Part 6
KG Evaluation, Exploitation and Distribution

Part 6.1
KG Evaluation

The KG's evaluation

- **How to evaluate the quality of the final KG, as well as the entire KG construction porcess ?**
- The iTelos methodology's structure includes different **evaluation activities** to be executed at the end of each phase, to check if the intermediate outputs is suitable to be processed by the next phase or it needs to be revised. [42]

[42] In this course, for lack of time, the evaluation is done only at the end of the process.

The KG's evaluation

- iTelos provides different criteria to evaluate the primary and secondary objectives of a process execution.
- The criteria, described below, consider both the Knowledge and Data layer evaluation.
  - **Primary objective - Purpose satisfaction**: How much the final KG is able to satisfy the Competency Queries ?
    - **Knowledge layer**: Evaluation of CQs vs KG's Teleonotlogy
    - **Data layer**: Evaluation of KG connectivity
  - **Secondary objective - Reusability**: How much reusable is the final KG ?
    - **Knowledge layer**: Evaluation of Teleontology vs Reference Ontologies

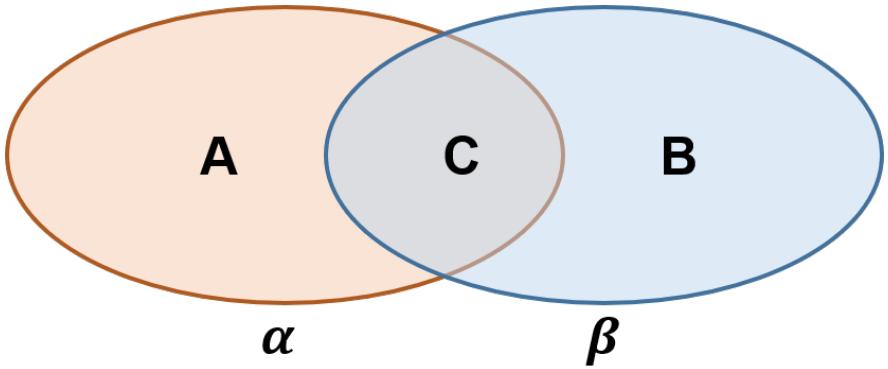
KG's Evaluation

- Knowledge Layer Evaluations
  - Primary goal - Purpose-based evaluation
  - Secondary goal - Reusability evaluation
- Data Layer Evaluations
  - Final KG evaluation
  - KG construction process evaluation

Evaluation metrics

- iTelos provides a set of metrics to be used for the above evaluations.
- Between them one of the most useful is:
  - **Coverage**: How much a portion of knowledge (shaped as etypes and properties) is covered by a KG.
- To evaluate the **Knowledge layer** for the primary and secondary objectives the coverage is used as follows:
  - **Primary objective** (Teleontology vs CQs): How much the Teleontology covers the Entities and properties extracted from the CQs.
  - **Secondary objective** (Teleontology vs Reference Ontologies): How much the Teleontology covers the etypes, and properties, extracted from the reference ontologies.

Metric definitions: Coverage



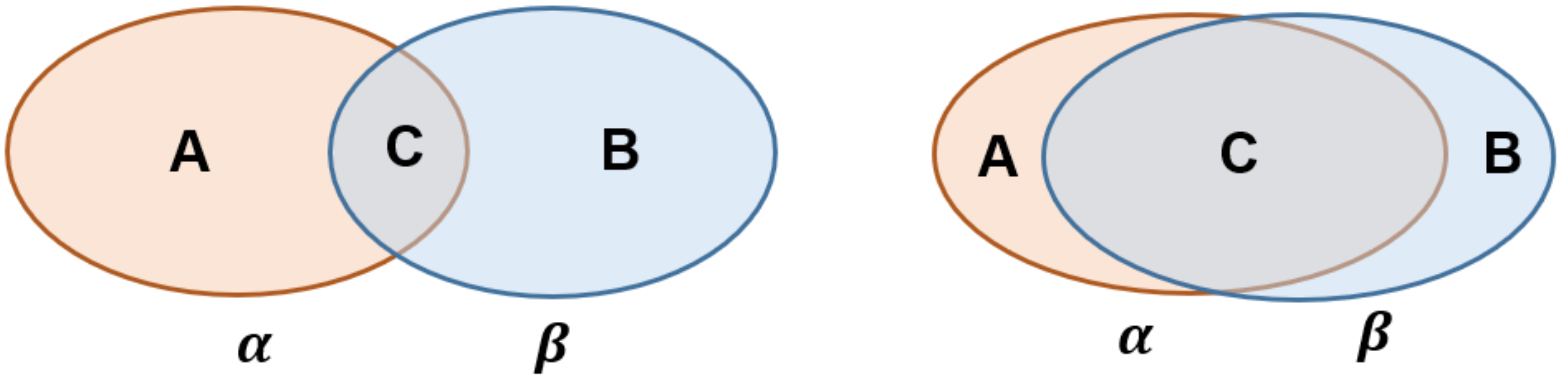
The Coverage is computed as the ratio between the intersection of α and β and the whole α sets:

$$Cov = (\alpha \cap \beta)/\alpha = C/(A + C) \tag{2}$$

Where:

- α is a portion of knowledge to be verified.
- β is the KG's Knowledge layer.

Metric definitions: Coverage (extreme cases)



$$Cov = (\alpha \cap \beta)/\alpha = C/(A + C)$$

$Cov \simeq 0$ $\quad$ $Cov \simeq 1$

Metric definitions: Coverage

About the Coverage used to evaluate KGs: $Cov = (\alpha \cap \beta)/\alpha$

- Values are always within the interval [0,1].
- **High values of Coverage** mean that the KG's knowledge is appropriate for the domain.
- For **low values of Coverage**, we can have two possibilities.
  - The reference schema is not appropriate for the domain and maybe a further lookup should be performed.
  - The domain targeted by the knowledge graph is mostly unexplored.

Teleontology vs CQs - EType level

Given a set of (CQ), the **etype coverage** (Cov_E) of the Teleontology (T) is:

$$Cov_E(CQ_E) = \frac{|CQ_E \cap T_E|}{CQ_E} \tag{3}$$

Where:

- CQ_E is the number of etypes extracted from the CQs.
- T_E is the number of etypes of the Teleontology.

Teleontology vs CQs - Property level

Given a set of (CQ), the **property coverage** (Cov_p) of the Teleontology (T) is:

$$Cov_p(CQ_p) = \frac{|CQ_p \cap T_p|}{CQ_p} \tag{4}$$

Where:

- CQ_p is the number of properties extracted from the CQs.
- T_p is the number of properties of the Teleontology.

Teleontology vs Reference Ontologies (ROs) - EType level

Given a set of (RO), the **etype coverage** (Cov_E) of the Teleontology (T) is:

$$Cov_E(RO_E) = \frac{|RO_E \cap T_E|}{RO_E} \tag{5}$$

Where:

- RO_E is the number of etypes extracted from the ROs.
- T_E is the number of etypes of the Teleontology.

Teleontology vs Reference Ontologies (ROs) - Property level

Given a set of (RO), the **property coverage** (Cov_p) of the Teleontology (T) is:

$$Cov_p(RO_p) = \frac{|RO_p \cap T_p|}{RO_p} \tag{6}$$

Where:

- RO_p is the number of properties extracted from the ROs.
- T_p is the number of properties of the Teleontology.

KG's Evaluation

- Knowledge Layer Evaluations
  - Primary goal - Purpose-based evaluation
  - Secondary goal - Reusability evaluation
- Data Layer Evaluations
  - Final KG evaluation
  - KG construction process evaluation

The KG's evaluation - Data layer

- Evaluating the KG's data layer, aims to understand how **"dense"** or **"connected"** is the KG, at the end of the iTelos process, and during the KG's construction.
- We can evaluate **the KG's connectivity** in two different moments:
  - **On the final KG**: this evaluation aims to understand how much connected is the KG at the end of the process.
  - **During the KG's construction**: this evaluation aims to understand how much each single dataset, handled during the process, improve the connectivity of the final KG.
- **Note**: the improvement of connectivity brought by a single dataset to the KG, can be different when the dataset is added to the partial KG (during construction), respect to the connectivity evaluated over the same dataset's values, over the final KG.
  - The difference is caused by the **entity matching conflicts** and their solutions.

The KG's evaluation - Data layer

- The **connectivity** of a KG can be evaluated over two dimensions:
  - **Entity connectivity**: How much the entities are connected to each other. It evaluates the grade of connection between the different entities in the KG.
  - **Property connectivity**: How much the entities are connected to their properties. It evaluates the grades of connection between each single KG's entity and its properties values.

The KG's evaluation - Data Layer - Final KG

- The Entity and Property Connectivity can be calculated by using the **connectivity matrix**.

	EType A	EType B	EType C	....	EType N
EType A	#	*	*	*	*
EType B	*	#	*	*	*
EType C	*	*	#	*	*
....	*	*	*	#	*
EType N	*	*	*	*	#

- Where the value of cell (X,Y) is:
  - # : (X = Y) the number of **non-null data properties** values, for the all the entities mapped on the EType X (or Y).
  - * : (X ≠Y) the number of **non-null object properties** values, for the object properties having the EType X as domain and the EType Y as range.

The KG's evaluation - Data Layer - Final KG

- The **Entity Connectivity** is calculated EType by EType, as the sum of the **"* values"** for each EType row in the matrix. The resulting row sum, relative to the EType X, is then divided by the number of **object properties** defined for the EType X.
  - The sum of the Entity Connectivity of all the KG's ETypes, defines the values of Entity Connectivity of the whole KG.
- The **Property Connectivity** is calculated EType by EType, dividing the **"# values"** for the cell (X,X) by the number of **data properties** defined for the EType X.
  - The sum of the Property Connectivity of all the KG's ETypes, defines the values of Property Connectivity of the whole KG.

The KG's evaluation - Data Layer - Final KG

- In formulae:
  - **EC(X) Entity Connectivity for the EType X**:

$$EC(X) = \frac{\sum_{Y=1}^{N}(X, Y)}{OP(X)} \tag{7}$$

  - **Where**: (X, Y) is a cell in the connectivity matrix, and OP(X) is the number of object properties of the ETypes X.
  - **EC(KG) Entity Connectivity for the whole KG**:

$$EC(KG) = \sum_{X=1}^{N} EC(X) \tag{8}$$

The KG's evaluation - Data Layer - Final KG

- In formulae:
  - **PC(X) Property Connectivity for the EType X**:

$$PC(X) = \frac{(X, X)}{DP(X)} \tag{9}$$

  - **Where**: (X, Y) is a cell in the connectivity matrix, and DP(X) is the number of object properties of the ETypes X.
  - **PC(KG) Property Connectivity for the whole KG**:

$$PC(KG) = \sum_{X=1}^{N} EC(X) \tag{10}$$

The KG's evaluation - Data Layer - Construction

- To evaluate the connectivity improvement brought by a new dataset that has to be integrated into the KG, we have to consider the following cases.
- It is possible to calculate the **entity and property connectivity** (see previous slides) to measure the impact of new datasets over the KG, in construction.
- **Assumption**: There are, a new dataset D_1 and the partially built graph KG. Moreover, D_1 has an etype E_1, with its property set A_1 and KG has an etype E_2, with its property set A_2.

The KG's evaluation - Data Layer - Construction

- **Case 1**: $[E_1 = E_2]$ The E_1 in D_1 is already present in KG.
- **Consequence**: By integrating D_1 into KG we are increasing the number of entities of E_1, thus **increasing the entity connectivity**.
  - **Case 1.1**: $[A_1 = A_2]$ The etypes share the same set of properties.
  - **Consequence**: Conflicts are possible between the value set of A_1 and A_2.
    - How many conflicts ?
    - How many new entities from D_1 are integrated into the KG ?
    - How many properties, in the property set A_1, with not null values remain after solving such conflicts ?

The KG's evaluation - Data Layer - Construction

- **Case 1**: $[E_1 = E_2]$ The E_1 in D_1 is already present in KG.
- **Consequence**: By integrating D_1 into KG we are increasing the number of entities of E_1, thus **increasing the entity connectivity**.
  - **Case 1.2**: $[A_1 \neq A_2]$ The etypes have different sets of properties.
  - **Consequence**: There are no conflicts between the value set of A_1 and A_2, and there is a greater increase of the integration over the etype E_1. Notice how in this case also **the property connectivity increases**.
    - How many new entities from D_1 are integrated into the KG ?
    - How many properties, in the property set $A_1 \cup A_2$, with not null values remain after the integration of D_1 ?

The KG's evaluation - Data Layer - Construction

- **Case 2**: $[E_1 \neq E_2]$ The E_1 in D_1 is not yet present in KG.
- **Consequence**: By integrating D_1 into KG we are increasing the number of etypes of KG.
  - **Case 2.1**: E_1 and E_2 are linked by at least one object property.
  - **Consequence**: The resulting KG, after the integration of D_1, is connected.
    - How many connections ?
    - How many entities of E_1 have not null values for the object properties linking E_1 with KG ?

The KG's evaluation - Data Layer - Construction

- **Case 2**: $[E_1 \neq E_2]$ The E_1 in D_1 is not yet present in KG.
- **Consequence**: By integrating D_1 into KG we are increasing the number of etypes of KG.
  - **Case 2.2**: There are no object properties linking E_1 and E_2.
  - **Consequence**: The resulting KG, after the integration of D_1, is not connected.
  - The integration of D_1 doesn't increase the connectivity, thus the information carried by D_1 cannot be reached by the KG.

Part 6.2
KG Exploitation

Exploiting the KG

- GraphDB
- SPARQL

What is a GraphDB?

GraphDB

It is an enterprise ready Semantic Graph Database developed by Ontotext and it is compliant with W3C Standards.

GraphDB is just one implementation, there is a lot of competition:

- Neo4J
- TerminusDB
- ArangoDB
- OpenRDF (previously Sesame)
- many other...

How to install GraphDB ?

- Download the package from `https://graphdb.ontotext.com`;
- Use Docker `$ docker run -d -p 127.0.0.1:7200:7200 ontotext/graphdb:10.1.1-arm64` If using ARM you need the proper image.

There is a *free version*, but also two proprietary license that include additional functionalities. For the course the free version is enough.

Concepts

- Database → Repository;
- Record → Node (aka Entity);
- Field → Triple;
- SQL → SPARQL;
- Table → Graph.

Datasets

- Documentation of the Dataset here (https://platform.ontotext.com/semantic-objects/datasets/star-wars.html)
- Dataset in the documentation or here (http://shorturl.at/exLOP)

Tricks

- GraphDB allows you to upload up to 200MB.

```
$ python -m SimpleHTTPServer 8081
$ python3 -m http.server --bind 127.0.0.1 8080
```

Exploiting the KG

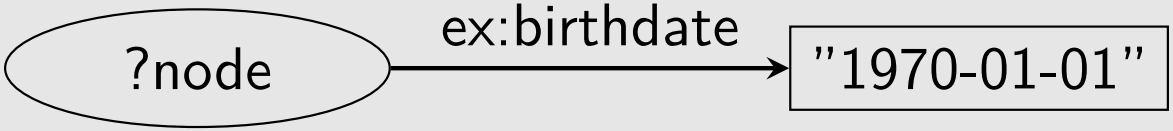
- GraphDB
- SPARQL

SPARQL

Definition

SPARQL is a **protocol** and a declarative **query language** for RDF.

Example



- Select a subgraphs that corresponds to the declaration;
- Uses pattern-matching.

Example Query

Example

```
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-
syntax-ns#>
SELECT ?s
WHERE {
    ?s ?p ?o .
} LIMIT 100
```

- Preamble
- Action (SELECT, DESCRIBE, ASK, CONSTRUCT)
- Source
- Pattern
- Solution modifier (LIMIT, ORDER BY, OFFSET)

Query Pattern

- Basic Graph Pattern
- Group Graph Pattern
- Optional Graph Pattern
- Union Graph Pattern
- *Graph Graph Pattern*
- Constraint (FILTER BY)

Is not over!

SPARQL is a protocol! It can query remote KGs, generate new entities, perform data migration and so on.

Cheatsheet (BASIC) → `https://shorturl.at/ejouR`

Full documentation → `https://www.w3.org/TR/sparql11-query/`

Part 6.3
KG Distribution

Distributing KGs

- Sharing Resources
- Metadata definition

KGE Projects

- An iTelos project, driven by a specific purpose, is a cooperative, composite project comprised of:
  - A *community* of *researchers* and *participants* including:
    1. producers, who are interested in *sharing* the different resources generated by a iTelos project for potential reuse.
    2. consumers, who are interested in *reusing* the different resources generated by an already existing iTelos project.
    3. intermediaries, who *generate* purpose-specific reusable resources to reduce language, knowledge and data heterogeneity between producers and consumers.
  - and resources ...

DataScientia Community (Click Here!)

Tweets from @DatascientiaF

DataScienti... @Da... · Oct 17

Announcing the return of the @LogmlSchool Summer School, taking place on July 8-12 Applications for mentors to guide groups of ~5 graduate students on a week-long project of the mentor's choosing to be submitted before 17/11/2023 here:

docs.go... LOGML 2024 ...

DataScienti... @Da... · Sep 28

DataScientia Community

Who are the members of our community?

Anybody can be a member of the DataScientia Community. You do not need any specific skills to become a member. The only requirement is that you must be willing to learn about data, AI, and their impact on society. You can decide the amount of effort devoted as a function of how much and how fast you want to learn.

How can you contribute?

There are at least four different types of contributions. You can choose the one or more which best fit your competence, skills, and interests.

- The first type of contribution is to support the collection of person-centric data, as needed in one of the many experiments that we set up.
- The second type, if you have programming skills, is to participate to the open data and open-source community, in charge of development of the AI tools and software which allow DataScientia and its partners to operate.

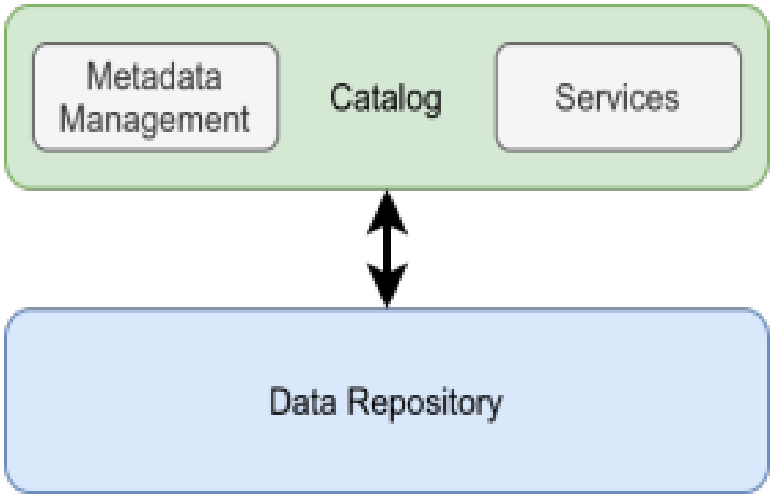
What is a citizen science community?

A Citizen Science Community consists of people who voluntarily help conduct scientific research. Citizen scientists may be involved in various aspects of the research process, e.g., design experiments, collect data, analyze results, and solve problems.

Project Resources

- The various resources generated by a iTelos project are stratified into:
  - *language resources*, e.g., domain language annotation spreadsheet.
  - *knowledge resources*, e.g., teleologies, teleontologies.
  - *data resources*, e.g., datasets, KGs.

Sharing Resources - Catalog

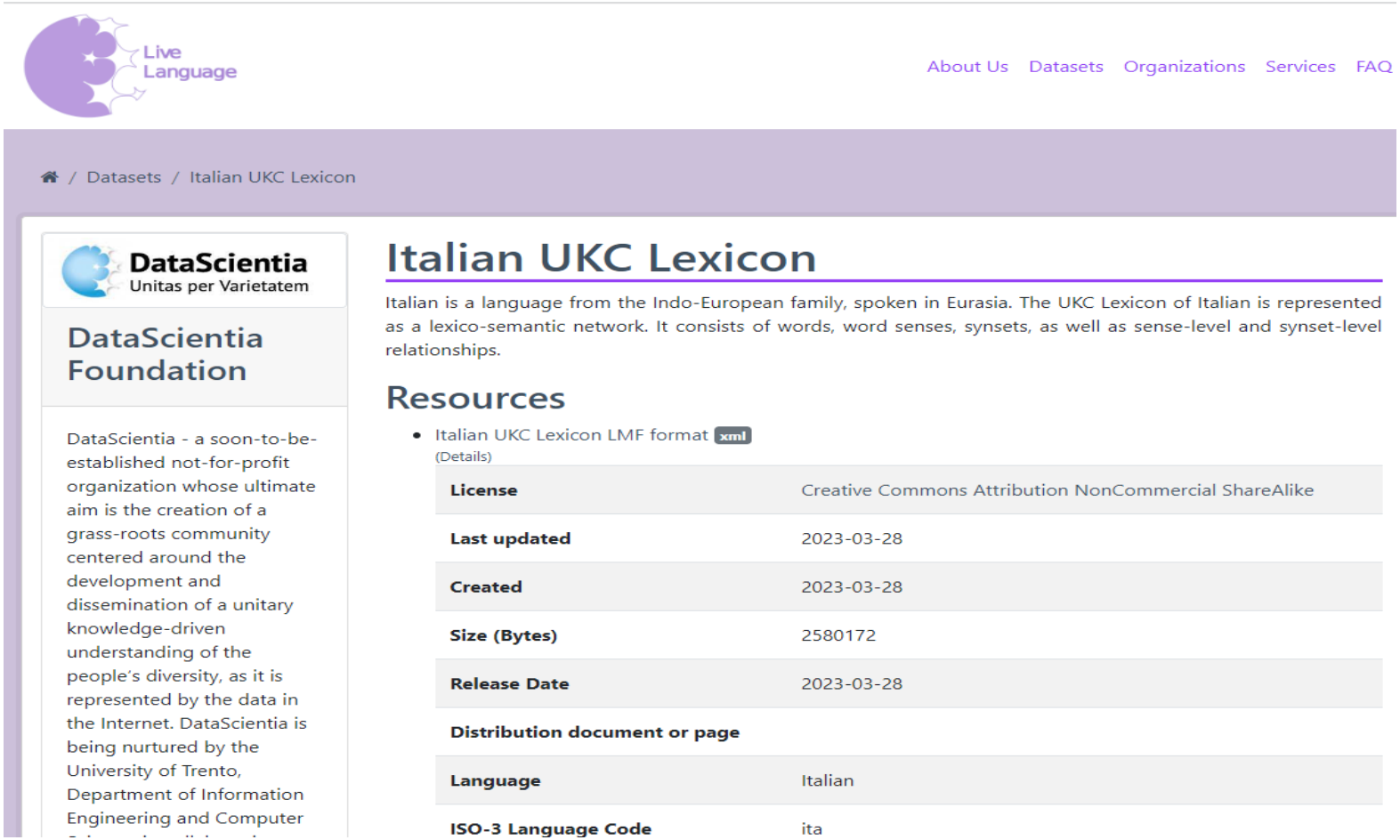


- A web-base unique access point for data repositories
- Smart search and easier navigation for datasets
- Catalog only host dataset metadata

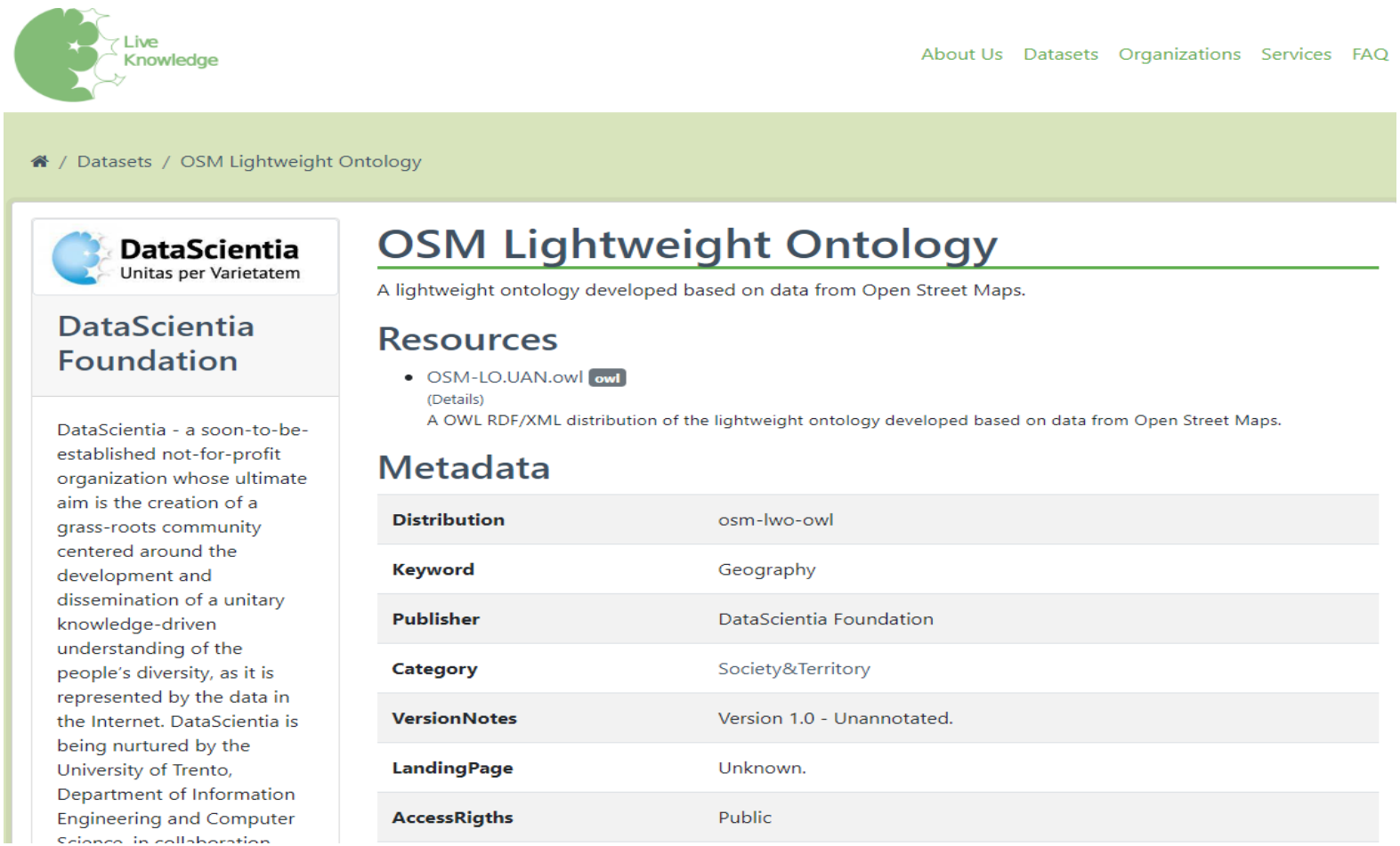
Data Intermediary Catalogs

- The iTelos resources are organized and indexed in their respective (data) catalogs, namely:
  1. LiveLanguage Catalog
  2. LiveKnowledge Catalog
  3. LiveData Catalog
- The KGE Projects are also organized and indexed in a dedicated catalog: LiveData Catalog

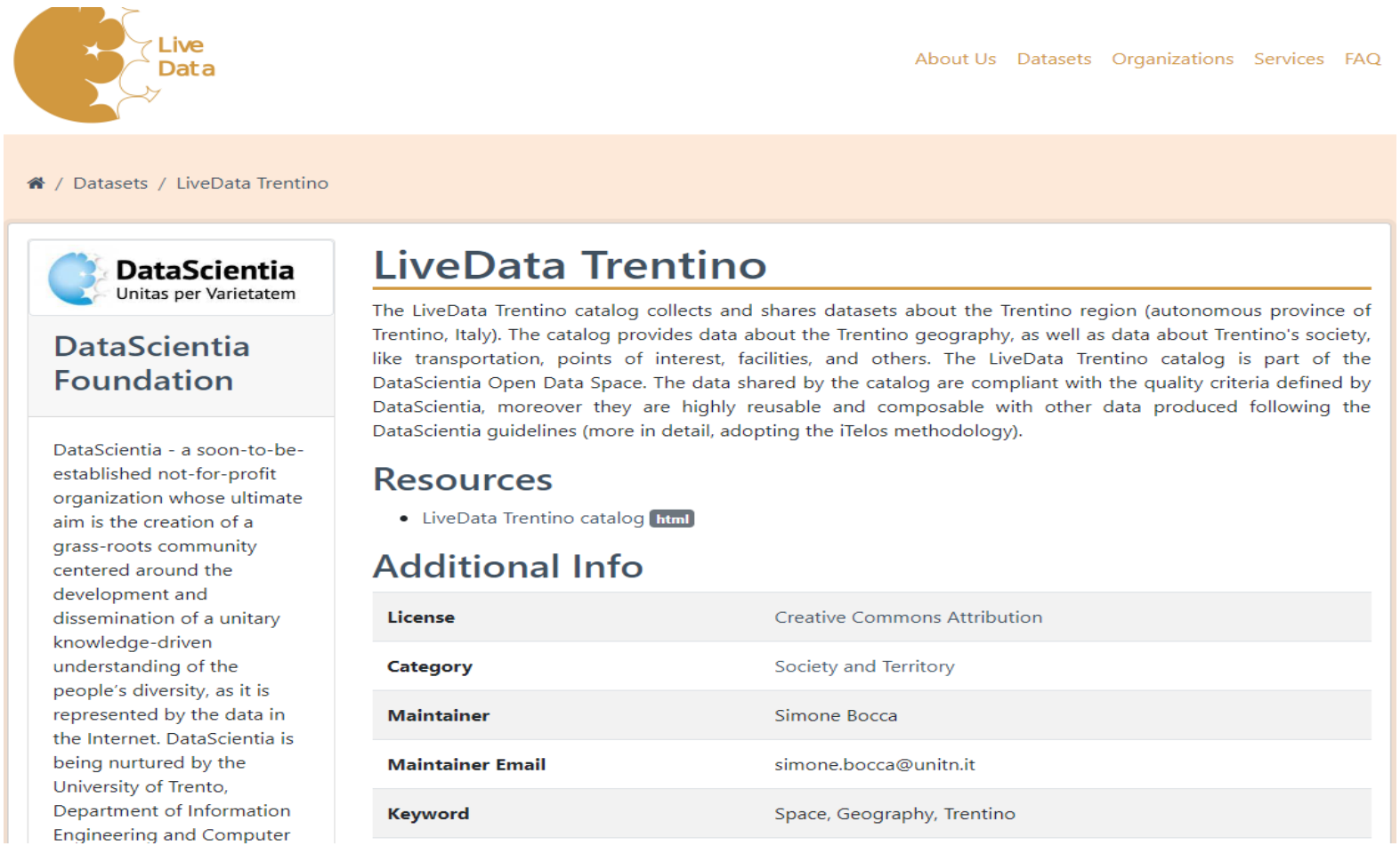
LiveLanguage: Example



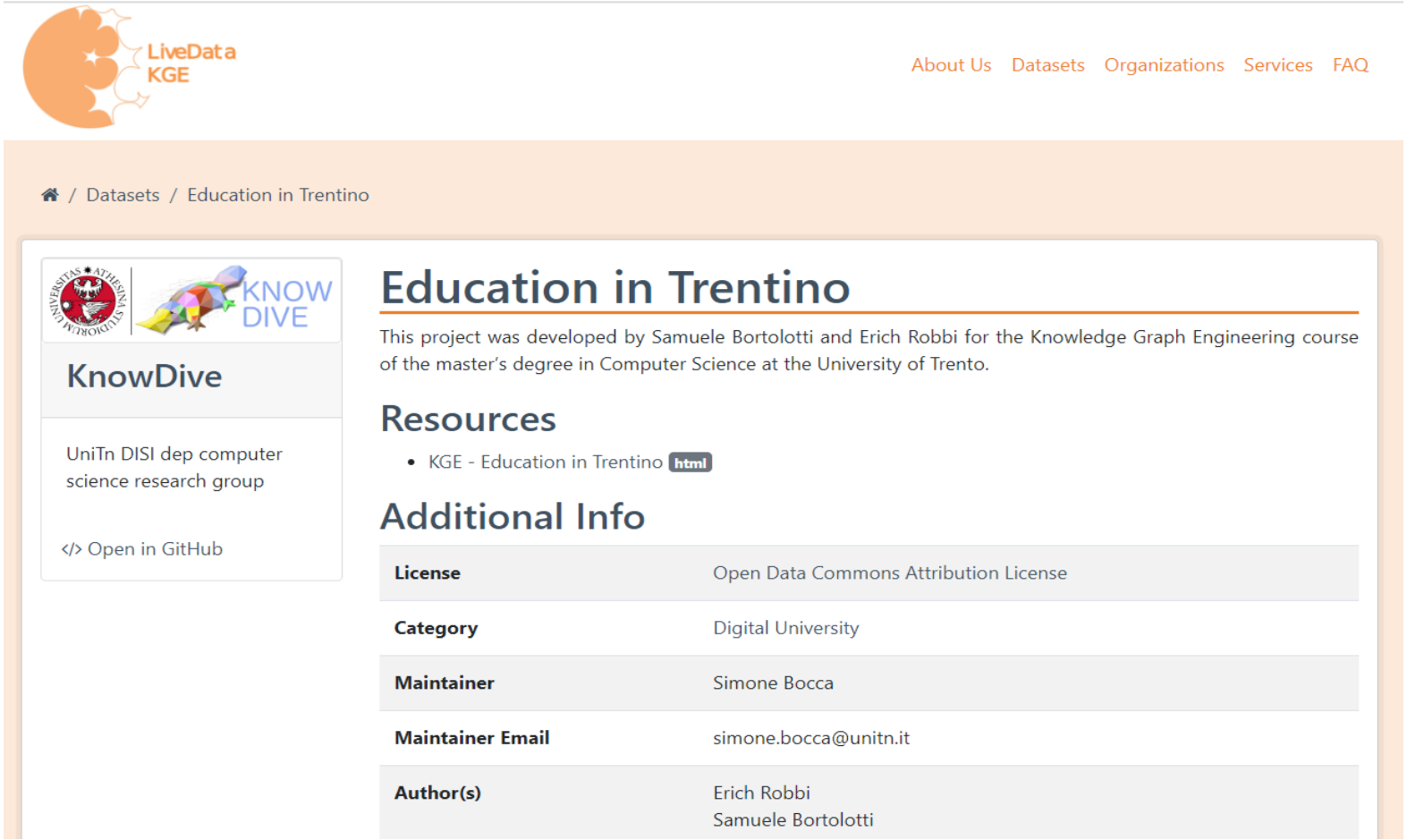
LiveKnowledge: Example



LiveData: Example



KGE Catalog: Example



Distributing KGs

- Sharing Resources
- Metadata definition

Metadata definition

- Metadata is "*structured information that describes, explains, locates or otherwise makes it easier to retrieve, use or manage an information resource*" [NISO, 2017]
- Metadata, in general, has three main purposes:
  1. facilitate description of information resources
  2. facilitate organization of information resources
  3. facilitate discovery of information resources
- In the context of the iTelos Projects, the information resources include language, knowledge and data resources as stated earlier.

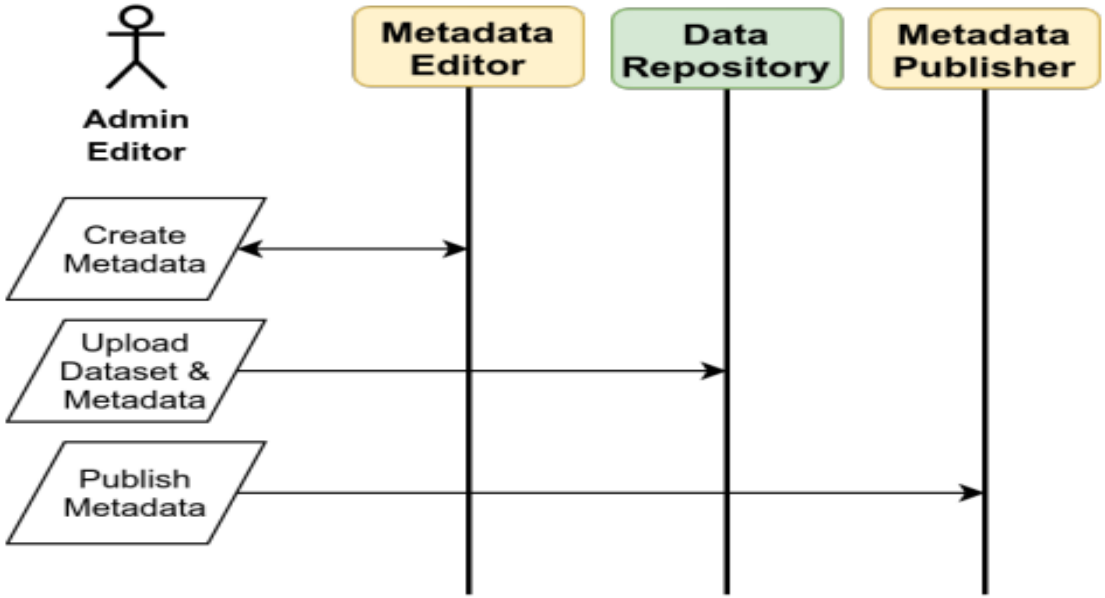
Metadata scope

- In the context of *iTelos*, the scope of metadata is important with respect to two dimensions - *Quality* and *Reusability*
- Firstly, metadata allows the user to determine *data quality and fitness* for their DI project by helping them assess the usefulness of a data resource or a teleology relative to their requirement specification.
- Secondly, *"iTelos assumes the existence of a repertoire of teleologies and provides a rich set of metadata for reusing them"* (Giunchiglia *et al.*, 2021)
- It is essential to always adhere to a *metadata standard* for ensuring *reusability* and *shareability* of data and knowledge resources.

Metadata and Catalogs

Catalog - Services (1/2)

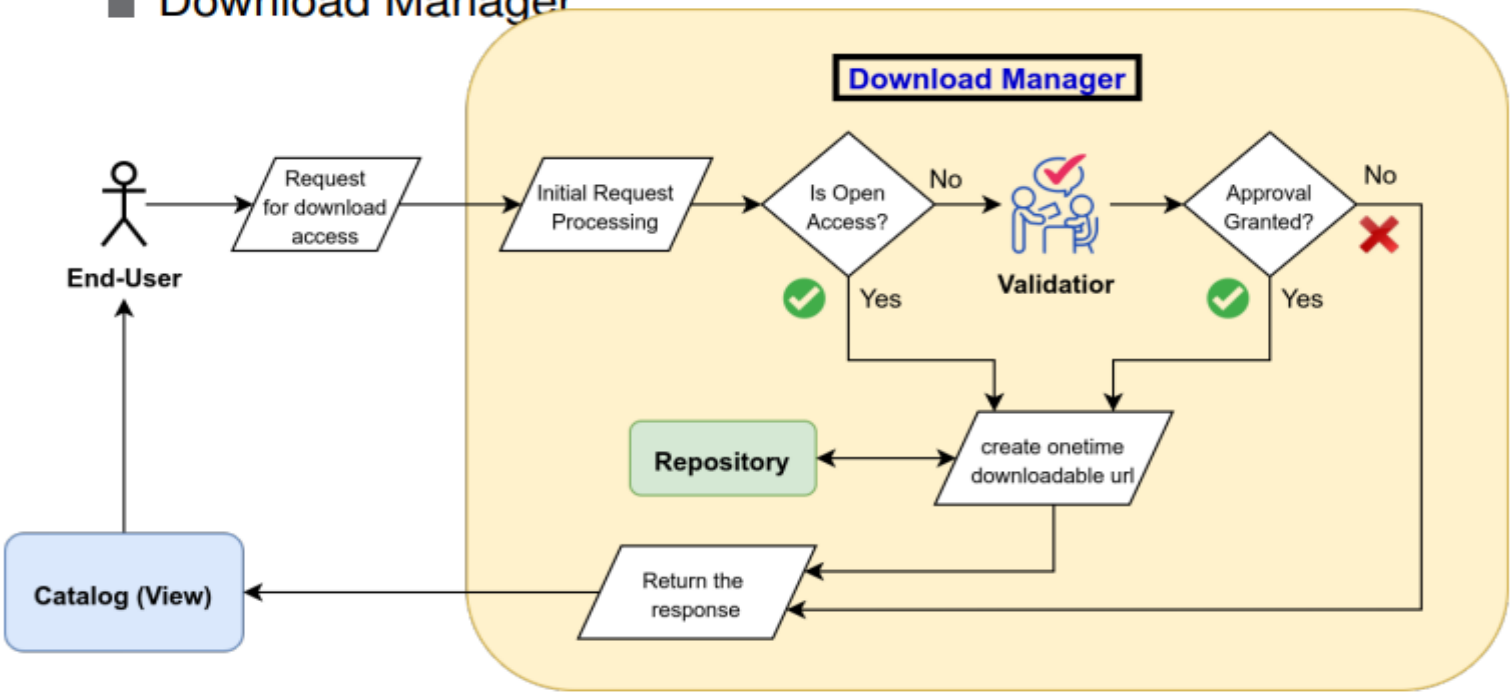
- Metadata Editor
- Metadata Publisher



Metadata and Catalogs (Contd.)

Catalog - Services (2/2)

- Download Manager



iTelos Metadata Organization

- The iTelos Metadata schema is organized into the following layers:
  1. **People Metadata**: metadata attributes about individuals involved in a iTelos project.
  2. **Project Metadata**: metadata attributes about a iTelos project.
  3. **Dataset Metadata**: metadata attributes about dataset resources (whether language, knowledge or data resources) involved in a iTelos project.
- **"NR"** for a metadata attribute means it is non-repeatable, Information for that attribute should be recorded only once.
- iTelos metadata is recorded via **spreadsheets** where column headings are the individual metadata attributes (see next slides).

People Metadata

(a) ds:comIdentifier: this attribute encodes the DataScientia community identifier uniquely identifying a person within the DataScientia ecosystem. ***NR***

(b) ds:firstName: this attribute encodes the first name of the person in a natural language.

(c) ds:lastName: this attribute encodes the last name of the person in a natural language.

(d) ds:email: this attribute encodes the email id of the person.

(e) ds:nationality: this attribute encodes the nationality of the person.

(f) ds:gender: this attribute encodes the gender of the person. ***NR***

(g) ds:affiliation: this attribute encodes the organization to which the person is affiliated in a natural language.

(h) ds:personalWebpage: this attribute encodes the URL of the personal webpage of the person. ***NR***

Project Metadata

1. ds:prjTitle: this attribute encodes the name of the DataScientia project in a natural language as a string.
2. ds:prjURL: this attribute encodes the dereferenciable URL of the DataScientia project. ***NR***
3. ds:prjKeywords: this attribute encodes the various keywords in a natural language that can be utilized to quickly understand the theme of the project.
4. ds:prjType: this attribute encodes the type of the DataScientia project. e.g., Knowledge Resource Generation, Knowledge Resource Annotation, etc. ***NR***
5. ds:prjDescription: this attribute can be used to provide a description of the DataScientia project in a natural language.
6. ds:prjStartDate: this attribute encodes the date of the commencement of a DataScientia project. ***NR***

Project Metadata (Contd.)

7. ds:prjEndDate: this attribute encodes the date of conclusion of a DataScientia project. ***NR***
8. ds:prjFundingAgency: this attribute encodes the name of the agency or institution funding a DataScientia project.
9. ds:prjInput: this (repeatable) attribute encodes the various inputs (e.g., datasets, specifications, etc.) with respect to a DataScientia project.
10. ds:prjOutput: this (repeatable) attribute encodes the various outputs (e.g., datasets, domain languages, etc.) with respect to a DataScientia project.
11. ds:prjCoordinator: this attribute encodes the name of the research coordinator in charge of a DataScientia project. ***NR***
12. ds:prjObservations: this attribute can be used to record any observations about a DataScientia project in a natural language.

Datasets Metadata

1. ds:DatLicense: this attribute encodes the license of the dataset, e.g., CC-BY-SA-4.0 ***NR***
2. ds:DatURL: this attribute encodes the dereferenceable URL of the dataset. ***NR***
3. ds:DatKeyword: this attribute encodes the keywords which can quickly convey the topic of the dataset.
4. ds:DatPublisher: this attribute encodes the publisher of the dataset.
5. ds:DatCreator: this attribute encodes the creator of the dataset.
6. ds:DatOwner: this attribute encodes the owner of the dataset.
7. ds:DatLanguage: this attribute encodes the natural language(s) in which the dataset information is represented.
8. ds:DatLevel: this attribute encodes the knowledge level of the dataset, e.g., L1-2, L4. ***NR***

Datasets Metadata (Contd.)

9. ds:DatSize: this attribute encodes the byte size of the dataset. ***NR***
10. ds:DatName: this attribute encodes the name of the dataset in a natural language.
11. ds:DatPublicationTimestamp: this attribute encodes the timestamp of the publication of the dataset in the respective catalog. ***NR***
12. ds:DatDescription: this attribute encodes the description about the dataset in a natural language.
13. ds:DatVersion: this attribute encodes the version of the dataset. ***NR***
14. ds:DatDomain: this attribute encodes the domain to which the dataset belongs, e.g., society and territory.
15. ds:DatFileFormat: this attribute encodes the file format of the dataset, e.g., OWL RDF/XML, Excel. ***NR***

Projects Metadata Sheet - Example

	A	B	C	D	E	F	G
1	**Title**	**URL**	**Project Type**	**Description**	**Start Date**	**End Date**	**Funding Agency**
2							
3	OSM General Lightweight Ontology General	To Be Added.	Knowledge Resource Genera	This project focused on the g	11-03-2023	11-04-2023	DataScientia Foundation
4	OSM General Lightweight Ontology Annotat	To Be Added.	Knowledge Resource Annota	This project focused on the a	07-06-2023	10-06-2023	DataScientia Foundation
5							
6							
7							

Dataset Metadata Sheet - Example

2	Description	Title	Distribution	Keyword	Publisher	Theme	Version Notes	Landing Page	Access Rights	Creator
3	The DBpedia on	The DBpedia Or	dbpedia-owl	Wikipedia, Multil	DBpedia Organi	Upper Level	new version 4.2-	https://dbpedia.c	Public	DBpedia Organi
4	The Bibliographi	The Bibliographi	bibo-owl	Biblioigraphy		Upper Level	As of today, the	http://purl.org/on	Public	Bruce D'Arcus, F
5	FOAF is a projec	Friend Of A Frie	foaf-owl	People	Dan Brickley	Upper Level	No updates sinc	http://www.foaf-p	Public	Libbly Miller, Dar
6	This is the encoc	CIDOC Concept	crm-owl	Customer managment		Upper Level	2016: Annual rev	https://cidoc-crm	Public	FORTH-ICS
7	CAT is an RDF v	Data Catalog Vo	dcat.owl	Catalogs	W3C Data Excha	Upper Level	(2020-11-01) Gh	http://www.w3.or	Public	Richard Cygan
8	The Geonames	The Geonames	geonames-owl	geography	Geonames.org	Upper Level	2020: Annual rev	http://www.geon	Public	Bernart Vatant
9	A Geographic	OGC GeoSPAR	GeoSPARQL-ow	Geometry	Open Geospatia	Upper Level	2016: Annual rev	http://www.open	Public	Open GeoSpatia
10	A vocabulary fo	NeoGeo Spatial	geovocab-owl	Topology		Upper Level	2016: Annual rev	http://geovocab.	Public	Juan Martin
11	A specializatio	OGC Geometry	gml-owl	Geometry		Upper Level	2016: Annual rev	http://www.open	Public	Open GeoSpatia
12	This ontology i	General Transit	gtfs-owl	Travel		Upper Level	2016: Fixed vers	https://raw.githut	Public	Pieter Colpaert,
13	A vocabulary to	The Opening Ho	ical-owl	Time	W3C	Upper Level	2019: Annual rev	https://github.co	Public	Pieter Colpaert
14	A specification	Simplified Featu	sf-owl	Geometry		Upper Level	2016: Fixed vers	http://www.open	Public	Open Geospat
15	The Simple Kn	Simple Knowled	skos-owl	Concept scheme	W3C	Upper Level	2015: Annual rev	http://www.w3.or	Public	Alistair Miles, S
16	This ontology i	Sensor, Observa	sosa-owl	IoT, Environmen	W3C/OGC Spati	Upper Level		http://www.w3.or	Public	W3C/OGC Spati
17	This ontology c	Semantic Senso	ssno-owl	IoT	W3C	Upper Level		http://www.w3.or	Public	W3C/OGC Spati