one of the best way to describe the skeleton of our KGE. We identified the need of seven tables to cover all the objects we want to outline.

5.1 Tables in depth description

Dataset is our first table. For us datasets are any collections of data. They can be described with the following properties: Dataset_ID, a unique id identifying a specific dataset, Title, a name for the repository, Theme, general topic of the collected data, Description, in-depth specification about the dataset it self, Creation_Date, the day in which the dataset was created, Format, , list of all the format of the file contained in the repository (txt, owl, ...), Language, the language of the data contained in the dataset, Geography, describing the geographic area interested in the data collection, License, information about the copyright of the resource, Web_Page, the URL of the page were data are stored, Layer, our distinctive property that highlight which level of data are available in the resource (among Language knowledge and data), Language_Resource, Knowledge_Resource and Data_Resource. The last three, are basically pointers to resource of a specific layer that can be used with the dataset described. Dataset are produced by an Author, are used in some Projects and belongs to a specific Catalog.

Our second table is Catalog. Catalogs are considered as a container of Datasets. They are 4 and are described with a Catalog_ID, a unique id identifying a specific Catalog, Title, a name for the repository, Description, in-depth specification about the catalog it self, Creation_Date, the day in which the catalog was created, Language, the language of the data contained in the catalog, License, information about the copyright of the resource, Web_Page, the URL of the catalog homepage.

Moving forward we are now describing the Person Table. People can be seen as Author of Dataset and/or Project instances as well as an integral part of Data Scientia. The community is one of the most important factors in the Data Scientia development, but according to our

domanin expert, for now we won't expand this aspect for now. A Person is defined with the following metadata property: ID, a unique id identifying a single person ,First_Name, Last_Name, Affiliation, underlining a person belonging to a company or a University, Role, such as student or researcher, Pubblication, the list of the specific person, and Contact, such as a direct e-mail.

Project is a peculiar e-type because we felt the need to distinguish among three project types: KGE Project, (distinctive property: Layer) including all the student projects produced in the Knowledge Graph Engineering course offered at University of Trento, Application Project, (distinctive property: Programming_Language and Release) composed by all the project that produced a sort of application for their purpose, and Study project, (distinctive property: Study_Documentation) that involve all the rest. The shared property are: Project_ID, a unique id identifying a specific project, Title, a name for the Project, Purpose, general topic of the research, Description, in-depth specification about the project it self, Creation_Date, the day in which the project was created, Last_Update, the day in which the project was updated the last time, Completion_Date, the day in which the project ended, Access,a boolean indicating if the work is public or need some sort of authentication, Language, the language of the data contained in the project, License, information about the copyright of the resource, Web_Page, the URL of the page were data are stored.

Data_Scientia is our last label. Data Scientia is the container of almost all our other e-types and can be described using the 'Webpage' property, containing the URL of the Data Scientia web portal homepage, along with the ID of the objects that are present in the portal: Catalogs, Projects and Person.

We identified that:

- Dataset, Catalog and Project are core entities since are the basic unity of our project and are well described with standards. However Dataset needs contextual property in order to accomplish our project's purpose.
- Person is a Common entity, since we just use standard metadata property to describing them
- Data_Scientia is our only contextual entity since it's definition is peculiar and unique with respect to other common web portal definitions we cn find in standards

5.2 Tables Metadata

Using SHAPEness Metadata Editor, we have generated the metadata associated to each data table corresponding to the above described etypes. Since we have collected and partially produced data values, we are addressed, with the object property dct:publisher, as epos-dcat-ap:Agents of each Dataset (Figure 2). We have defined 7 epos-dcat-ap:Datasets with their attached epos-dcat-ap Distributions, employing the object property dcat:distribution. Other than the mandatory properties we tried to fill at least 8 recommended properties for Dataset (dcat:distribution, dct:publisher, dcat:keyword, dct:modified, dct:created, dct:issued, dct:accessRights, dct:language), 7 recommended properties for Distribution (dct:format, dcat:downloadURL, dct:language, dct:license, dct:modified, dct:title, dct:issued) 1 recommended property dct:type) and 1 optional property (foaf:name) for Agent. To further describe

Agent the dct:type data property is used in combination with the epos-dcat-ap:Concept (Data Resources Creator and Publisher), linked with the skos:inScheme object property to epos-dcat-ap:ConceptScheme (Data intermediary).

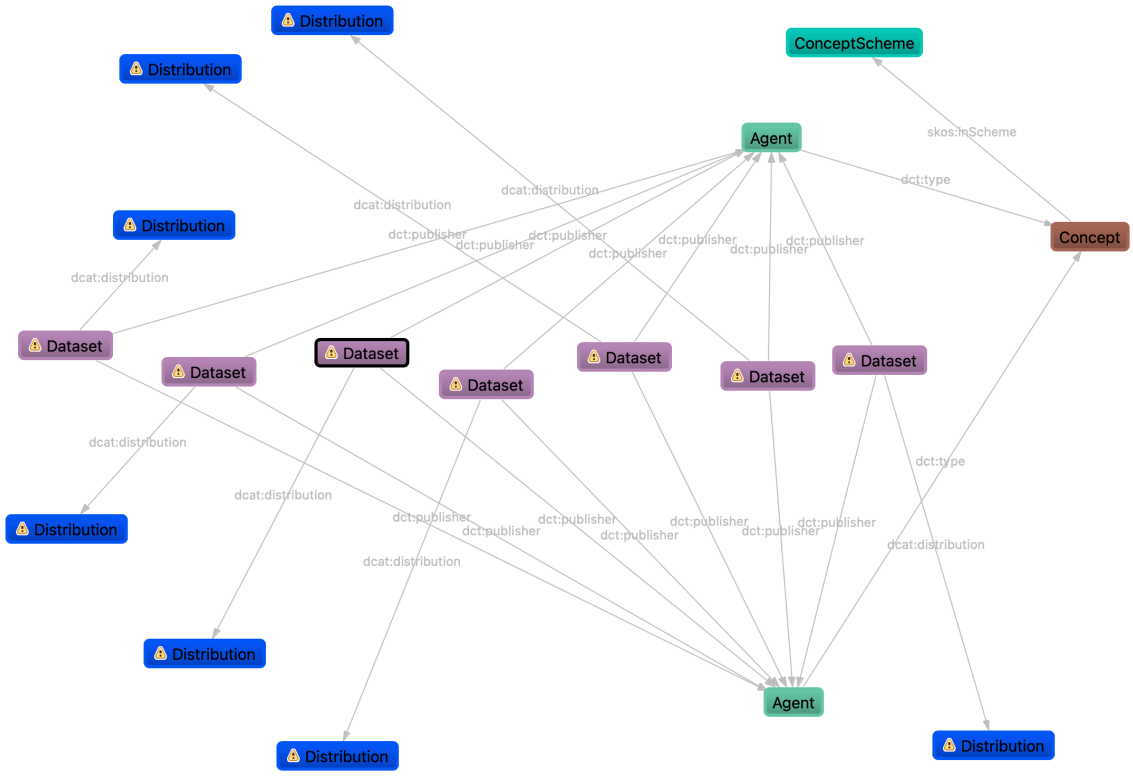


Figure 2: Project Metadata.

5.3 Schema

Having this e-types definition is pretty easy to identify a basic schema that stand beside the project's objects. It is reported in 3

As briefly explained above, we consider as relevant this e-types bond: Data Scientia for us is a container for almost all the other existing objects. It is a composition of 4 Catalogs (Live People, Live Data, Live Language and Live Knowledge), Projects and Person, that works on the portal it self and in the future will became the Data Scientia community, a group of people able to exchanging resources of different layers. Dataset is instead our atomic entity representing the smallest object defined in our work composing both Catalog and Project. Person in this specific representation stands mainly as 'Author-of-Things': they crated Dataset, Project, Catalog and also the Data Scientia portal.

All this bonds are seen as edge in our schema.

Resources formatting (semi-formal transformation) Defined our schema we developed it using Protegé creating a semi-formal version of it. As visible in 4 we created our common core and contextual entities as sub-classes of Thing all on the same level of depth. We set to true their

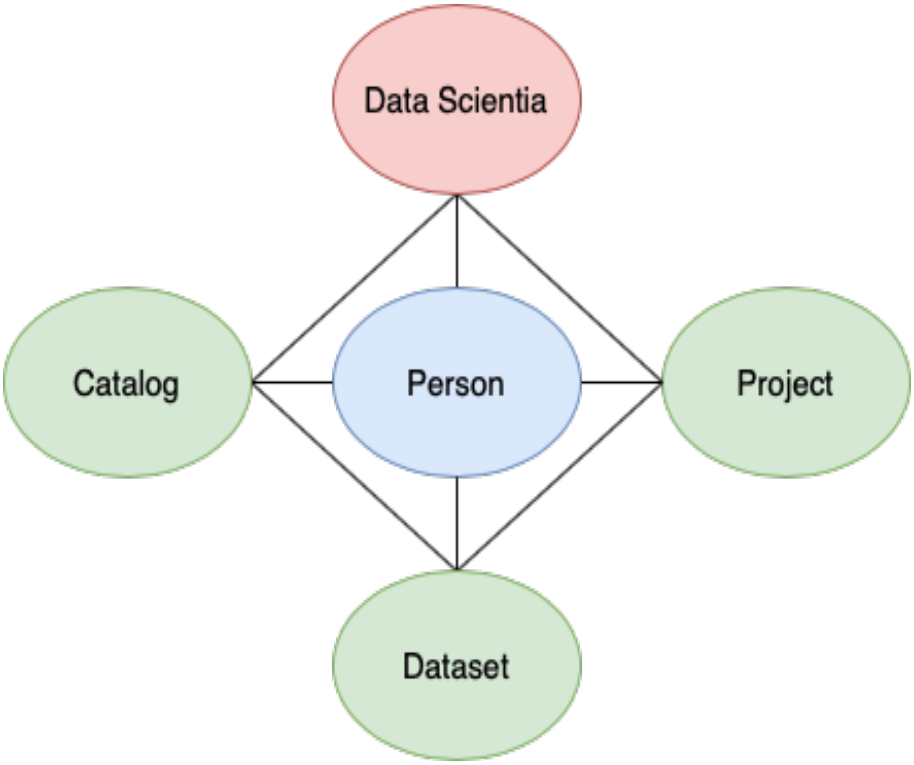


Figure 3: Etype Mapping

e-type annotation and developed all their data property. We than implemented the entity-entity connection as object property specifying for each one domain and range (5). The protegé will be uploaded on our github repository as an owl file.

As final part of the paragraph we'd like to add a small description of other possible schema definition we found and can be eventually chosen.

Inspecting the existing catalogs we found that Dataset can be divided in two subcategories: Dataset Single and Dataset Composed. In fact we found that in some circumstances (such as researches uploaded in the catalog Live People) we can have multilayered dataset. This means that a single dataset contains data associated to two or more different layers. This is our definition of Dataset composed. In the meantime we defined Dataset single as a data resource containing resources associated to just one informational layer. Is important to highlight that we call a dataset Dataset Single also if contains many different resources all associated with the same layer (it does not need to be composed by just one resource).

We opted to avoid this definition since we were not able to find a way to distinguish the two entities in terms of data and object properties. However we let the possibility to introduce this definition in the future using different type of ID to distinguish them in our on going version of the project.

Another possibility to slightly change the dataset metadata definition is swapping out the

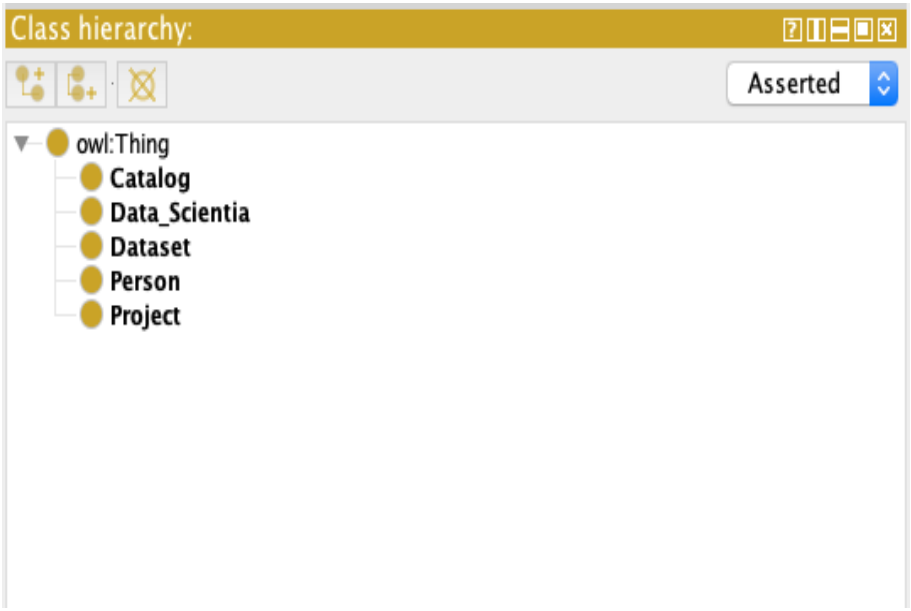


Figure 4: Entities type in Protegé

Figure 5: Basic Object Property

three 'Knowledge resource', 'data resource' and 'language resource' property for a single 'points at' property. Basically the only difference we have between the property we use nowadays is the interested informational layer of the resource pointed by the considered dataset. This information can be obtained simply inspecting the pointed dataset data property 'Layer'. In practice in the actual state of the work we have a small redundancy that cooperate to enlarge the Dataset e-type dimension. However we think it can be useful to have a quicker way to distinct in the tables the type of pointed resource since should lead to easier queries further in the project.

6 Informal Modeling

6.1 Teleology in depth description

Our informal modeling phase derives from a backtracking with respect to the inception phase. The etype **Data Scientia** represents the Data Scientia web portal, which has Web_Page as its only data property. Data Scientia is a contextual resource because it carries specific and unique information from the domain of interest, the Open Data environment, of which Data Scientia is a concrete realisation. Data Scientia is characterized by its Mission, Vision and Manifesto: Unitas per Varietatem. It is our universal set, container of all resources. Teleology allows us to model the heterogeneity of knowledge it contains.**Data Scientia** is a spatialPartOf **Everything** because it is our root, nothing is bigger than Data Scientia, our spatio-temporal context.
Data Scientia's web portal is composedBy 4 **Catalog**: Live People, Live Knowledge, Live Data and Live Language. Live People is a web page collector of diachronic data about university students over a period of time, and also additional synchronic data about profile, e.g., demographics, routines, personality; Live Knowledge is a web page catalog, containing all knowledge base resources; Live Data is a web page collector of data resources, such as the actual datasets, pointing at actual data values; finally, Live Language is a web page storage unit for projects' language layers.
The etype **catalog** is a core resource because of the aforementioned classification based on informational layers. Catalogs are aligned with the iTelos methodology and the stratified process of Knowledge Graph Engineering. Other than its unique ID, Catalog_ID, which is necessary for the creation of the actual dataset, Catalog has multiple data properties: Web_Page, which is the access point; Creation_Date, Title, Description, Language, License. These metadata allow users to find their routes in the web portal: if they are looking for schema resources, then Live Knowledge, which is the Catalog Title will redirect them accordingly. If users are interested in the License of Live Data, because they want to re-use it for commercial or non-commercial

purpose, they have this information available. Each (1) **Catalog** is composedBy N **Dataset**.
The etype **Dataset** is a set of resources belonging to a specific informational layer: either Knowledge, Language, or Data. It shares the following data properties with Catalog: Title, Web_Page, Creation_Date, Language, Description and License. The properties Format and Theme and geography make the single dataset retrievable by users interested in a particular data format, field of research, e.g. Healthcare, Transportation or Education, and place, e.g. Italian region, US, UE. These properties give the impression that Dataset could be represented as a common resource, because it contains aspects that are common to all domains, also outside the domain of interest. The added properties that make it become a core resource are: Layer, Language_Resource, Data_Resource and Knowledge_Resource. As shown in figure 18, A **Dataset** layer is accessible through the tag Layer, which let users know straight away the kind of resource they are looking at, without the need of downloading it to inspect its content. Furthermore, users can uncover one of main core features of Data Scientia: each (1) **Dataset** is spatialPartOf N **Dataset**. This means that users searching for a specific data resource can retrieve also Knowledge and Language Resources that are used as a whole, together. This consideration applies also if users are searching for a knowledge or a language resource: they are related to the the other 2 remaining layers if thus resources exists as instances of the etype **Dataset** itself.

There is another link that represents a semantically different point of view: Each (1) **Dataset** pointsAt N **Dataset**. Not only the user is able to decompose a complex dataset in its different layers, but also each informational resource points at the others that are used to recompose the Dataset: it is possible that two different KGs use the same Language and Knowledge Resources, that appear both linked to each other and to the KG, but a different Data Resource: these meaningful information is clearly displayed through metadata, which is a disruptive innovation, this composability and decomposability of a KG is unprecedented.
We need to introduce now the etype **Project**, since we have stated the following object properties

- Each (1) **Dataset** is spatialPartOf N **Project**;
- Each (1) **Project** is is composedBy N **Dataset**.

The first function emphasize that each Dataset can be part of different projects (highlighting the possible reusability of the resources that this model offers). The second function underline that each project usually contains many different Dataset resources that are linked and provided to the user. From our knowledge of Data Scientia's web portal we can describe 3 different kinds of **Project**, all sharing the data properties related to Project_ID, Title, Purpose, Description, Creation_Date, Last_Update, Completion_Date, Access, License, Language, Layer and

Web_Page. The distinctive properties that make Project a core resource is the presence of the tag Layer and Purpose, which are exclusive of **KGE Project**. We can define also **Application Project**, which includes Partner, Programming_Language and Release as its supplementary data properties, and **Study_Project**, which link to a specific Documentation_URI and Partner. We are limiting for now our investigation to a general concept of Project since **Data Scientia** is composedBy N **Project**: Projects are stand alone elements, they have their web page (github repository), but Data Scientia embeds them as its part, so it is a natural consequence that Data Scientia is a function of Project, a consumer of them. The last etype to be addressed is **Person**. Its data properties are: Person_ID, First_Name, Last_Name, Affiliation, Publication, Contact, and Role. These properties make Person a common resource: these metadata are common and standardized.

Is important to define that there are Each (1) **Person** that worksOn **Data Scientia** since it's a web resource that is nowadays in construction. They maintain and eventually update the portion of the portal currently available while providing new resources for the portal itself.

The **Data Scientia** web portal is composedBy N **Person**. This means that there is a embedded community in the portal itself but, since according to our domain expert it is not defined yet how it will be structured and how it will operate along with Data Scientia, we decided to not consider the community aspect for now.

In our schema **Person** objects are mainly seen as both **Dataset** and **Project** producers: we can have M **Dataset** producedBy N **Person**, as well M **Project** producedBy N **Person**.
This means that in our teleology **Project** and **Dataset** assume the role of the consumer with respect to the producer **Person**.
All the above inferences are shown in 6.

6.2 KarmaLinker

After importing ns_Teleology as Owl Ontology
Data_Scientia.csv

- The column *Web_Page* becomes *uri of Class* and its corresponding *semantic type* is *etype:Data_Scientia1*;
- The column *Catalog* becomes *uri of Class* (etype) and its corresponding *semantic type* is *etype:Catalog1*;

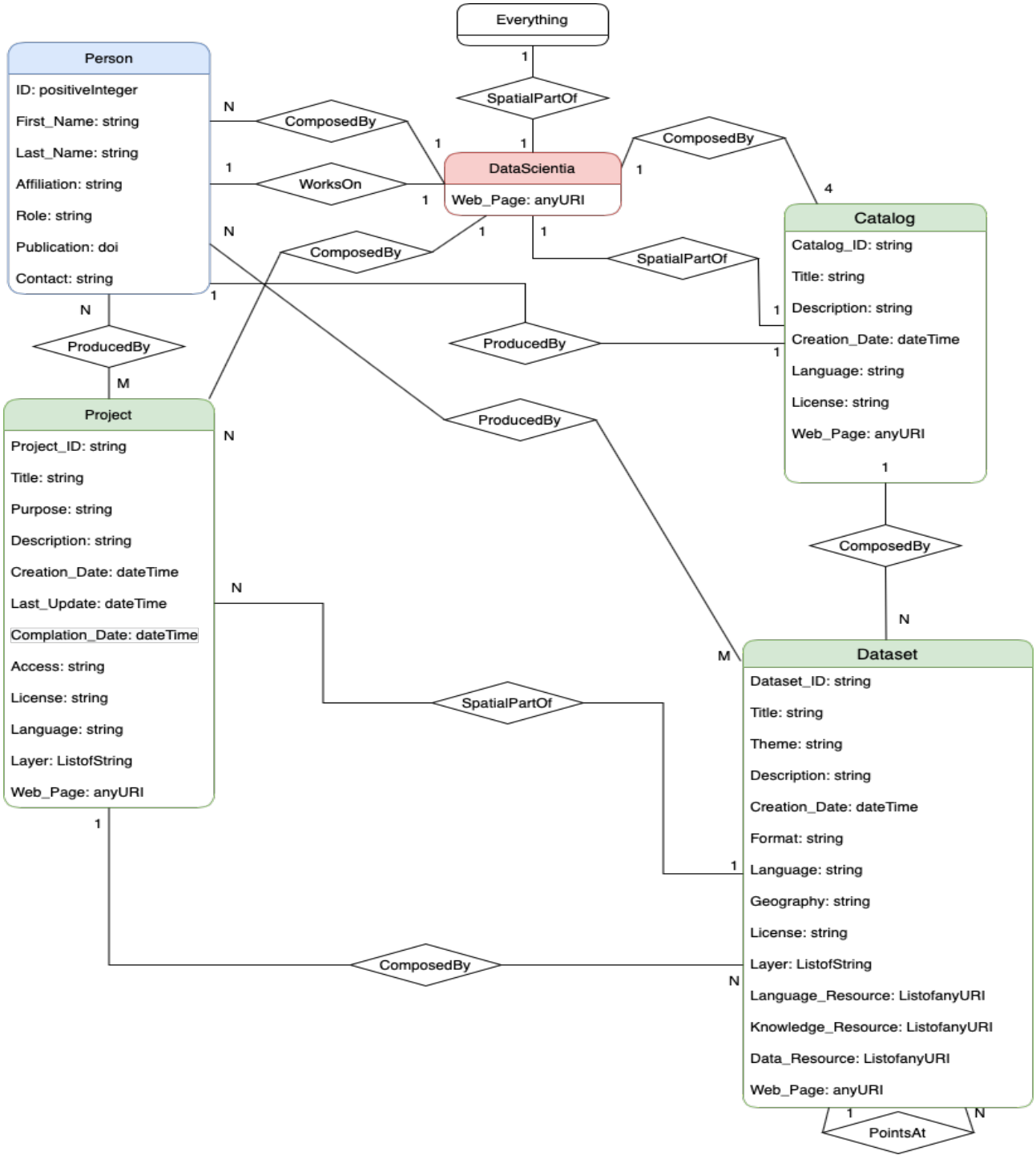


Figure 6: Teleology

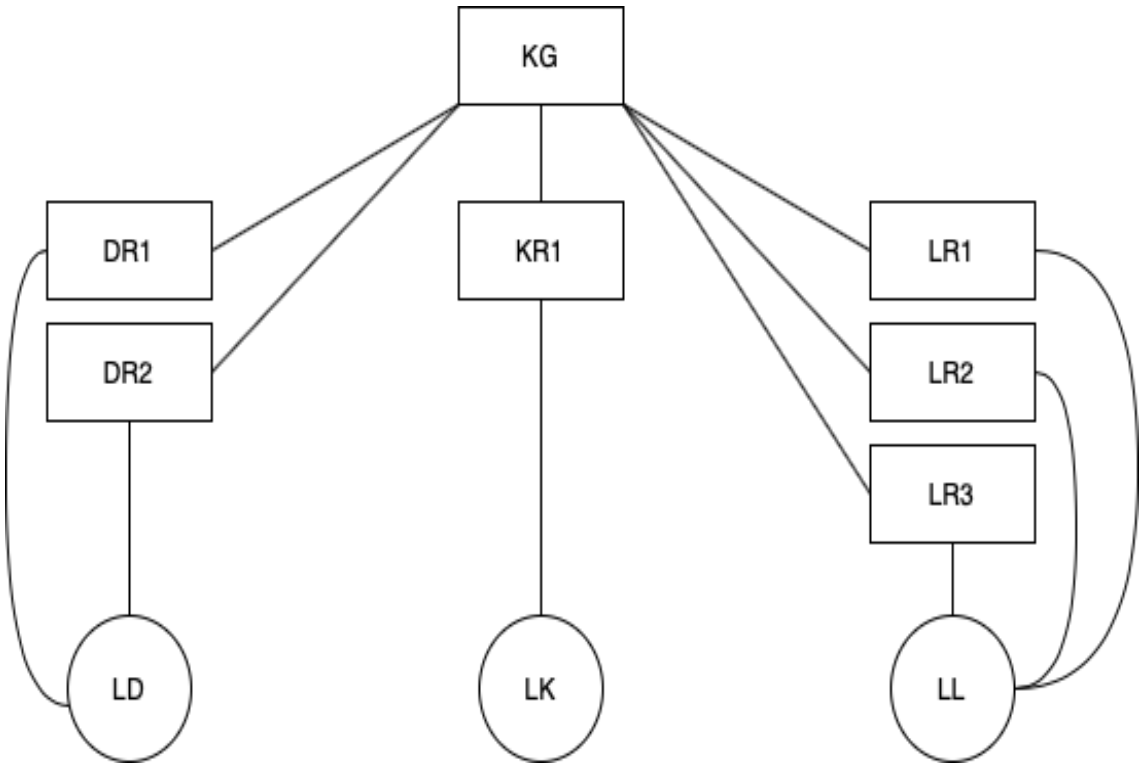


Figure 7: Dataset Composed structure

- The column *Person* becomes *uri of Class* and its corresponding *semantic type* is *etype:Person1*
- The column *Project* becomes *uri of Class* and its corresponding *semantic type* is *etype:Project1*

Outgoing links for *Data_Scientia1*: to classes *etype:Catalog1*, *etype:Person1*, *etype:Project1* with the property *etype:hasPart*.

7 Formal Modeling

Aiming at maximum reusability and shareability of teleologies in similar spatio-temporal contexts, we decided to conform as much as possible to ontologies available to describe Open data Catalog/web portals. The Data Catalog Vocabulary (DCAT) is an RDF vocabulary developed by W3C: it is designed to facilitate semantic interoperability between data catalogs published on the Web. We have delved into DCAT documentation, which defines the vocabulary schema and provides examples for its use, to figure out how to assign namespaces to our concepts in order to make our etypes as much reusable as possible. We have used Data Catalog Vocabulary (DCAT) - Version 2 W3C Recommendation 04 February 2020 (`https:`

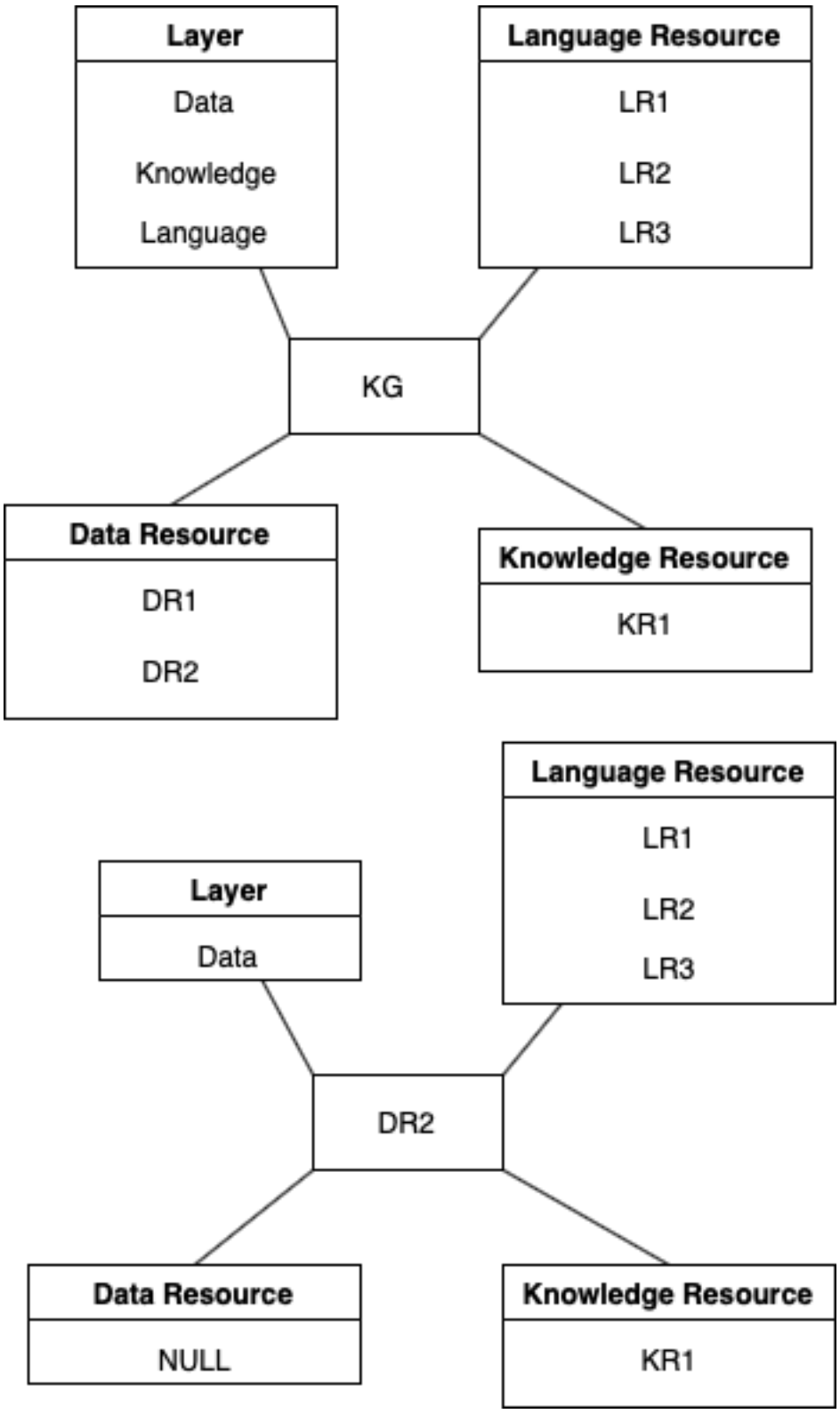


Figure 8: Dataset Basic Configuration

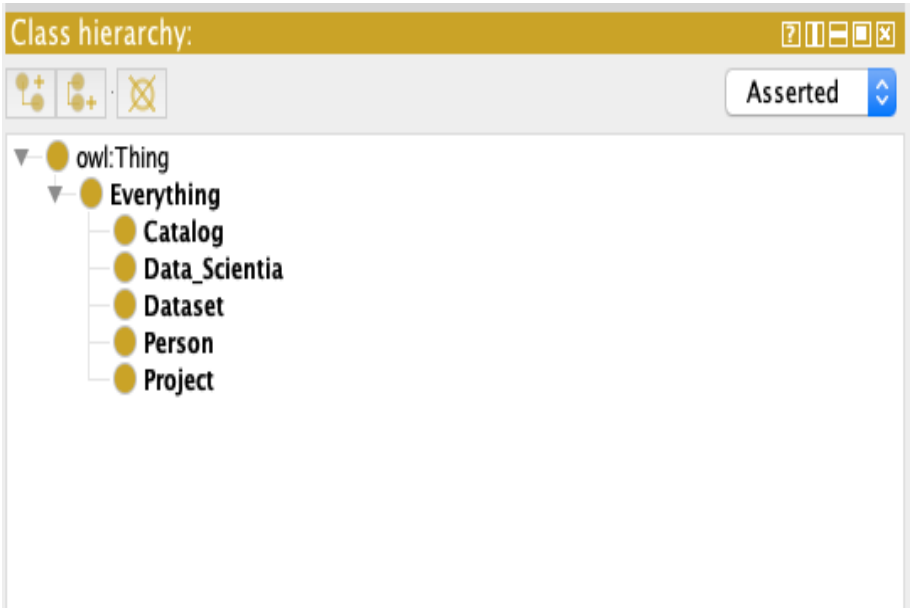


Figure 9: Teleology E-Types

Figure 10: Teleology object property

//www.w3.org/TR/vocab-dcat-2/), in conjunction with its specification for metadata records of European data portals, that is DCAT Application Profile (DCAT-AP), version 2.1.0. (https://joinup.ec.europa.eu/collection/semantic-interoperability-community-semic/solution/dcat-application-profile-data-portals-europe/release/210).

Prefix	Namespace
dcat:	http://www.w3.org/ns/dcat#
dct:	http://purl.org/dc/terms/
doap:	http://usefulinc.com/ns/doap#
ds:	http://knowdive.disi.unitn.it/etype#
foaf:	http://xmlns.com/foaf/0.1/
rdfs:	http://www.w3.org/2000/01/rdf-schema#
schema:	http://schema.org/
skos:	http://www.w3.org/2004/02/skos/core#
xsd:	http://www.w3.org/2001/XMLSchema#

Figure 11: Selected Namespaces for the Formal Modelling.

Through this process we conceptualised our ontology to make it as shareable as possible defining a formal explicit specification of our etypes (12). Now that we have our personal vision of a small space of the world we are interested in 13 along with the ontological representation of the world it self, we can merge the two obtaining our teleontology 14.
In the following subsections we are going to explore in details our final schema analyzing one by one each etype, its characteristics and properties.

7.1 Catalog

Currently, our **Catalog** etype has a KGE course-specific namespace URI (Protégé IRI), that is http://knowdive.disi.unitn.it/etype#, which can be referenced by the prefix ds:, to which Catalog is added. However, our etype can be fully mapped into the DCAT-AP class Catalogue, with URI dcat:Catalog, defined as:

> "A catalogue or repository that hosts the Datasets or Data Services being described"

The corresponding reference is: http://www.w3.org/TR/2013/WD-vocab-dcat-20130312/#class-catalog. The namespace URI is dcat:http://www.w3.org/ns/dcat# to which Catalog should be added.

Furthermore, the data and object properties - that have the KGE course-specific namespaces - relative to our **Catalog** etype, can be wholly translated into DCAT-AP properties, thus achieving our objective of maximum reusability. The following data properties of the previous

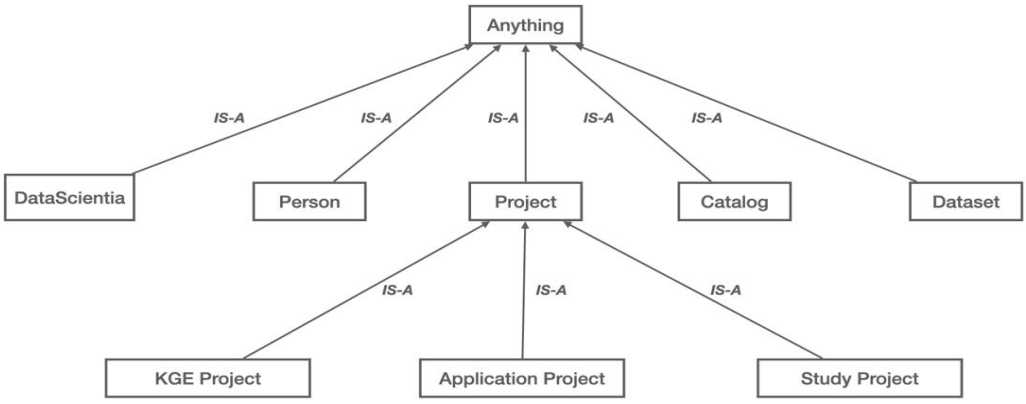


Figure 12: Ontology

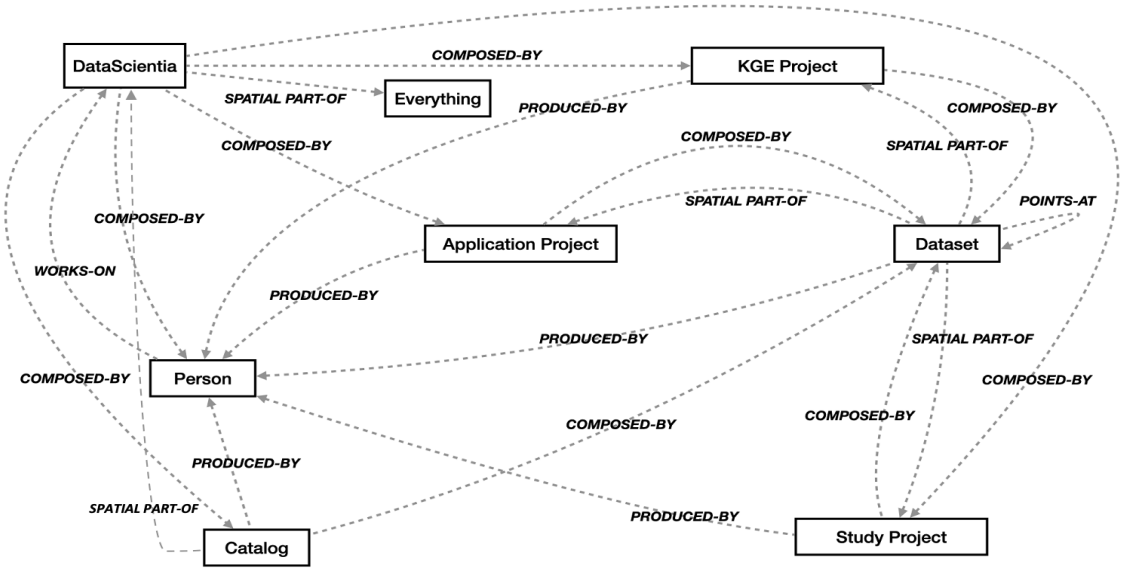


Figure 13: Teleology

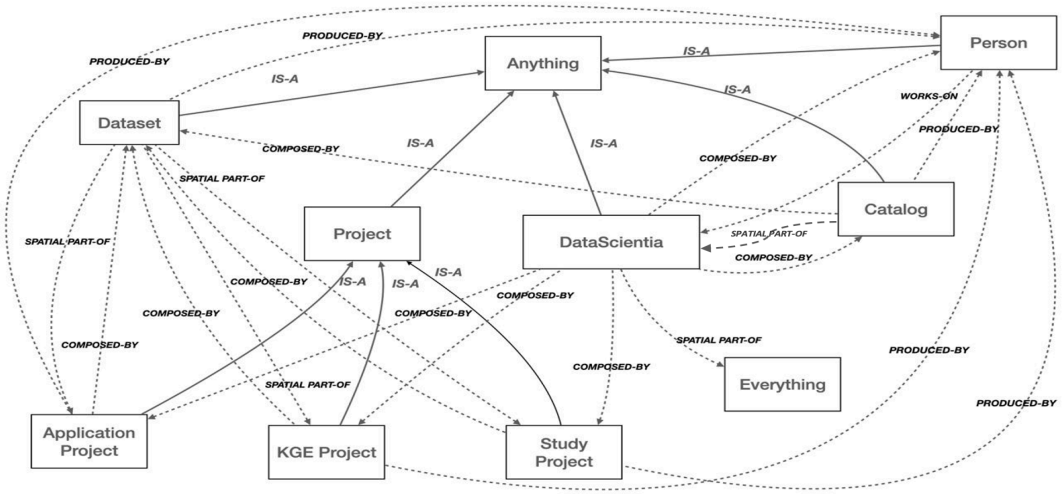


Figure 14: Teleontology

etype ds:Catalog, that is now dcat:Catalog, will be changed accordingly to match the DCAT standardization:

- **has_Catalog_Catalog_ID**: a general unique identifier, corresponding to dct:identifier, which is related to the namespace dct:http://purl.org/dc/terms/, to which identifier is added. This property belongs to the specification of all metadata terms maintained by the Dublin Core™ Metadata Initiative.
- **has_Catalog_Title**: the catalog title, such as LiveKnowledge or LiveData, a simple string. This property is matched by dct:title.
- **has_Catalog_Description**: A free-text account of the catalog, which coincides with dct:description.
- **has_Catalog_Creation_Date**: a time reference (year-month-day if present) for the catalog publication or just creation on the web portal; the catalog may be modified after this date. The related property on DCAT-AP documentation still belongs to DCMI and is dct:issued, which should be used to describe the date of formal issuance (e.g., publication) of an item.
- **has_Catalog_Language**: the language used to present the catalog, consistent with dct:language
- **has_Catalog_License**: coincides with dct:license, since it is fundamental to provide information about the licence under which the catalog can be used or reused.

  The last four properties are inherited from the super-class dcat:Resource.

- **has_Catalog_Web_Page**: the main page which shows all datasets contained in the catalog under observation. According to the DCAT-AP 2 documentation, this property can be mapped into foaf:homepage, where foaf: stands for the namespace http://xmlns.com/foaf/0.1/. DCAT incorporates this term from the pre-existing FOAF vocabulary, which is a project based around the use of machine readable Web homepages for people, groups, companies and so on.

Our **dcat_Catalog** etype and its data properties have accordingly been changed in Protégé to maximize reuse. In particular, the tool automatically changed the DCMI given prefix dct: into dcterms:, decision that is not compliant with DCAT-AP 2. However, since we have to meet our data types restrictions for ranges, we cannot use the suggested ranges for data properties by both DCAT-AP and DCAT (DCMI and FOAF, as well), which stated that dct:identifier, dct:title, dct:description, and dct:issued should have range rdfs:Literal, where the prefix rdfs: stands for the namespace http://www.w3.org/2000/01/rdf-schema#. Also, dct:language should be in the range of dct:LinguisticSystem and dct:license of dct:LicenseDocument. We are forced to convert all these ranges to xsd:string, since the only allowed data type that is similar to literal and the other complex ranges is string. Note that the prefix xsd: stands for the namespace http://www.w3.org/2001/XMLSchema#string

Let's now focus on object properties and see how we can exploit DCAT-AP to describe the interactions between Catalog, Dataset and DataScientia. The last two etypes are also going to be aligned with DCAT and DCAT-AP ontologies. The existing object properties for the previous ds:Catalog (now dcat:Catalog) are the following:

- **composedBy**: one catalog is physically and logically composed by many datasets, so the property that matches this one-to-many relationship is a function of catalog, which contains datasets as parts, in agreement with the teleology stated during the Informal Modeling phase.
  DCAT-AP provides the property dct:hasPart to describe a resource that is included either physically or logically in the described resource, that is our catalog. This property is very general, we need to underline that catalog is composed by datasets only. The property dcat:dataset is more suitable and specific because it defines a collection of data that is listed in the catalog.
  The object property ds:composedBy has accordingly been changed in Protégé to maximise reusability. In particular, the domain dcat:Catalog is re-defined and the range dcat:Dataset as well (also the field subClass of belonging to dcat:Catalog mirrors the applied change).
- **spatialPartOf**: a catalog is in turn part of the etype **DataScientia**. DCAT-AP provides the property dct:isPartOf, which accounts for a resource in which the described resource is physically or logically included. In this case many catalogs are contained in DataScientia:

Catalog_ID	Title	Description	Creation_Date	Language	License	Web_Page	Data_Scientia	Dataset	Person
Catalog_1	Live Data	a web page collector of data resources, such as the actual datasets, pointing at actual data values.	2022	EN	CC-BY 4.0	https://ds.datascientia.eu/catalog/livedata/home	https://ds.datascientia.eu	Dataset_S_1	Person_0
Catalog_1	Live Data	a web page collector of data resources, such as the actual datasets, pointing at actual data values.	2022	EN	CC-BY 4.0	https://ds.datascientia.eu/catalog/livedata/home	https://ds.datascientia.eu	Dataset_S_2	Person_0
Catalog_1	Live Data	a web page collector of data resources, such as the actual datasets, pointing at actual data values.	2022	EN	CC-BY 4.0	https://ds.datascientia.eu/catalog/livedata/home	https://ds.datascientia.eu	Dataset_S_3	Person_0
Catalog_1	Live Data	a web page collector of data resources, such as the actual datasets, pointing at actual data values.	2022	EN	CC-BY 4.0	https://ds.datascientia.eu/catalog/livedata/home	https://ds.datascientia.eu	Dataset_S_4	Person_0
Catalog_2	Live Knowledge	a web page catalog, containing all knowledge base resources.	2022	EN	CC-BY 4.0	https://ds.datascientia.eu/catalog/liveknowledge/home	https://ds.datascientia.eu	Dataset_S_5	Person_0
Catalog_4	Live People	a web page collector of diachronic data about the everyday life of university students over a period of time, and also additional synchronic data about profile, e.g., demographics, routines, personality.	2022	EN	CC-BY 4.0	https://livepeople.datascientia.eu/home	https://ds.datascientia.eu	Dataset_C_7	Person_0
Catalog_4	Live People	a web page collector of diachronic data about the everyday life of university students over a period of time, and also additional synchronic data about profile, e.g., demographics, routines, personality.	2022	EN	CC-BY 4.0	https://livepeople.datascientia.eu/home	https://ds.datascientia.eu	Dataset_C_8	Person_0
Catalog_2	Live Knowledge	a web page catalog, containing all knowledge base resources.	2022	EN	CC-BY 4.0	https://ds.datascientia.eu/catalog/liveknowledge/home	https://ds.datascientia.eu	Dataset_S_9	Person_0
Catalog_1	Live Data	a web page collector of data resources, such as the actual datasets, pointing at actual data values.	2022	EN	CC-BY 4.0	https://ds.datascientia.eu/catalog/livedata/home	https://ds.datascientia.eu	Dataset_S_10	Person_0
Catalog_2	Live Knowledge	a web page catalog, containing all knowledge base resources.	2022	EN	CC-BY 4.0	https://ds.datascientia.eu/catalog/liveknowledge/home	https://ds.datascientia.eu	Dataset_S_11	Person_0
Catalog_3	Live Language	a web page storage unit for projects' language layers.	2022	EN	CC-BY 4.0	https://livelanguage.datascientia.eu/home	https://ds.datascientia.eu		Person_0

Figure 15: Catalog table alignment after Formal Modelling. The blue column fields are the data properties, while the orange ones are object properties. The yellow column is the new addition

it is a many-to-one relationship.

At the moment, the actual DataScientia web portal, hosted at *https://ds.datascientia.eu/* contains five Catalogs, of which we are only considering the following four: LiveData, LiveKnowledge, LiveLanguage and LivePeople. So, as far as we are concerned the property dcterms:isPartOf, as dct:isPartOf gets translated in Protégé, reflects a 4-to-1 relationship between dcat:Catalog and DataScientia, being the first a function of the latter.

A more in-depth look at the DCAT-AP 2 documentation made us realize that one of the mandatory properties for dcat:Catalog was missing: dct:publisher, which defines the fundamental figure of an agent, entity (organisation) responsible for making the catalog available. So we have decided to add as an object property dct:publisher, which replaces our general **producedBy** property. In order to finalize this change, we had to add a **Person** column to the Catalog table.

7.2 Person

Person is our most generic etype, since it will probably be extended and specified in the future to comply with the development of DataScientia's platform and website. As a matter of fact, DataScientia is a people-centric data space, which aims at exploiting its community diversity as a key feature. Cooperation between users from different cultural, social and economical context will allow the shareability of huge information amounts, contributing to the creation of a circular benefit environment, where each user helps the others by achieving his/her personal goals. Also, supporting users by providing them the resources they need to satisfy their purposes is

actually our metadata project purpose.
However, the discrimination between different types of users and contributors has still fuzzy boundaries, so we are limiting ourselves to the description of a general Person that could be later used as a basis for addition of further properties and attributes. A person can be a user, a single private not associated to any organization, a researcher, a data scientist, a data intermediary, a student, a company and so on, who might be searching data for primary or secondary use.
DCAT does not provide a proper class Person, but it redirects to foaf:Person for people and foaf:Organisation for government agencies or other entities in its documentation (section 6.12), saying that several properties to describe these entities are available on FOAF. Since we were looking for a generic Person class to match our etype **Person** and its data and object properties, we explored FOAF Vocabulary Specification (Namespace Document 1 May 2004) to find analogs for the following data properties:

- **has_Person_ID**: a general identifier of the person, in order to properly relate the actual intersecting data tables of this project; FOAF only provides identifiers relative to Online Accounts, so we did not find a matching property;
- **has_Person_Name**: since Person could be an individual, with name and surname, but also an organisation, we have decided to keep this field simple. The corresponding property in FOAF Basics would be foaf:name
- **has_Person_Contact**: a way to contact the person under observation, we have chosen email as the main contact point: the corresponding in FOAF is foaf:mbox.
- **has_Person_Affiliation**: the workplace, or research institute/group, or any organisation a person is currently linked to. A similar, but not equivalent property in FOAF Personal Info category is foaf:workplaceHomepage. The documentation also remarks that:

  > "FOAF often indirectly identifies things via Web page identifiers where possible, since these identifiers are widely used and known."

- **has_Person_Publication**: if the Person is a researcher, professor, or PhD, published scientific articles, books, held seminars, newspaper interviews and so on, should be taken into account. foaf:publications is equivalent to our ds specific data property.
- **has_Person_Role**: the role a Person has in the affiliation he is linked to. The closest property in FOAF would be foaf:workInfoHomepage, which redirects to a page regarding someone's professional role within an organisation or project.

Let's see if our only object property finds a corresponding relation in FOAF Personal Info:

- **worksOn**: connects a person with the DataScientia Web portal. FOAF does not currently have a term for the name of the relation (eg. ”workplace”) that holds between a foaf:Person and an foaf:Organization that they work for. The closest property to depict the fact that a Person is working on something is foaf:currentProject, which does not seem a right fit.

If we want to use FOAF Vocabulary Specification, as DCAT suggests, we would have to change the values of Affiliation and Role columns of the Person table. Also we would need to deal with the absence of a property for identifier, which links our intersecting tables, for example Project and Person, DataScientia and Person and so on. As an alternative we could use the Person type provided by *Schema.org*. The corresponding namespace is https://schema.org/, summarized in the prefix schema:, to which Person is added. If we read the documentation, we can see that every data and object properties specific to our http://knowdive.disi.unitn.it/etype# namespace can be subsumed by already existing properties of schema:Person:

- schema: identifier: any kind of identifier for any kind of Thing (inherited from Thing);
- schema:name: general name of an item (inherited from Thing)
- schema:affiliation: An organization that this person is affiliated with. For example, a school/university;
- schema:hasOccupation: the person’s occupation, which can be easily mapped to our **has_Person_Role** data property;
- schema:contactPoint: a way to contact the person: it is general enough to be specified in the future, also with multiple ones, such as email, phone, twitter and linkedin. Currently, in our Person table, we are using email as a value for **has_Person_Contact**, so we could switch to the more specific schema:email property;
- schema:author:the author of this content or rating. This property accounts for any kind of resource produced by a person, including publications. So, it can be aligned with
- **has_Person_Publication**. We think it could also be an improvement since publication could be a restrictive term: a professor could have also published a book or a video-lesson. As for our **ds:worksOn**, there is a matching scheme:worksFor. Since the Data Scientia community is still a broad term, and if a user shares resources he is actually contributing to the portal, he/she can be considered as working for the platform and the whole community as well.

We have decided to use schema:Person and its properties to map our etype **Person**: in this way we are maximizing reusability and keeping our teleontology more compact.

7.3 Dataset

To describe our **Dataset** etype, we are switching back to DCAT-AP 2: this class is fundamental for the definition of an open data web portal, so it must be compliant with DCAT Vocabulary, at least in its basic properties. The DataScientia ds:Dataset etype carries additional data and object properties that make it a contextual resource, like the informational layers, or the possibility to compose and decompose a dataset, so it can embed only some DCAT properties, and if that is not exhaustive, it may also use properties belonging to different Vocabulary specification or reference ontologies. DCAT-AP 2 lists, as mandatory properties, dct:description and dct:title, taken from DCMI, which are analogs of our ds data properties **has_Dataset_Description** and **has_Dataset_Title**.
DCAT includes as recommended properties:

- dcat:theme:a category (or multiples) of the Dataset. The range is skos:Concept: SKOS stands for Simple Knowledge Organization System and, according to the Namespace Document - HTML Variant 18 August 2009 Recommendation Edition, it is:

  > "a common data model for sharing and linking knowledge organization systems via the Semantic Web."

  The above mentioned data property fully corresponds to **has_Dataset_Theme**, so we can reuse it;

- dct:spatial: refers to a geographic region or named place that is covered by the Dataset, its range isdct:Location. We can substitute our ds **has_Dataset_Geography** with this equivalent DCMI property.

Our ds:Dataset is also characterized by temporal metadata: it shares the ds specific data property **has_Dataset_Creation_Date** with Catalog. This is then going to be mapped as well to dct:issued. We have decided to add two more data properties to describe the Dataset etype: **has_Dataset_Completion_Date** and **has_Dataset_Last_Update**, since it is very important to keep track of possible ongoing modifications of a created dataset, which could eventually lead to a finalised version, and later be updated to conform to new standards or key data drifts. DCMI provides dct:modified to account for the date on which the resource was changed. Scrolling DCMI Metadata Terms documentation webpage (*https://www.dublincore.org/specifications/dublin-core/dcmi-terms/*), we have noticed the term dct:dateAccepted that could mirror our data property **ds:has_Dataset_Completion_Date**, because when a dataset is actually in its final version, it is accepted by the authors that issued it in the first place. DCAT-AP considers the following properties optional:

- dct:identifier: the identifier might be used as part of the URI of Dataset, but still having it represented explicitly is useful, because we need a unique identifier to facilitate the linkage between different etype tables that share key object relationships; the corresponding ds data property is **has_Dataset_Dataset_ID**;
- dcat:landingPage: a Web page that can be navigated to gain access to the dataset, its distributions and/or additional information. We have not taken into account the possibility of adding an etype *dcat:Distribution* because the Data Scientia web portal is still under construction and for now datasets' and catalogs' structures are the main focus of investigation. Our matching property is **has_Dataset_Web_Page**;
- dct:language: the language of the dataset, replacing **has_Dataset_Language**;
- dct:accessRights: refers to information that indicates under which constraints the Dataset is provided, for example open data, access restrictions, being (or not) public, privacy, security, or other policies. This property is different from our ds data property **has_Dataset_License**, which is about stating the type of permission under which the resource is (or is not) reusable. We use a literal value to identify the license, like "CC-BY 4.0", so a better fit would probably be the property dct:license, which is recommended for Distributions of Datasets. Since, from our point of view, these 2 objects are conceptually merged in Dataset, we have chosen the latter to substitute our **ds:has_Dataset_License** property.

We can use another dcat:Distribution property, inherited from DCMI, dct:format, to match **has_Dataset_Format**, which is a literal value that identifies the file format in the Dataset table.
The remaining data property, according to Protégé, is **has_Dataset_Layer**: this explains why ds:Dataset is a contextual resource. This property is mirrored in the table with a string value, indicating the layers (can be more than one) that the dataset is composed of: "Knowledge", "Data", "Language". There isn't a matching property in any other reference schema, because the term Layer is specific to our domain of interest. dcat:keyword and dcat:theme are available properties for dcat:Dataset to make a label or category explicit. However, Layer is part of the project metadata Vocabulary specification, it cannot be subsumed by a different term, even if there would be a gain in reusability. In order to support the user's need Data Scientia handles the resources adopting a stratified approach, namely the distinction in informational layers, to exploit as much as possible the information provided by diverse suppliers, from different geographical and cultural areas. This theoretical background also explains the following DataScientia object properties, which cannot be mapped to already existing ontologies, because they represent a novel and innovative stratified approach to data:

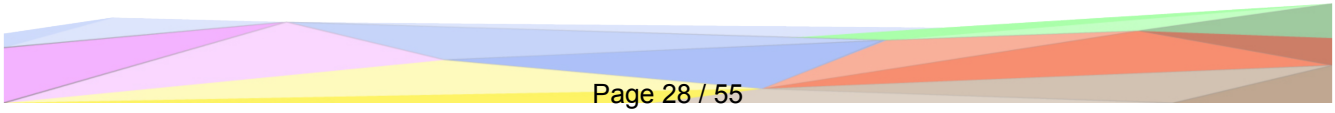
- **points_Dataset_at_Language_Resource**: each ds:Dataset is linked to a Lan-

guage_Resource, if present. For example, if the dataset under observation is a Data_Resource, it may have been build using a specific Language_Resource, which the dataset author may or may have not made available as a separate language-specific dataset, along with the actual dataset of values. Note that the corresponding column table *points_Dataset_at_Language_Resource* of our **Dataset** etype has as its range a ds:Dataset type. The same consideration holds for the remaining two object properties;

- **points_Dataset_at_Data_Resource**: each ds:Dataset is linked to a Data_Resource, if present. For example, if the dataset under observation is a Knowledge_Resource, it may have been applied to a domain-specific Data_Resource, which the dataset author may or may have not made available as a separate dataset, along with the proposed teleology/ontology/teleontology;
- **points_Dataset_at_Knowledge_Resource**: each ds:Dataset is linked to a Knowledge_Resource, if present. For example, if the dataset under observation is a Language_Resource, it may have been developed to match a precise collection of data, which the dataset author may or may have not made available as a separate actual dataset of values.

We need to underline that the type of resource that a dataset contains, and that qualifies it, is stored in the ds:layer data property. If the dataset is a Language_Resource, its layer will be "Language"; if it is a Knowledge_Resource, then the Layer is "Knowledge"; if the dataset is a Data_Resource, the corresponding layer would be "Data".
The above mentioned three object properties are mapped into the Dataset table through three columns: "Language_Resource", "Data_Resource", and "Knowledge_Resource". If a dataset is of layer "Knowledge", it is likely that the column value for "Knowledge_Resource" will be empty, since the dataset itself is a "Knowledge_Resource". The other two columns values may be present or not, depending on the type of resources made available along with the dataset under observation. There could also be multiple values for the same column: for example, we could have two different language resources connected to the same teleontology. Also, the column value matching the Dataset type of resource could be non-empty: this could happen if the dataset is actually composed by multiple single datasets of the same kind.
If we can extract all the different layers of a dataset, it means that the dataset actually represents a working Knowledge Graph at its full potential: all three object properties are functioning and the Dataset table should have non-empty values in each corresponding resource column: every resource is linked both to the others and to the main dataset entity, allowing compositionality.

The remaining object properties, that we still have to map to reference ontologies, if possible, are the following:



Layer	Language_Resource	Knowledge_Resource	Data_Resource	Web_Page	Author	Project
Data		Dataset_S_5		https://www.trentinotrasporti.it/open-data	Trentino Trasporti S.p.A.	KGE_P_1
Data				https://os.smartcommunitylab.it/core.mobility/bikesharing/pergine_valsugana	Servizio Trasporti pubblici	KGE_P_1
Data				https://dati.trentino.it/dataset/stazioni-bike-sharing-emotion-trentino/resource/e642bac9-06db-43a8-80df-684d031db2ec	Servizio Trasporti pubblici	KGE_P_1
Data			Dataset_S_2	https://dati.trentino.it/dataset/stazioni-bike-sharing-emotion-trentino	Servizio Trasporti pubblici	KGE_P_1
Data			Dataset_S_3	https://dati.trentino.it/dataset/stazioni-bike-sharing-emotion-trentino	Servizio Trasporti pubblici	KGE_P_1
Knowledge			Dataset_S_1	https://developers.google.com/transit/gtfs/reference	Google Transit	KGE_P_1
Data; Knowledge		Dataset_S_5	Dataset_S_1	https://drive.google.com/drive/folders/1d7QPcG3ul9aHw_UXUsKuiDtCzOwapA-f	Person_3	KGE_P_1
Data; Knowledge		Dataset_S_5	Dataset_S_1	https://drive.google.com/drive/folders/1d7QPcG3ul9aHw_UXUsKuiDtCzOwapA-f	Person_4	KGE_P_1
Data; Knowledge		Dataset_S_5	Dataset_S_2	https://drive.google.com/drive/folders/1d7QPcG3ul9aHw_UXUsKuiDtCzOwapA-f	Person_3	KGE_P_1
Data; Knowledge		Dataset_S_5	Dataset_S_2	https://drive.google.com/drive/folders/1d7QPcG3ul9aHw_UXUsKuiDtCzOwapA-f	Person_4	KGE_P_1
Data; Knowledge		Dataset_S_5	Dataset_S_3	https://drive.google.com/drive/folders/1d7QPcG3ul9aHw_UXUsKuiDtCzOwapA-f	Person_3	KGE_P_1
Data; Knowledge		Dataset_S_5	Dataset_S_3	https://drive.google.com/drive/folders/1d7QPcG3ul9aHw_UXUsKuiDtCzOwapA-f	Person_4	KGE_P_1

Figure 16: Dataset table alignment after Formal Modelling. The red column fields are the data properties, while the pink ones are object properties.

- **producedBy**: accounts for the creator of the dataset, or anyone responsible or contactable for it. We can reuse the ds:publisher property, as we did for Catalog. This time we do not have to modify the Dataset table, since the connection between a Dataset and its Author (represented by the schema:Person etype) is already present;
- **spatialPartOf**: a dataset is part of a Project, as previously stated by our teleology. We can reuse the dct:isPartOf property. The same part-whole relationship between Catalog and DataScientia holds between Dataset and Project, but the difference lies in the cardinality: many Catalogs are part of one DataScientia portal while many datasets can be part of many different projects, depending on the purpose, domain of interest and considered resources.

7.4 Project

The **Project** parent class and its children (Figure 17), cannot be mapped to an already existing Project definition, because it provides the DataScientia specific data property ds:purpose. As stated by Professor Fausto Giunchiglia in the KGE 2022 course:

> "We define the user objective, The Purpose which will lead the entire KGE process. Such an objective implicitly includes the user "point of view", the representation that the user uses to model (a portion of) the world, where the desired information lives. "

The Purpose becomes a self-contained property, essential to DataScientia, whose purpose in turn is to allow the user to retrieve different resources singularly and all the different kind of

resources related to a specific KGE process.
The next ds:Project data properties to analyse are the ones that can be reused from reference ontologies, shared with the already formally modelled etypes (dcat:Catalog, schema:Person and ds:Dataset):

- **has_Project_Project_ID**: can be translated into dct:identifier, as already done with the above mentioned etypes;
- **has_Project_Title**: can be converted into dct:title, as already done with ds:Dataset, ds:Project and dcat:Catalog;
- **has_Project_Description**: can be translated into dct:description, as already done for ds:Dataset, ds:Project and dcat:Catalog;
- **has_Project_Creation_Date**: can be converted into dct:issued, as already done for ds:Dataset, ds:Project and dcat:Catalog;
- **has_Project_Completion_Date**: can be mapped into dct:dateAccepted, as we did for ds:Dataset and ds:Project;
- **has_Project_Last_Update**: can be mapped into dct:modified, as we did for ds:Dataset and ds:Project;
- **has_Project_License**: can be mirrored by dct:license, as already stated for ds:Dataset, ds:Project and dcat:Catalog;
- **has_Project_Language**: can be translated into dct:language, like we did for ds:Dataset, ds:Project and dcat:Catalog;
- **has_Project_Web_Page**: can be mapped into dcat:landingPage, as already done for ds:Dataset. Note that dcat:landingPage is different from foaf:homepage, chosen for dcat:Catalog, because the first one refers to a Web page that needs to be visited in order to gain access to the actual Project, for example its github repository, while the latter is the main page where the resource is directly accessible through the DataScientia web portal, in our case the Catalogs' homepages.

The remaining ds:Project property that needs to be translated, if possible, to an existing ontology, from scratch is **has_Project_Access**. We have tackled dct:accessRights when discussing Dataset properties: in this case it is the proper fit, since we are interested in the privacy, publicity, or authentication policies relative to a project. For example, if a Project has a corresponding github repository, its access could be private or public.
We have already encountered and mapped the object properties designed for the Project

class. They are **composedBy** and **producedBy**, which have been successfully translated into dcat:Dataset and dct:publisher respectively.

7.4.1 KGE Project

Let's now delve into the KGE_Project specific data property **has_Project_Layer**, which can be translated into ds:layer, as already stated for ds:Dataset.

7.4.2 Study Project and Application Project

A Study_Project specific data property is **has_Project_Study_Documentation**. The closest data property found, resembling a Project documentation, is foaf:page, which relates a thing to a document about that thing. So the Study_Documentation column available in the Study_Project csv table is going to be filled by a reference, probably a URL, to the actual Documentation wiki.

The data properties specific to Application_Project are **has_Project_Programming_Language** and **has_Project_Release**. They have been mapped using DOAP, which stands for Description Of A Project (`https://github.com/ewilderj/doap/wiki`). DOAP is a project to create an XML/RDF vocabulary to describe software projects, and in particular open source projects. We gained access to this Descriptor through LOV (Linked Open Vocabulary), which is in turn made available by *liveschema.eu* soon to become the liveKnowledge Catalog at `https://ds.datascientia.eu/catalog/liveknowledge/home`. The corresponding retrieved data properties are doap:programming-language and doap:release. The first one is straightforward, it is the programming language a project is implemented in or intended for use with, and the latter one refers to version information of a project release, which perfeclty matches our release property.

Application_Project and Study_Project share the common data property **has_Project_Partner**. After browsing different reference ontologies (DCMI, Schema.org, DOAP, DCAT, DCAT-AP 2, FOAF and so on) to find a data property matching ours, we realised that Partner would imply the involvement of a Person etype, which is already listed in our Project table, under the column name Author. So we decided to keep the umbrella term schema:Affiliation data property, also present in the schema:Person attributes, instead of a Partner property, because they actually contain the same information, without the need of an object property and a duplicated, rather unnecessary, link between the Person and Project tables; a Person can be linked to University of Trento, as well as a Project, if it was developed with the University resources.

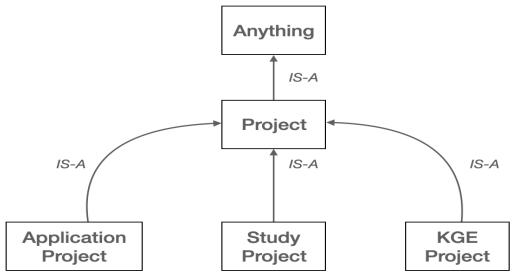


Figure 17: Teleontology portion. The focus is given to the Project class and its sub-classes Application, Study and KGE Projects, which were chosen to be three different etypes.

7.5 Data Scientia

The **Data_Scientia** etype cannot be mapped into an open data catalog according to DCAT-AP 2. The etype ds:DataScientia does not contain all DCAT-AP mandatory properties (dct:description, dct:publisher and dct:title) and, most importantly, DataScientia is an unprecedented type of open data web portal, that cannot be included in any existing class. It is extremely specific in its meaning and philosophy, Unitas per Varietatem. It is an inherently collaborative people-centric data space, offering dedicated multi-purpose services and it is supported by a dedicated infrastructure, exploitable for research and innovation studies/experiments/projects. The data produced is of quality and reusable. It is ontologically wrong to make it extend anything existing, it stands on its own original ground-truth.
Data Scientia data and object properties can all be mapped to already existing and already seen reference ontologies **has_DataScientia_Web_Page** to foaf:homepage; **composedBy** to dct:hasPart, which comprehends dcat:dataset as its sub-property; and **spatialPartOf** to dct:isPartOf.

7.6 UKC alignment

In order to manage the conceptual diversity of the chosen Etypes and of the data and object properties across different languages, we have exploited the Universal Knowledge Core (UKC) resource, provided by the KOS platform. UKC is a multilingual lexico-semantic resource, which turns our language and purpose specific concepts into codified terms of a *Language Entity Graph* (LEG), where semantic relations are established and unified across different languages into a UKC-aligned hierarchy, solving the problem of semantic heterogeneity and aiming at a standardization of our project[Giu+22].

In this paragraph we will give an in-depth overview of each language annotation defined for each chosen etype, object property and data property.

Project_ID	Title	Purpose	Description	Creation_Date	Last_Update	Completion_Date	Access	License	Language	Web_Page	Affiliation	Study_Documentation	Author	Dataset
STD_P_1	Smart Society / Smart University	Move towards hybrid systems where people and machines tightly work together to build a smarter society. The vision is a new generation of Collective Adaptive Systems where humans and machines synergically complement each other and operate collectively to achieve their, possibly conflicting, goals, but which also exhibit an emergent behaviour that is in line with their designers' objectives.	Empirical evidence has shown how students' time management ability and its consequent translation into time allocation between academic and other daily activities may have an impact on students' performance. Given their wide adoption among students, smartphones can help in understanding their behaviour and develop strategies for administrators and academic staff to enact policies that help students learn better strategies for managing their time and their academic workload. SmartUniversity is jointly developed by the Department of Information Engineering and Computer Science and the Department of Sociology and Social Research of the University of Trento. The main goal of the SmartUniversity project is to fill the empirical gap concerning students' time allocation and academic performance by providing a detailed description of how their time management affects their academic achievement.	2016-10	2022-03-14	2018-06	Controlled		EN	https://livepeople.datascientia.eu/workspace/smartuniintwo	University of Trento (IT)	Dataset_S_9	Person_8	Dataset_C_7
STD_P_1	Smart Society / Smart University	Move towards hybrid systems where people and machines tightly work together to build a smarter society. The vision is a new generation of Collective Adaptive Systems where humans and machines synergically complement each other and operate collectively to achieve their, possibly conflicting, goals, but which also exhibit an emergent behaviour that is in line with their designers' objectives.	Empirical evidence has shown how students' time management ability and its consequent translation into time allocation between academic and other daily activities may have an impact on students' performance. Given their wide adoption among students, smartphones can help in understanding their behaviour and develop strategies for administrators and academic staff to enact policies that help students learn better strategies for managing their time and their academic workload. SmartUniversity is jointly developed by the Department of Information Engineering and Computer Science and the Department of Sociology and Social Research of the University of Trento. The main goal of the SmartUniversity project is to fill the empirical gap concerning students' time allocation and academic performance by providing a detailed description of how their time management affects their academic achievement.	2016-10	2022-03-14	2018-06	Controlled		EN	https://livepeople.datascientia.eu/workspace/smartuniintwo	University of Trento (IT)	Dataset_S_9	Person_8	Dataset_C_8
STD_P_1	Smart Society / Smart University	Move towards hybrid systems where people and machines tightly work together to build a smarter society. The vision is a new generation of Collective Adaptive Systems where humans and machines synergically complement each other and operate collectively to achieve their, possibly conflicting, goals, but which also exhibit an emergent behaviour that is in line with their designers' objectives.	Empirical evidence has shown how students' time management ability and its consequent translation into time allocation between academic and other daily activities may have an impact on students' performance. Given their wide adoption among students, smartphones can help in understanding their behaviour and develop strategies for administrators and academic staff to enact policies that help students learn better strategies for managing their time and their academic workload. SmartUniversity is jointly developed by the Department of Information Engineering and Computer Science and the Department of Sociology and Social Research of the University of Trento. The main goal of the SmartUniversity project is to fill the empirical gap concerning students' time allocation and academic performance by providing a detailed description of how their time management affects their academic achievement.	2016-10	2022-03-14	2018-06	Controlled		EN	https://livepeople.datascientia.eu/workspace/smartuniintwo	University of Trento (IT)	https://doi.org/10.1140/epjds/s13688-021-00299-2	Person_8	Dataset_C_7
STD_P_1	Smart Society / Smart University	Move towards hybrid systems where people and machines tightly work together to build a smarter society. The vision is a new generation of Collective Adaptive Systems where humans and machines synergically complement each other and operate collectively to achieve their, possibly conflicting, goals, but which also exhibit an emergent behaviour that is in line with their designers' objectives.	Empirical evidence has shown how students' time management ability and its consequent translation into time allocation between academic and other daily activities may have an impact on students' performance. Given their wide adoption among students, smartphones can help in understanding their behaviour and develop strategies for administrators and academic staff to enact policies that help students learn better strategies for managing their time and their academic workload. SmartUniversity is jointly developed by the Department of Information Engineering and Computer Science and the Department of Sociology and Social Research of the University of Trento. The main goal of the SmartUniversity project is to fill the empirical gap concerning students' time allocation and academic performance by providing a detailed description of how their time management affects their academic achievement.	2016-10	2022-03-14	2018-06	Controlled		EN	https://livepeople.datascientia.eu/workspace/smartuniintwo	University of Trento (IT)	https://doi.org/10.1140/epjds/s13688-021-00299-2	Person_8	Dataset_C_8

Figure 18: Project table alignment after Formal Modelling. The green column fields are the data properties, while the light-green ones are object properties. The yellow column name has been modified, from Partner to Affiliation, to match the already existing property schema:affiliation reference ontology

ds_Study_Project is a new concept. It is a noun having Project (Planned Undertaking) as parent and with description: a project which results from a research study with a specific purpose, characterised by the provided scientific documentation and the involved partners (affiliation).

ds_Application Project is a new concept. It is a noun having *Project* (Planned Undertaking) as parent with description: a project which results into a software application, characterised by the programming language used, its release version and the involved partners (affiliation).

ds_KGE_Project is a new concept. It is a noun having *Project* (Planned Undertaking) as parent with description: a project that follows the iTelos methodology and results in a knowledge graph, characterised by its informational layers (Knowledge, Language, Data).

dcat_Catalog is a new concept. It is a noun with *Catalog* (a complete list of things, usually arranged systematically) as parent and with description: A curated collection of metadata about resources (datasets).

ds_Dataset is a new concept. It is a noun having *Dataset* (any collection of data) as parent and with description: any collection of data which belongs to a (some) specific informational layer(s): Knowledge, Language, Data. It may be linked to complementary ds_Dataset

instances (Knowledge Resource, Language Resource, Data Resource).

ds_DataScientia is a new concept. It is a noun having *Space* (an area reserved for some particular purpose) as parent and with description: a collaborative people centric data space, supported by a dedicated infrastructure (Open Data Web Portal). The latter stores context-specific catalogs of dataset and projects, aiming at producing quality and reusable data following the iTelos methodology.

schema_Person is a new concept. It is a noun having *Thing* (something not named specifically) as parent and with description: a person (alive, dead, undead or fictional).

ds_Points_Dataset_At_Data_Resource is a new concept. It is a verb having *Connect* (be or become joined or united or linked) as parent and with description: the action of linking a ds_Dataset to its complementary Data resource, if available.

ds_Points_Dataset_At_Knowledge_Resource is a new concept. It is a verb having *Connect* (be or become joined or united or linked) as parent and with description: the action of linking a ds_Dataset to its complementary Knowledge resource, if available.

ds_Points_Dataset_At_Language_Resource is a new concept. It is a verb having *Connect* (be or become joined or united or linked) as parent and with description: the action of linking a ds_Dataset to its complementary Language resource, if available.

dct_hasPart is an adjective matching with the already existing concept *Divided* (separated into parts or pieces).

dct_isPartOf is a verb matching with the already existing concept *divide* (separate into parts or portions).

dcat_dataset is a new concept. It is a verb having *divided* (separated into parts or objects) as parent and with description: Separated into datasets.

dct_publisher is a new concept. It is a verb having *publish* (the act of produce/create/implement) as parent and with description: the act of being produced/created/implemented by someone (passive form).

schema_WorksFor is a verb matching with the already existing concept *serve* (works for

or be servant to).

schema_affiliation is a noun matching with the already existing concept *affiliation* (a social or business relationship).

doap_programming-language is a new concept. It is a noun having *Language* (a systematic means of communicating by the use of sounds or conventional symbols) as parent and with description: a programming language a project is implemented in, or intended for use with.

dct_format is a new concept. It is a noun having *Format* (general appearance of a publication) as parent and with description: The file format, physical medium, or dimension of the resource.

dcat_landingPage is a noun matching with the already existing concept *webpage* (a document connected to the World Wide Web) and viewable by anyone connected to the internet who has a web browser.

foaf_Page is a noun matching with the already existing concept *study* (written document describing the findings of some individual or group).

foaf_Homepage is a noun matching with the already existing concept *homepage* (opening page of a website).

dct_spatial is a noun matching with the already existing concept *place* (any area set aside for a particular purpose).

dct_modified is a new concept, it is a noun having *Date* (the particular day, month, or year, usually according to the Gregorian calendar) that an event occurred) as parent and with description: date on which the resource was changed.

dct_issued is a new concept. It is a noun having *Date* (the particular day, month, or year, usually according to the Gregorian calendar, that an event occurred) as parent and with description: date of formal issuance of the resource.

dct_DateAccepted is a new concept. It is a noun having *Date* (the particular day, month, or year, usually according to the Gregorian calendar, that an event occurred) as parent and with description: date of acceptance of the resource.

dct_Language is a noun matching with the already existing concept *Language* (a systematic means of communicating by the use of sounds or conventional symbols).

schema_hasOccupation is a noun matching with the already existing concept *role* (normal or customary activity of a person in a particular social setting).

schema_email is a noun matching with the already existing concept *address* (The place where a person or organization can be found or communicated with).

schema_author is a verb matching with the already existing concept *author* (be the author of).

dct_license is a noun matching with the already existing concept *license* (a legal document giving official permission to do something).

dct_identifier is a noun matching with the already existing concept *identifier* (a symbol that establishes the identity of the one bearing it).

schema_identifier is a verb matching with the already existing concept *identifier* (a symbol that establishes the identity of the one bearing it).

dcat_theme is a noun matching with the already existing concept *theme* (a knowledge domain that you are interested in or are communicating about).

dct_description is a noun matching with the already existing concept *descriptio* (a statement that represents something in words).

dct_accessRights is a noun matching with the already existing concept *access* (the right to obtain or make use of or take advantage of something as services or membership).

doap_release is a new concept. It is a noun having *Adaptation* (a written work as a novel that has been recast in a new form) as parent and with description: the released version of a software applicative.

ds_Purpose is a noun matching with the already existing concept *objective* (the goal intended to be attained).

schema_name is a noun matching with the already existing concept *name* (a language unit by which a person or thing is known).

dct_title is a noun matching with the already existing concept *title* (the name of a work of art or literary composition, etc).

ds_layer is a new concept. It is a noun having *layer* (a resource type) as parent and with description: level(s) of representation diversity (Knowledge, Language, Data).

7.7 Open issues

We could merge the three object properties ds_points_Dataset_at_Knowledge_Resource, ds_points_Dataset_at_Data_Resource and ds_points_Dataset_at_Language_Resource into one generic ds_points_at property, since the domain and range of these three properties is ds:Dataset. DCAT-AP 2 provides dct:relation to account for related resources, so we could also reuse an existing relation for this object property. The property dct:conformsTo is also available to refer to a linked established schema, which would correspond to our ds_points_Dataset_at_Knowledge_Resource. However, we would loose the possibility to distinguish them at first glance. If a user wants to know to which other resources the dataset is connected to, he/she would know after following the link, by checking the metadata layer, provided for the dataset etype. By keeping three different properties, the user would know directly which kind of resource is the one linked to the current dataset.

8 KGC

As said before (3), finding suitable data for our needs resulted to be quite challenging especially due to the fact that the portal we worked on (Data Scientia) was under construction.
What we did was creating our own tables using the most precise data we were able to find, adding sometimes some fictional once to cover specific cases we needed, but weren't able to find elsewhere.

8.1 Identity problem

It is clear that **schema_Person_GID-300020** is the most connected etype in our KG since both dct_hasPart_GID-95878 and dct_publisher_GID-300025 have it as range. Through this two object properties Person is connected to every other etype in our KG and this may lead to some difficulties trying to adapt a unique identification method to all the tables. However, since we generated the data ourselves, we avoided the problem during the labels construction phase creating unique identifiers each time we found useful to add a person for a new and specific purpose. This reasoning applies to our etypes, which have been labeled with unique identifiers since the beginning of our project, avoiding any kind of identity problems.

8.2 Entity Matching

In the same way we anticipated the entity matching process while building our tables. Moreover we started from a personal representation we gave to our etypes through object properties and data properties and then we integrated our data description with more specific metadata standardization we found. In this way we handled entity matching with ease. Furthermore is also relevant to enlighten that, in order to keep the data as clean as possible, we decided when possible to avoid to fill void data we were not able to find and this might result in lack of data for specific cases. Anyway we think that in case more data will be available, the hardest obstacle to overcome will be finding similar metadata annotations in many different datasets, however the deep and focused analysis we made on standardization and relevant property detection should be enough to overcome this problem, since we expect that the most important metadata we choose to keep for this project are necessary for the basic definition of the etypes manipulated in this work.

8.3 KG's Evaluation

Of course our KG evaluation is biased. Having produced our own data, is easy to say that our project works perfectly and cover all the needs exposed by the previously submitted competency questions. However we have to highlight that we built plausible and various tables and we expect the model to work in the same way with biggest and more complex datasets.

9 Outcome Exploitation

9.1 Entity Graph analysis

The Metadata project final outcome, which is the Data Scientia Entity Graph (Figure 19), is obtained by the integration of the EType Graph, UKC-aligned, derived from *Protégé*, with the previously generated etype data tables in *KarmaLinker*. The resulting *ttl* files, *Data_Scientia*, *Catalog*, *Dataset*, *Application_Project*, *KGE_Project*, *Study_Project*, and *Person*, along with the corresponding *csv* data tables, are imported in the *GraphDB* metadata project repository to compose the final Data Scientia EG.

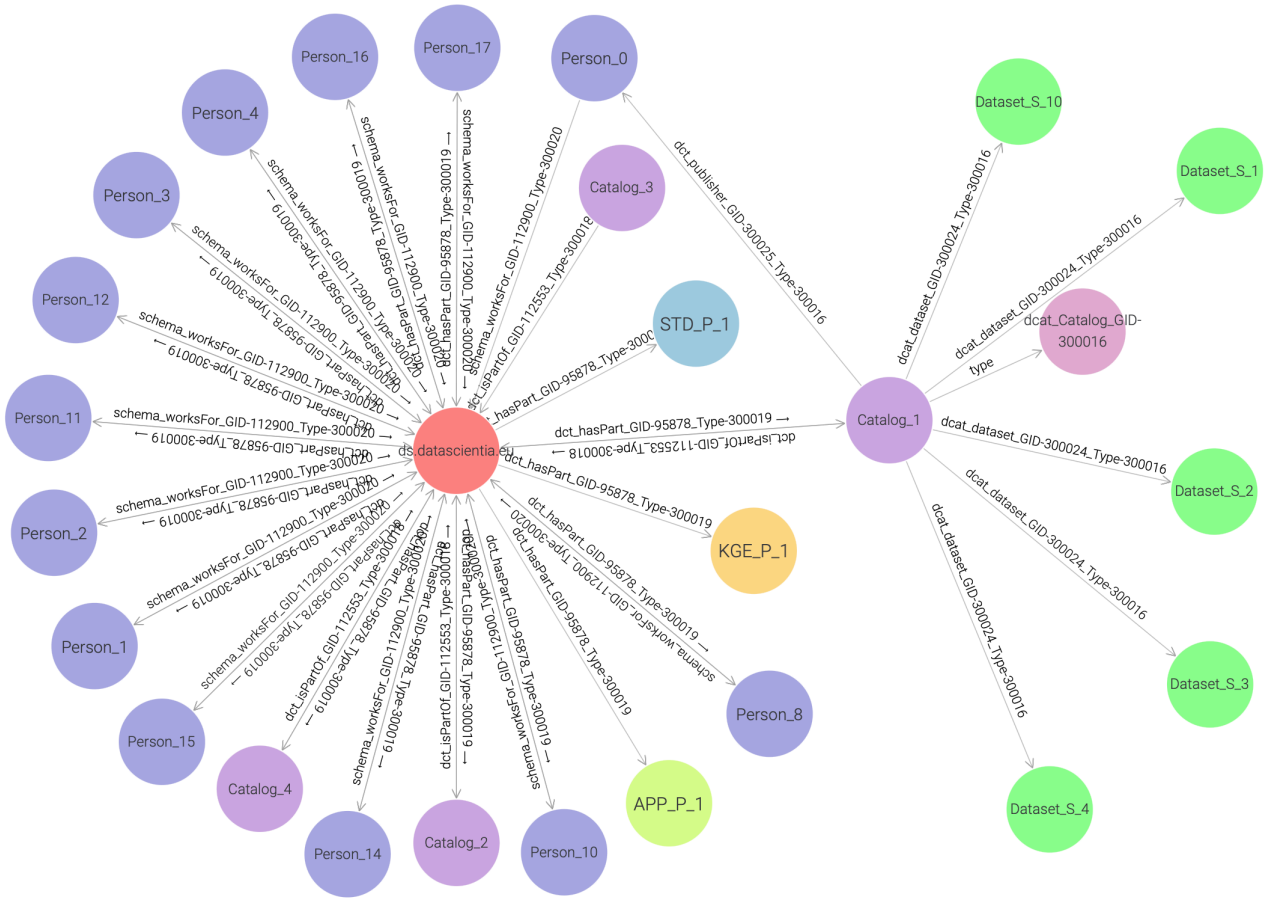


Figure 19: Final Metadata ETG snapshot from GraphDB. All the 7 etypes are included in the sub-graph generated by searching in the Visual Graph box the RDF resource `https://ds.datascientia.eu`

The EG is made of 7 different etypes:

- ds:Data_Scientia_GID-300019 (1 instance);
- dcat:Catalog_GID-300016 (4 instances);

- ds:Dataset_GID-300018 (11 instances);
- ds:Application_Project_GID-300014 (1 instance);
- ds:KGE_Project_GID-300017 (1 instance);
- ds:Study_Project_GID-300013 (1 instance);
- schema:Person_GID-300020 (23 instances).

The number of instances is limited since the *csv* tables were synthetically generated by hand, and they are just examples or placeholders for the actual data that the web portal will gather in the future, being currently under construction.

GraphDB maps an additional class (Figure 20), named ds:Project (3 instances), the parent of ds:Application_Project, ds:KGE_Project and ds:Study_Project, with 29 total (incoming + outgoing) links (Figure 21), respectively 15 towards ds:Dataset_GID-300018 (⇆), 11 towards schema:Person_GID-300020 (→) and 3 towards ds:Data_Scientia_GID-300019 (←). However, we decided not to include it as an etype during the Formal Modelling phase, since the Data Scientia web portal collects three specific types of projects, that should be considered unique etypes.

The highest ranking class for the number of links (86) is ds:Dataset_GID-300018, with 26 links towards schema:Person_GID-300020 (→), 12 towards itself (⇆), 15 towards ds:Project (⇆), 6 towards ds:KGE_Project_GID-300017 (→), 4 towards ds:Study_Project_GID-300013 (→), 1 towards ds:Application_Project_GID-300014 (→) and 10 towards dcat:Catalog_GID-300016 (←). These statistics match the corresponding *csv* data tables, especially considering the self-loop links that account for the bottom-up reconstruction of a KG from its separate informational layers: language, data and knowledge.

The class schema:Person_GID-300020 ranks at the second position for the number of links (79): 38 towards ds:Data_Scientia_GID-300019 (⇆), 26 towards ds:Dataset_GID-300018 (←), 11 towards ds:Project (←) and 4 towards dcat_Catalog_GID-300016 (←). These results validate our output graph since Person appears as a column in the above mentioned etype data tables, acting as both an author and a member of the Data Scientia portal.

The EG shows 25 data properties, in line with the Formal Modelling phase:

- **schema:author_GID-108806_Type-300020** (of schema:Person);
- **schema:email_GID-45803_Type-300020** (of schema:Person);
- **schema:hasOccupation_GID-3742_Type-300020** (of schema:Person);
- **schema:identifier_GID-39085_Type-300020** (of schema:Person);
- **schema:name_GID-2_Type-300020** (of schema:Person);

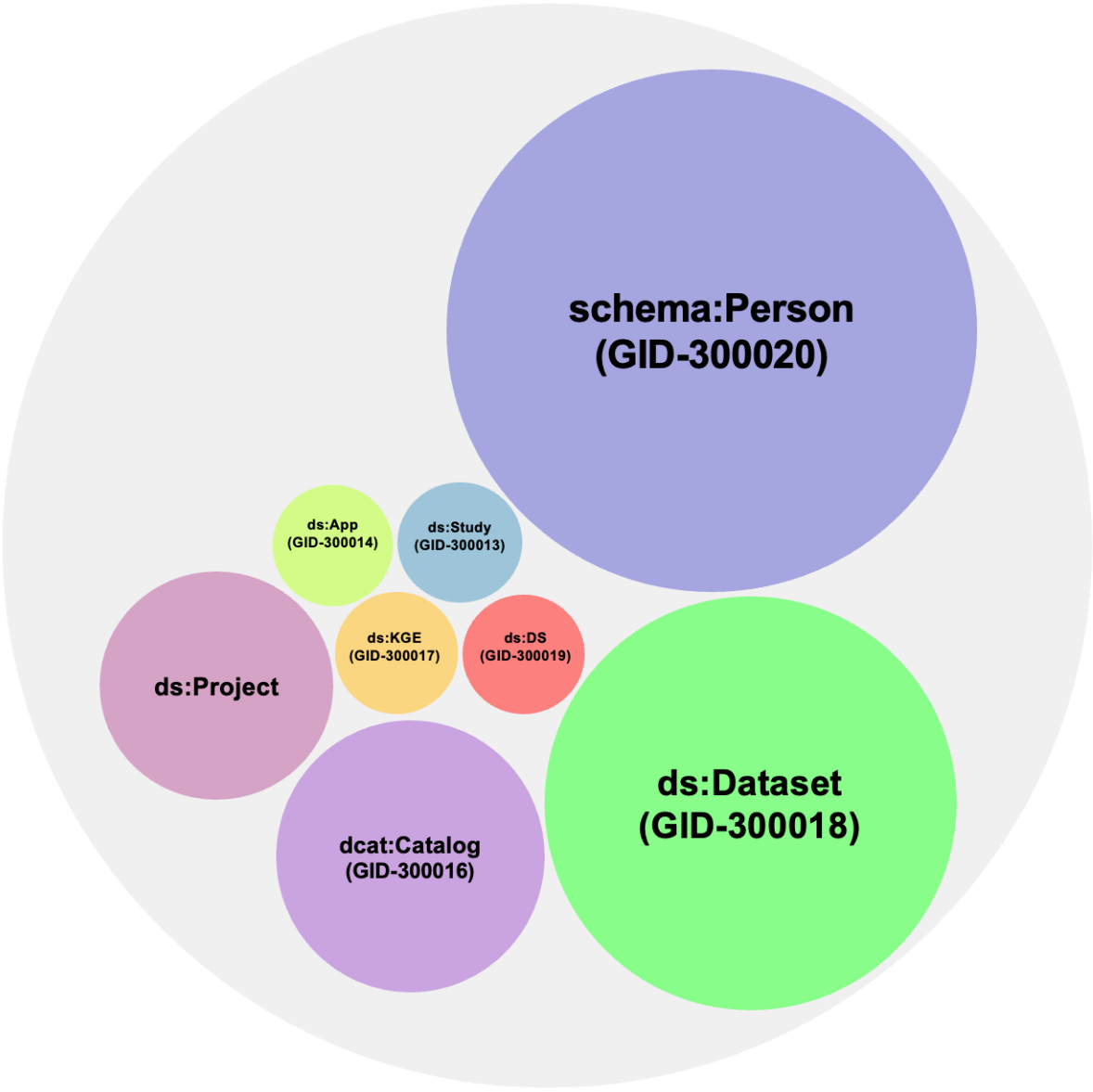


Figure 20: Class hierarchy. GraphDB represents as circles the identified classes from the data integration phase: Person, Dataset, Catalog, Data_Scientia, Application_Project, KGE_Project, Study_Project and Project, whose radius depends on the number of instances present per class.

- **dcat:theme_GID-32585_Type-300018** (of ds:Dataset);
- **dct:format_GID-300027_Type-300018** (of ds:Dataset);
- **dct:spatial_GID-45911_Type-300018** (of ds:Dataset);
- **dct:accessRights_GID-28487** (of ds:Project);
- **ds:purpose_GID-32512** (of ds:Project);
- **doap:programming-language_GID-300026_Type-300014** (of ds:Application Project);

Class	Links	Direction
http://knowdive.disi.unitn.it/etype#schema_Person_GID-300020	79	⇆
http://knowdive.disi.unitn.it/etype#ds_Dataset_GID-300018	86	⇆
http://knowdive.disi.unitn.it/etype#ds_Data_Scientia_GID-300019	51	⇆
http://knowdive.disi.unitn.it/etype#ds_Project	29	⇆
http://knowdive.disi.unitn.it/etype#dcat_Catalog_GID-300016	21	⇆
http://knowdive.disi.unitn.it/etype#ds_KGE_Project_GID-300017	7	←
http://knowdive.disi.unitn.it/etype#ds_Study_Project_GID-300013	5	←
http://knowdive.disi.unitn.it/etype#ds_Application_Project_GID-300014	2	←

Figure 21: Class relationships. For each class identified by GraphDB, the number of the total (incoming + outgoing) links and their direction is provided.

- **doap:release_GID-300031_Type-300014** (of ds:Application Project);
- **foaf:page_GID-38788_Type-300013** (of ds:Study Project);
- **ds:layer_GID-300032_Type-300017** (of ds:Dataset and ds:KGE Project);
- **dcat:landingPage_GID-34124** (of ds:Dataset and ds:Project);
- **dct:dateAccepted_GID-300030** (of ds:Dataset and ds:Project);
- **dct:modified_GID-300028** (of ds:Dataset and ds:Project);
- **foaf:homepage_GID-34125_Type-300019** (of dcat:Catalog and ds:Data Scientia);
- **dct:title_GID-34045_Type-300016** (of dcat:Catalog, ds:Project and ds:Dataset);
- **dct:description_GID-3_Type-300016** (of dcat:Catalog, ds:Project and ds:Dataset);
- **dct:identifier_GID-39085_Type-300016** (of dcat:Catalog, ds:Project and ds:Dataset);
- **dct:language_GID-33764_Type-300016** (of dcat:Catalog, ds:Project and ds:Dataset);
- **dct:license_GID-35108_Type-300016** (of dcat:Catalog, ds:Project and ds:Dataset);
- **dct:issued_GID-300029_Type-300016** (of dcat:Catalog, ds:Project and ds:Dataset);
- **schema:affiliation_GID-74059_Type-300013** (of schema:Person, ds:Study Project and ds:Application Project).

The EG takes into account 8 object properties, matching the Fornal Modelling phase:

- **dct:hasPart_GID-95878_Type-300019**
  (between ds:DataScientia and ds:Project, schema:Person, dcat:Catalog);
- **dcat:dataset_GID-300024_Type-300016**
  (subclass of dct:hasPart, between dcat:Catalog, ds:Project and ds:Dataset);

- **dct:isPartOf_GID-112553_Type-300018**
  (between ds:Dataset, dcat:Catalog and ds:Project, ds:DataScientia);
- **dct:publisher_GID-300025_Type-300016**
  (between ds:Project, dcat:Catalog, ds:Dataset and schema:Person);
- **ds:points_Dataset_At_Data_Resource_GID-300021_Type-300018**
  (between ds:Dataset and ds:Dataset);
- **ds:points_Dataset_At_Knowledge_Resource_GID-300022_Type-300018**
  (between ds:Dataset and ds:Dataset);
- **ds:points_Dataset_At_Language_Resource_GID-300023_Type-300018**
  (between ds:Dataset and ds:Dataset);
- **schema:worksFor_GID-112900_Type-300020**
  (between schema:Person and ds:DataScientia).

9.2 SPARQL validation

In order to verify that the EG graph is searchable by an external user to gather metadata information about the Data Scientia web portal available etypes and entities, the first two competency questions, about the civil engineer Giulia and the undergraduate student Alessandra are translated into SPARQL queries, being these CQs complex and tackling all the main features of our final EG. These queries are performed in the GraphDB SPARQL Query interface. Both the query code and result are shown to account for a successful fullfilling of the user needs and purpose.

From the catalog *Live Data* Giulia can browse all available data resources:

Query 1A

```
PREFIX ds: <http://knowdive.disi.unitn.it/etype#>

SELECT ?data_resource  WHERE {
    ?catalog a ds:dcat_Catalog_GID-300016 .
    ?catalog ds:dct_title_GID-34045_Type-300016 "Live Data" .
    ?data_resource ^ds:dcat_dataset_GID-300024_Type-300016 ?catalog .
}
```

Being each dataset associated with its theme and description, she can easily understand if these resources are compliant with her needs:

	data_resource
1	http://knowdive.disi.unitn.it/tables/Dataset_S_1
2	http://knowdive.disi.unitn.it/tables/Dataset_S_2
3	http://knowdive.disi.unitn.it/tables/Dataset_S_3
4	http://knowdive.disi.unitn.it/tables/Dataset_S_4
5	http://knowdive.disi.unitn.it/tables/Dataset_S_10

Figure 22: **Query 1A** result (Response time: 0.1s).

Query 1B

```
PREFIX ds: <http://knowdive.disi.unitn.it/etype#>

SELECT ?data_resource ?theme ?description WHERE {
    ?catalog a ds:dcat_Catalog_GID-300016 .
    ?catalog ds:dct_title_GID-34045_Type-300016 "Live Data" .
    ?data_resource ^ds:dcat_dataset_GID-300024_Type-300016 ?catalog .
    ?data_resource ds:dcat_theme_GID-32585_Type-300018 ?theme .
    ?data_resource ds:dct_description_GID-3_Type-300016 ?description .
}
```

	data_resource	theme	description
1	http://knowdive.disi.unitn.it/tables/Dataset_S_1	"Transportation"	"Urban and suburban public transport data in GTFS format. Main information available: list of stops (georeferenced), list of lines, routes and arrival and departure times."
2	http://knowdive.disi.unitn.it/tables/Dataset_S_2	"Transportation"	"Real-time data, coming from the e-Motion project, relating to the availability of bicycles at bikesharing stations in the municipalities of Pergine Valsugana"
3	http://knowdive.disi.unitn.it/tables/Dataset_S_3	"Transportation"	"Real-time data, coming from the e-Motion project, relating to the availability of bicycles at bikesharing stations in the municipalities of Lavis"
4	http://knowdive.disi.unitn.it/tables/Dataset_S_4	"Transportation"	"Real-time data, coming from the e-Motion project, relating to the availability of bicycles at bikesharing stations in the municipalities of Lavis, Mezzolombardo, Mezzocorona, Pergine Valsugana, Rovereto, San Michele all'Adige, Trento"
5	http://knowdive.disi.unitn.it/tables/Dataset_S_10	"University"	"A sample of variables available in the dataset"

Figure 23: **Query 1B** result (Response time: 0.1s).

After choosing a dataset, she can also know right away the dataset language, format and license without having to scan the dataset itself row by row.

Query 1C

```
PREFIX ds: <http://knowdive.disi.unitn.it/etype#>
PREFIX dst: <http://knowdive.disi.unitn.it/tables/>

SELECT  ?language ?format ?license WHERE {
    dst:Dataset_S_1 ds:dct_language_GID-33764_Type-300016 ?language .
    dst:Dataset_S_1 ds:dct_format_GID-300027_Type-300018 ?format .
    dst:Dataset_S_1 ds:dct_license_GID-35108_Type-300016 ?license .
}
```

	language	format	license
1	"EN"	"CSV-GTFS"	"CC-BY 2.5"

Figure 24: **Query 1C** result (Response time: 0.1s).

Giulia can filter data resources according to their geography: each dataset has its geographical metadata; if she finds a resource for Trentino, she may be interested in a dataset which provides the same information, but for a different region, e.g Marche, for the sake of comparison.

Query 1D

```
PREFIX ds: <http://knowdive.disi.unitn.it/etype#>

SELECT ?data_resource ?region WHERE {
    ?catalog a ds:dcat_Catalog_GID-300016 .
    ?catalog ds:dct_title_GID-34045_Type-300016 "Live Data" .
    ?data_resource ^ds:dcat_dataset_GID-300024_Type-300016 ?catalog .
    ?data_resource ds:dct_spatial_GID-45911_Type-300018 ?region
}
```

	data_resource	region
1	http://knowdive.disi.unitn.it/tables/Dataset_S_1	"Trentino (IT)"
2	http://knowdive.disi.unitn.it/tables/Dataset_S_2	"Trentino (IT)"
3	http://knowdive.disi.unitn.it/tables/Dataset_S_3	"Trentino (IT)"
4	http://knowdive.disi.unitn.it/tables/Dataset_S_4	"Trentino (IT)"
5	http://knowdive.disi.unitn.it/tables/Dataset_S_10	"Trentino (IT)"

Figure 25: **Query 1D** result (Response time: 0.1s).

Also, if she wants to go beyond the chosen data resource, she can see if there is/are any

knowledge and/or language resources attached to it. In this way, she can analyse the underlying schema that gives structure and meaning to data and check if the language resources can be re-used in her research study.

Query 1E

```
PREFIX ds: <http://knowdive.disi.unitn.it/etype#>
PREFIX dst: <http://knowdive.disi.unitn.it/tables/>

SELECT ?knowledge_resource ?language_resource WHERE {
    dst:Dataset_S_1
     ↪  ds:ds_points_Dataset_At_Knowledge_Resource_GID-300022_Type-300018
     ↪  ?knowledge_resource .
    OPTIONAL{
        dst:Dataset_S_1
         ↪  ds:ds_points_Dataset_At_Language_Resource_GID-300023_Type-300018
         ↪  ?language_resource
    }
}
```

	knowledge_resource	language_resource
1	http://knowdive.disi.unitn.it/tables/Dataset_S_5	

Figure 26: **Query 1E** result (Response time: 0.1s).

Giulia can check if there are existing projects tackling her same domain, by performing a different kind of search in the web portal: she can look at complete projects with a purpose similar to hers: these projects contain all the informational layers and could be referenced by Giulia in her study.

Query 1F

```
PREFIX ds: <http://knowdive.disi.unitn.it/etype#>

SELECT ?project ?purpose ?layer  WHERE {
    ?project a ds:ds_Project .
    ?project ds:ds_purpose_GID-32512 ?purpose .
    ?project ds:ds_layer_GID-300032_Type-300017 ?layer
    FILTER (regex(?purpose, "[Tt]ransportation"))

}
```

	project	purpose	layer
1	http://knowdive.disi.unitn.it/tables/KGE_P_1	"With the development of big data technology and cloud storage technology, we are in an era of rapid increase in information, with countless data or knowledge. How to manage these data and achieve more efficient sharing and utilization is a filed that many researchers are exploring, that is, to fulfill the integration of knowledge and data in specific, rather than leaving information be unorganized. This report focuses on integrating all the public transportation as well as sharing vehicles information within Trentino so that the more complete transport information system could help people make a better decision and save time or money as much as possible. Specifically, we pay attention to the application of vehicles that GTFS has not covered, such as sharing bikes, sharing cars, and so on so forth, which is added to the system and then residents have more choices to determine paths."	"Data; Knowledge; Language"

Figure 27: **Query 1F** result (Response time: 0.1s).

She could also browse them to see their documentation and datasets, if present.

Query 1G

```
PREFIX ds: <http://knowdive.disi.unitn.it/etype#>
PREFIX dst: <http://knowdive.disi.unitn.it/tables/>

SELECT ?documentation ?dataset WHERE {
        dst:KGE_P_1 ds:dcat_landingPage_GID-34124 ?documentation .
        dst:KGE_P_1 ds:dcat_dataset_GID-300024_Type-300016 ?dataset .
}
```

	documentation	dataset
1	"https://carlocorradini.github.io/Trentino_Transportation/"	http://knowdive.disi.unitn.it/tables/Dataset_C_6

Figure 28: **Query 1G** result (Response time: 0.1s).

A more sophisticated search could involve not only the dataset related to the chosen project and its corresponding layer/s, but also the dataset linked resources if any, so Giulia would be able to reconstruct the Knowledge Graph bottom-up.

Query 1H

```
PREFIX ds: <http://knowdive.disi.unitn.it/etype#>
PREFIX dst: <http://knowdive.disi.unitn.it/tables/>

SELECT ?dataset ?layer ?language_res ?data_res  ?knowledge_res WHERE {
    dst:KGE_P_1 ds:dcat_dataset_GID-300024_Type-300016 ?dataset .
    ?dataset ds:ds_layer_GID-300032_Type-300017 ?layer
    OPTIONAL {
        ?dataset
        ↪ ds:ds_points_Dataset_At_Data_Resource_GID-300021_Type-300018
        ↪ ?data_res .
    }
    OPTIONAL {
        ?dataset
        ↪ ds:ds_points_Dataset_At_Knowledge_Resource_GID-300022_Type-300018
        ↪ ?knowledge_res .
    }
    OPTIONAL {
        ?dataset
        ↪ ds:ds_points_Dataset_At_Language_Resource_GID-300023_Type-300018
        ↪ ?language_res .
    }
}
```

	dataset	layer	language_res	data_res	knowledge_res
1	http://knowdive.disi.unitn.it/tables/Dataset_C_6	"Data; Knowledge"		http://knowdive.disi.unitn.it/tables/Dataset_S_1	http://knowdive.disi.unitn.it/tables/Dataset_S_5
2	http://knowdive.disi.unitn.it/tables/Dataset_C_6	"Data; Knowledge"		http://knowdive.disi.unitn.it/tables/Dataset_S_2	http://knowdive.disi.unitn.it/tables/Dataset_S_5
3	http://knowdive.disi.unitn.it/tables/Dataset_C_6	"Data; Knowledge"		http://knowdive.disi.unitn.it/tables/Dataset_S_3	http://knowdive.disi.unitn.it/tables/Dataset_S_5

Figure 29: **Query 1H** result (Response time: 0.1s).

From the catalog Live People Alessandra could browse a large amount of collected data in the field of human behavior and social interactions.

Query 2A

```
PREFIX ds: <http://knowdive.disi.unitn.it/etype#>

SELECT ?catalog ?dataset  WHERE {
    ?catalog a ds:dcat_Catalog_GID-300016 .
    ?catalog ds:dct_title_GID-34045_Type-300016 "Live People" .
    ?catalog ds:dcat_dataset_GID-300024_Type-300016 ?dataset
}
```

	catalog	dataset
1	http://knowdive.disi.unitn.it/tables/Catalog_4	http://knowdive.disi.unitn.it/tables/Dataset_C_7
2	http://knowdive.disi.unitn.it/tables/Catalog_4	http://knowdive.disi.unitn.it/tables/Dataset_C_8

Figure 30: **Query 2A** result (Response time: 0.1s).

Each available dataset could be explored through its metadata: title, creation date, and layer, so Alessandra can immediately know if the dataset she is looking at contains actual data, schema representations or vocabularies and if the dataset contains up-to-date information.

Query 2B

```
PREFIX ds: <http://knowdive.disi.unitn.it/etype#>

SELECT ?catalog ?dataset ?title ?creation_date ?layer  WHERE {
    ?catalog a ds:dcat_Catalog_GID-300016 .
    ?catalog ds:dct_title_GID-34045_Type-300016 "Live People" .
    ?catalog ds:dcat_dataset_GID-300024_Type-300016 ?dataset .
    ?dataset ds:dct_title_GID-34045_Type-300016 ?title .
    ?dataset ds:ds_layer_GID-300032_Type-300017 ?layer .
    ?dataset ds:dct_issued_GID-300029_Type-300016 ?creation_date .
}
ORDER BY desc(?creation_date)
```

	catalog	dataset	title	creation_date	layer
1	http://knowdive.disi.unitn.it/tables/Catalog_4	http://knowdive.disi.unitn.it/tables/Dataset_C_8	"SmartUnitn2"	"2016-11"	"Data; Knowledge"
2	http://knowdive.disi.unitn.it/tables/Catalog_4	http://knowdive.disi.unitn.it/tables/Dataset_C_7	"SmartUnitn1"	"2016-10"	"Data; Knowledge"

Figure 31: **Query 2B** result (Response time: 0.1s).

Alessandra might want to explore existing study projects which share her same purpose: Data Scientia would allow her to look at study projects characterized by the presence of the metadata documentation. The latter would redirect Alessandra to the scientific development and explanation of the project, helping her both with theory and bibliography construction: many useful papers and authors would be made accessible to her.

Query 2C

```
PREFIX ds: <http://knowdive.disi.unitn.it/etype#>

SELECT ?Study_Project ?documentation ?author_name ?author_email  WHERE {
    ?Study_Project a ds:ds_Study_Project_GID-300013 .
    ?Study_Project ds:foaf_page_GID-38788_Type-300013 ?documentation .
    ?Study_Project ds:dct_publisher_GID-300025_Type-300016 ?author .
    ?author ds:schema_name_GID-2_Type-300020 ?author_name .
    ?author ds:schema_email_GID-45803_Type-300020 ?author_email
}
```

	Study_Project	documentation	author_name	author_email
1	http://knowdive.disi.unitn.it/tables/STD_P_1	"Dataset_S_9"	"Fausto Giunchiglia"	"fausto.giunchiglia@unitn.it"
2	http://knowdive.disi.unitn.it/tables/STD_P_1	"Dataset_S_9"	"Ivano Bison"	"ivano.bison@unitn.it"
3	http://knowdive.disi.unitn.it/tables/STD_P_1	"Dataset_S_9"	"Matteo Busso"	"matteo.busso@unitn.it"
4	http://knowdive.disi.unitn.it/tables/STD_P_1	"Dataset_S_9"	"Marcelo Dario Rodas Britez"	"marcelo.rodasbritez@unitn.it"
5	http://knowdive.disi.unitn.it/tables/STD_P_1	"https://doi.org/10.1140/epjds/s13688-021-00299-2"	"Fausto Giunchiglia"	"fausto.giunchiglia@unitn.it"
6	http://knowdive.disi.unitn.it/tables/STD_P_1	"https://doi.org/10.1140/epjds/s13688-021-00299-2"	"Ivano Bison"	"ivano.bison@unitn.it"
7	http://knowdive.disi.unitn.it/tables/STD_P_1	"https://doi.org/10.1140/epjds/s13688-021-00299-2"	"Matteo Busso"	"matteo.busso@unitn.it"
8	http://knowdive.disi.unitn.it/tables/STD_P_1	"https://doi.org/10.1140/epjds/s13688-021-00299-2"	"Marcelo Dario Rodas Britez"	"marcelo.rodasbritez@unitn.it"
9	http://knowdive.disi.unitn.it/tables/STD_P_1	"https://doi.org/10.1007/s42979-021-00714-5"	"Fausto Giunchiglia"	"fausto.giunchiglia@unitn.it"
10	http://knowdive.disi.unitn.it/tables/STD_P_1	"https://doi.org/10.1007/s42979-021-00714-5"	"Ivano Bison"	"ivano.bison@unitn.it"
11	http://knowdive.disi.unitn.it/tables/STD_P_1	"https://doi.org/10.1007/s42979-021-00714-5"	"Matteo Busso"	"matteo.busso@unitn.it"
12	http://knowdive.disi.unitn.it/tables/STD_P_1	"https://doi.org/10.1007/s42979-021-00714-5"	"Marcelo Dario Rodas Britez"	"marcelo.rodasbritez@unitn.it"
13	http://knowdive.disi.unitn.it/tables/STD_P_1	"https://doi.org/10.1016/j.chb.2017.12.041"	"Fausto Giunchiglia"	"fausto.giunchiglia@unitn.it"

Figure 32: **Query 2C** result (Response time: 0.1s).

Also, she could explore the Data Scientia web portal looking for professors and PhD students that work on the KnowDive research group, of which his thesis' supervisor is the founding member, to look at their publications.

Query 2D

```
PREFIX ds: <http://knowdive.disi.unitn.it/etype#>

PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
SELECT ?name ?role ?publication WHERE {
   ?data_scientia a ds:ds_Data_Scientia_GID-300019 .
   ?person ds:schema_worksFor_GID-112900_Type-300020 ?data_scientia .
   ?person ds:schema_name_GID-2_Type-300020 ?name .
   ?person ds:schema_hasOccupation_GID-3742_Type-300020 ?role .
   ?person ds:schema_author_GID-108806_Type-300020 ?publication
   FILTER (?role ="Full Professor"^^xsd:string || ?role ="PhD
    ↪  Student"^^xsd:string || ?role ="Assistant Professor"^^xsd:string)
}
```

	name	role	publication
1	"Fausto Giunchiglia"	"Full Professor"	"https://doi.org/10.48550/arXiv.2105.09432"
2	"Fausto Giunchiglia"	"Full Professor"	"https://doi.org/10.48550/arXiv.2105.09418"
3	"Fausto Giunchiglia"	"Full Professor"	"https://doi.org/10.48550/arXiv.2209.14049"
4	"Fausto Giunchiglia"	"Full Professor"	"https://doi.org/10.48550/arXiv.2004.01392"
5	"Fausto Giunchiglia"	"Full Professor"	"https://doi.org/10.48550/arXiv.2211.03009"
6	"Fausto Giunchiglia"	"Full Professor"	"https://doi.org/10.48550/arXiv.2202.12700"
7	"Fausto Giunchiglia"	"Full Professor"	"https://doi.org/10.48550/arXiv.2205.03608"
8	"Ivano Bison"	"Full Professor"	"https://doi.org/10.48550/arXiv.2004.01392"
9	"Ivano Bison"	"Full Professor"	"https://doi.org/10.48550/arXiv.2211.03009"
10	"Ivano Bison"	"Full Professor"	"https://doi.org/10.48550/arXiv.2202.12700"
11	"Simone Bocca"	"PhD Student"	"https://doi.org/10.48550/arXiv.2105.09432"
12	"Simone Bocca"	"PhD Student"	"https://doi.org/10.48550/arXiv.2105.09418"
13	"Simone Bocca"	"PhD Student"	"https://doi.org/10.48550/arXiv.2209.14049"
14	"Mayukh Bagchi"	"PhD Student"	"https://doi.org/10.48550/arXiv.2105.09432"
15	"Mayukh Bagchi"	"PhD Student"	"https://doi.org/10.48550/arXiv.2105.09418"
16	"Mayukh Bagchi"	"PhD Student"	"https://doi.org/10.48550/arXiv.2209.14049"
17	"Matteo Busso"	"PhD Student"	"https://doi.org/10.48550/arXiv.2211.03009"
18	"Matteo Busso"	"PhD Student"	"https://doi.org/10.48550/arXiv.2202.12700"
19	"Gábor Bella"	"Assistant Professor"	"https://doi.org/10.48550/arXiv.2205.03608"

Figure 33: **Query 2D** result (Response time: 0.1s).

Alessandra could also check the number of publications any KnowDive member has.

Query 2E

```
PREFIX ds: <http://knowdive.disi.unitn.it/etype#>

PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
SELECT (count(?publication) as ?count) ?name WHERE {
   ?data_scientia a ds:ds_Data_Scientia_GID-300019 .
   ?person ds:schema_worksFor_GID-112900_Type-300020 ?data_scientia .
   ?person ds:schema_name_GID-2_Type-300020 ?name .
   ?person ds:schema_author_GID-108806_Type-300020 ?publication
}
GROUP BY (?name)
```

	count	name
1	"3"^^xsd:integer	"Simone Bocca"
2	"3"^^xsd:integer	"Mayukh Bagchi"
3	"7"^^xsd:integer	"Fausto Giunchiglia"
4	"1"^^xsd:integer	"Gábor Bella"
5	"3"^^xsd:integer	"Ivano Bison"
6	"1"^^xsd:integer	"Enrico Bignotti"
7	"1"^^xsd:integer	"Mattia Zeni"
8	"1"^^xsd:integer	"Elisa Gobbi"
9	"2"^^xsd:integer	"Matteo Busso"
10	"1"^^xsd:integer	"Marcelo Dario Rodas Britez"
11	"1"^^xsd:integer	"Ronald Chenu-Abente Acosta"
12	"1"^^xsd:integer	"Vincenzo Maltese"

Figure 34: **Query 2E** result (Response time: 0.1s).

From these examples we can conclude that the Metadata project successfully reached the goal of creating a KG for web portal's users to find resources they are curious about and linked references.

10 Conclusions Open Issues

10.1 Conclusion

Supported by regular meetings with our domain expert, we delved into this project with a specific purpose:

*A KG supporting the **Data Scientia** web portal's users to find the most suitable **resource** for their needs, as well as all the resources **linked** by the one searched.*

At first we approached this request from an abstract point of view, thinking about what the key concepts of resource and linking meant and how to properly translate them into etypes or properties of the KG. Then we studied how the under construction Data Scientia web portal was structured, paying attention to the underlying skeleton of the sitemap. We tried to single out the main contents that could be viewed as a resource by an external user, avoiding to get out of context: our purpose was about finding useful metadata to describe the resources and not to describe the resources as self contained objects. We needed to focus on how to portray a meaningful link between datasets that were supposed to be accessible through their metadata, such as layers, without uncovering them. The links were to be interpreted as both meta-links, since they were based on properties of datasets that were as well to be defined, and as actual relationships between resources, upon which a Knowledge Graph could have been built. We realized that we had to build a knowledge graph of metadata, out of which it was possible to reconstruct real world knowledge graphs, arising from the resources collected. It was very challenging to keep the boundaries steady between our foundational knowledge graph of metadata and the actual sitemap of a web portal. Being Computer Science students, we tended to collapse the difference between them because they are both containers and abstractions of resources, but placed on different ontological levels: the web site map only tells you where resources are in a network, and how to reach them, while the KG is necessary to describe the resources for what they were, highlighting their extrinsic and intrinsic properties, supported by a Teleology that was constructed alongside. This parallel step helped us to adjust our data or object properties to fit the needs of a potential user, because categorizations used to describe entities are useless if not shareable and inter-subjective: the meaning embodied in metadata is what the user is looking for; they want to find what they need in the easiest and fastest way to match their purpose: a raw look at data values from a dataset is not worth the effort because data collection is just the starting phase of any project, it does not posses an added value. The added value comes from a potential use that a user sees in the data and that can only be done if the nature of the resource is explicated on a language, knowledge and data level, as our KG built with the iTelos methodology does. We followed iTelos methodology going through all the main steps:

- **Data collection & Purpose Formalization:** we created our data collection and started to analyze the requests and needs established by our purpose alongside with all the metadata we could possibly find useful.
- **Informal Modelling:** we started having a basic understanding of the problem and formu-

lated our teleology, the starting point for our final schema.

- **Formal Modelling:** we got more practical gaining a deep purpose understanding. We build our teleontology aligned with a standardized language maximizing reusability.
- **Data Integration:** we analysed possible problem we could face when manipulating a real and large amount of data and possibly how our KG would work in that case.

Even if the phases are easy to distinguish and sequential, iTelos is a middle-out methodology. This means that is not rare that working deeper in a specific phase of the project can lead to some reconsiderations on previously done job. In this sense we followed strictly the method trying to refine each time at best every aspect of the project and, most importantly, to deeply understand the meanings of each of the sequential phases.

10.2 Further Perspectives

In the end our project covers the purpose needs, but surely will need some updates in order to keep up with the Data Scientia web portal progresses. In our interpretation Project is not an etype since in Data Scientia there are not any project collection (such as catalogue or similar). Still it may be a nice feature since, for now, we see dataset strictly connected either to a project or a catalogue. Another improvement we should consider is defining a better conceptualization for the Person etype. Indeed now we have mixed under the same etype all people and companies independently even if there can be close knowdive partners, or external/third party users. This feature may become important especially if the methodology will grow in popularity and the number of people interested in making data available increase. We have also to highlight that most likely Services will be added to the web portal and maybe a dedicated etype could lead to a better representation of the environment as the purpose requires. Finally we need to consider that we had a specific aim, that is to create a supporting KG for the portal, as we did. In order to do this we created our fictional data. They of course emulate reality, but maybe having the possibility to reach more data could lead to a new phase of backtracking highlighting possible missing necessary metadata as well as some useless ones. Another advantage that access to actual datasets could bring is that the number of missing values in our tables would decrease and patterns could be extracted from the gathered data.

References

[GM67] David M. Griffel and Stuart D. McIntosh. *ADMINS : a progress report*. 1967. url: http://hdl.handle.net/1721.1/82974.

[Bac16] M. Baca. *Introduction to Metadata: Third Edition*. Introduction To. J. Paul Getty Trust, 2016. isbn: 9781606064795. url: https://books.google.it/books?id=xgZVDQAAQBAJ.

[Giu+21] Fausto Giunchiglia et al. *iTelos – Purpose Driven Knowledge Graph Generation*. 2021. doi: 10.48550/ARXIV.2105.09418. url: https://arxiv.org/abs/2105.09418.

[Giu22] Fausto Giunchiglia. *Lecture slides of Knowledge Graph Engineering*. Sept. 2022.

[Giu+22] Fausto Giunchiglia et al. *Popularity Driven Data Integration*. 2022. doi: 10.48550/ARXIV.2209.14049. url: https://arxiv.org/abs/2209.14049.