PABAKE: Linked Open Seas

Knowledge Engineering

Federico Battistella, Alessandro Pavesi

February 2022

Alma Mater Studiorum - Università di Bologna

Second Cycle Degree in Artificial Intelligence

1 Introduction

In this project we applied to model a dataset containing observation of sea parameters registered by sensors spread all over the Italian seas. Our approach has the purpose of building the best suited ontology and relative to represent the data making them accessible by whoever wants to retrieve information about the state of the seas. The use of ontology design pattern increases the readability, the extensibility and the similarity with the standard ontologies developed for similar tasks.

The definitive version of the ontology is published with the IRI:

$$https://w3id.org/stlab/ke/pabake\#$$

1.1 Tasks

We have been assigned ten tasks aiming to build the best ontology representation of the domain assigned. These tasks includes lot of works on the data, additionally it's asked to configure tools to better understand the philosophical approach to build the ontology and moreover to integrate the ontology and the data derived into tools for data visualization to be published.

The tasks are:

1. Analysis of existing datasets using heterogeneous formats, to produce RDF knowledge graphs
2. Application of the eXtreme Design methodology (competency questions, ODP reuse, testing, etc.) to develop OWL ontologies for the knowledge graphs
3. Definition of mapping rules for transforming input data into semantic web knowledge graphs, according to the developed ontologies
4. Generation of URIs and publication of ontologies and knowledge graphs (with permanent URIs)
5. Application/use/configuration of tools for entity linking
6. Application/use/configuration of tools for ontology alignment
7. Publication of a SPARQL endpoint
8. Integration of LODView for knowledge graph browsing
9. Integration of LODE for producing human-readable documentation of the ontologies
10. Creation of a docker that will contain every thing.

The goal of this project is to analyze an input data source made up of CSV files containing sensors and observation data. Sensors of various sorts (such as wave gauges, thermometers, and barometers) are part of two connected networks of buoys and tide gauges that monitor the environmental and meteorological conditions of Italian waters. The aim is to decipher the data source's schema in order to model and build a knowledge graph. A modular ontology network for describing subject knowledge at various levels of granularity and flavors; and connected open dataset must be included in the knowledge graph.

The consinstency of the knowledge graph will be checked with SSN/SOSA ontologies, and evaluated as a reorganization of the Italian Institute for Environmental Protection and Research's current KG (ISPRA).

1.2 Data

The original data sources provided to us include three dataset from the Linked Data by the Italian Institute for Environmental Protection and Research (ISPRA):

- RON Rete Ondametrica Nazionale
- RMN Rete Mareografica Nazionale
- RON and RMN observations

Given the huge amount of data present in the dataset, including values from different type of observation and different features to observe (like conductivity, or temperature or ph), in conclusion, according to professor Nuzzolese, we decided to represent the *wave* data.

The data have been provided as csv files, the main files are:

- *observations.csv*: it provides the list of files to read, with information about the time and the place where the data are taken
- *observations_schemata.csv*: defines the observed parameters with unit of measure and description
- *sensors.csv*: includes the basic information about the sensors
- *station.csv*: includes the basic information about the stations
- *csv/wave folder*: contains all the csv files with the observation values, the main data are in here.

During the analysis of the data we found out that only a small preprocessing was needed. The small part of the data to be cleaned needs only some adjustment on how they are written, so we chose to apply a function to lower the capitalized characters. Moreover we checked the dataset for error values or incomplete data, only few values were missing so this brought us to the decision to set those value at 0.0, which is a safe assignment as they can never assume that value.

2 Discussion

To elaborate the project we followed the *eXtreme Design methodology* that is a framework for pattern based ontology design.

2.1 Concepts

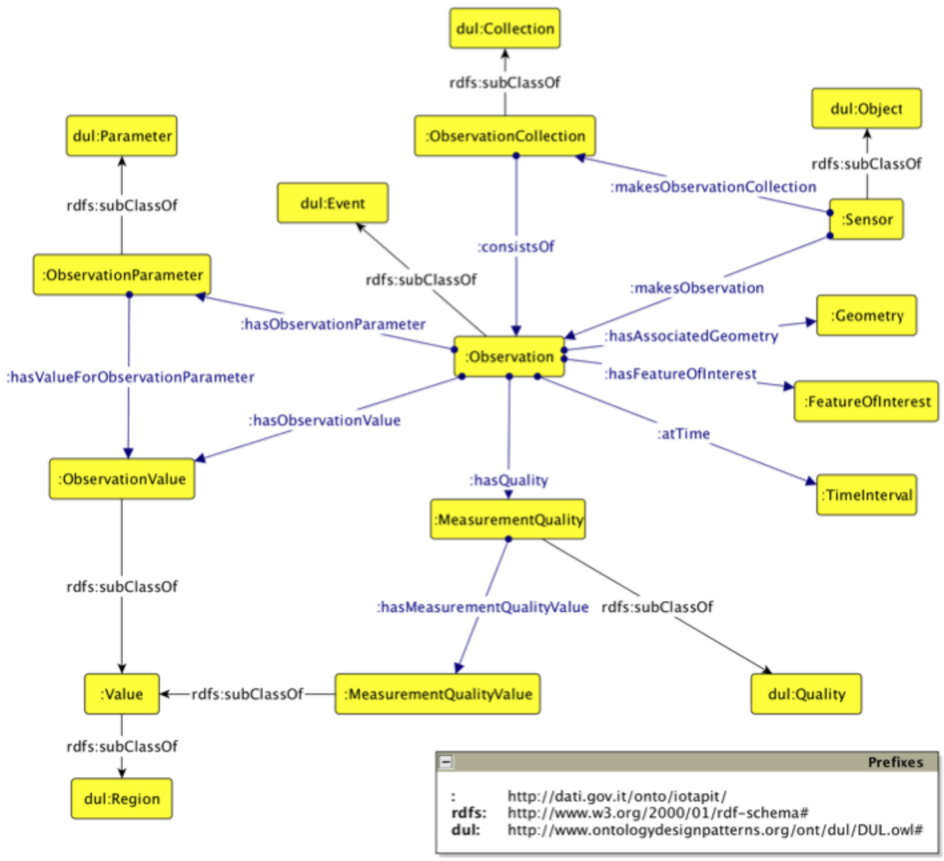
The main concepts are the observations from which the final users will extract the information about the sea. The *observations* include a lot of data provided by *sensors*, and also are connected to the *feature of interest* regarding them, to the *coordinates*, the *city* and more other parameters. The huge amount of data and connections between each data allow us to use different patterns, but also conditioned our approach during the modelling phase.

The main concepts defined are:

- **Provider**: The provider of the station
- **Station**: A station represents a physical object that can host Sensors.
- **Sensor**: It represents the physical object that makes the observation
- **ObservationCollection**: It represents a set of observations made at the same time and by the same station, each of them regarding different parameters
- **Observation**: It is a concept that represents a specific observation in a defined state
- **ObservationValue**: The value of the sensor reading
- **ObservationParameter**: The parameter observed, it can be the direction of the wave, its height or its direction
- **City**: The nearest city to a station or a sea
- **Sea**: The sea in which the station is

2.2 Reference Ontology

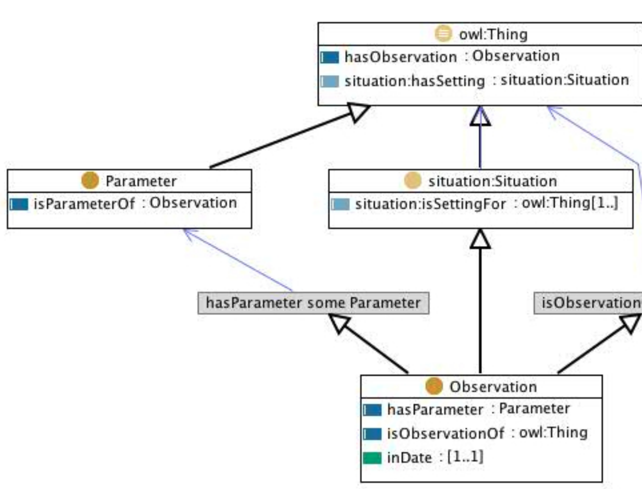
The ontology we took as reference is *A pattern-based ontology for the Internet of Things* [**Gangemi2017APO**]. This ontology was really suited for the representation of the given domain, so we took inspiration for the root design and then we added concepts specific for our problem. As it can be seen from the following picture, there some concepts common to both the ontologies, for example: observation with linkings to Parameter, ObservationValue, ObservationCollection, Sensor, Geometry... The ontology considered is very suited as baseline for the Linked Open Sea representation, but then we needed to specialize our ontology on the sea observations domain.



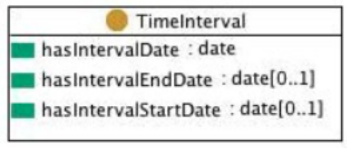
2.3 Ontology Design Pattern

Ontology Design Patterns (ODPs) are a middle ground approach to ontology development. They may be thought of as a stripped-down form of design principles similar to those found in basic ontologies, but with less 'clutter.' They can be smartly modularized basic ontology pieces that serve as design snippets for effective modeling methods, in other words. They may also be considered as a method of bottom-up pattern discovery that is then repeated across the ontologies and made available to others as a 'best practice' design solution for some modeling component.

Observation The Observation design pattern is the core of our ontology, the Observation concept perfectly suits with our needs, moreover we also adopted the Parameter concept. It is used to model a single Observation, the parameters and the properties between the Observation and the Sensor that made it.



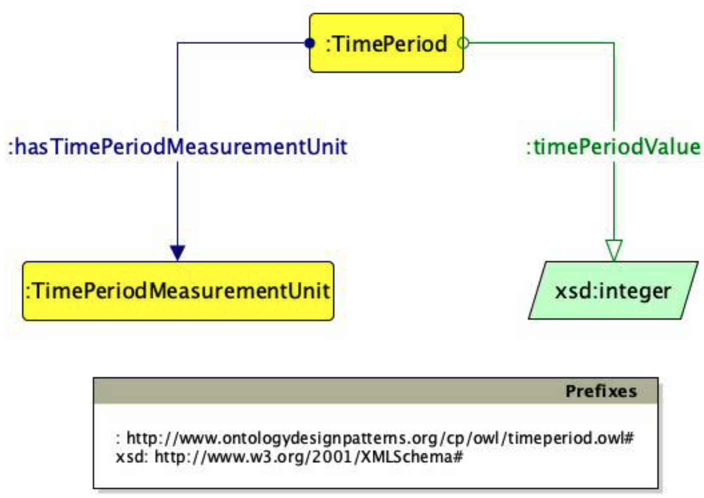
TimeInterval



It is used to model an interval of time in which something happens, in our case we use this pattern as the super class of another time related concept: *TimeInstant*.

We chose to use a subclass of TimeInterval to model our case in which the observation is made in a single instant of time, without a duration because it is only a read of sensors. We considered the TimeInstant as a subclass of TimeInterval defining it as a TimeInterval with duration equals to zero.

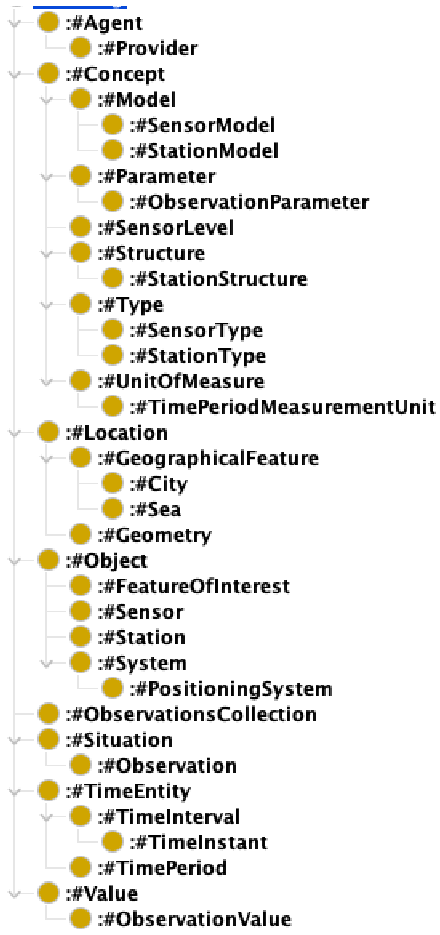
TimePeriod In the same category of patterns we decided to adopt also the *TimePeriod* pattern:



It differentiates from the previous one because in this case the period of time is not defined by a start and and end datetime,but it is described as an amount of time not localized in timeline. For example "one month" is a measure of time with no start/end defined.

2.4 Ontology

The ontology includes all the concepts we described before, plus some more defined to give to entirely cover the deep, specific structure of the data.



We set a lot of axioms to define the sub class properties, to restrict the object properties among classes and to define data properties type.

2.5 Competency Questions

We thought competency questions to better find potential Ontology Design Patterns (ODPs) for a powerful and expressive developement of our ontology, following the principles of eXtreme Design methodology. The competency questions are:

1. How many stations are deployed in the X seas?/deployed in the X city?
2. How many/What seas are monitored?
3. How many/What models of station are actually deployed?
4. How many/What materials are used for build the stations?
5. How many/What places are monitored?
6. What is the value of “parameter” for the sea X in a certain period Z? for the city Y? for different parameters
7. What is the parameter X for the city Y at time Z? And in the interval of time from Z1 until Z2?
8. What are the observable properties for the city X?
9. What are the sensors in the city X? for the sea X?
10. How many stations are part of the network X?

11. How many sensors are hosted on the station X?

The ontology contains a huge amount of data, we just stated some of the main questions that can be formulated, but there are another large number of possible questions.

SPARQL We will show a small part of the competency questions that can be defined, using the SPARQL program:

The query to extract the values of a specific parameter (Hm0) for a defined city (Alghero, ISTAT code 90003):

```
1)
Select ?value
where {
    ?obs pabake:hasParameter <https://w3id.org/stlab/ke/pabake/ld/observation_parameter/
        pabake:hasObservationValue ?value;
        pabake:isMadeBy ?sensor.
    ?sensor pabake:hasStation ?station.
    ?station pabake:isInProximityOf <https://w3id.org/stlab/ke/pabake/ld/city/090003> .
}
```

It can be modified to extract all the observed parameters or to extract data for different cities.

Here the query to extract the material with which the stations are made:

```
2)
Select DISTINCT(?material)
where {
    ?station pabake:hasStationModel ?stationModel.
    ?stationModel pabake:hasStationStructure ?material
}
```

Editing the query with the desired parameter, we can extract the desired station informations as: the positioning system, the diameter, the manufacturer and other parameters.

The query to extract the cities contained in the data:

```
3)
Select DISTINCT(?city)
where {
    ?station pabake:isInCity ?city.
}
```

2.6 Mapping Rules

To load the instances of the ontology we had to use a framework for data mapping called *PyRML*. This framework provides a natural way of mapping data from csv files into triples in RDF type. PyRML is a python package that allows to apply a template to the data, so it's easy to transform data into knowledge graphs. The template we made are:

- *observation.ttl*: the observation template is used to create the ObservationCollection classes and the other class TimeInstant
- *observationsDir.ttl*: this template model the subclass observation, here with the information about the *Direction of the wave*
- *observationsTp.ttl*: here the template model the information about the *Spectral peak wave period*
- *observationsTm.ttl*: here the template is used for the information about the *Average wave period*
- *observationsHm0.ttl*: used for the information about the *Significant wave height*
- *project.ttl*: contains all the other classes, including the stations, the sensors, the city or the sea

2.7 Ontology Alignment

The ontology alignment is the process consisting in the determination of correspondences between concepts in ontologies.The process has been performed on three fondational ontologies: Dolce, Ssn, Sosa.

The first ontology Dolce is a foundational conceptual ontology, due to its very general applicability we have found a lot of correspondencies between Pabake and it, however we can state that most of the Dolce concepts can generalize Pabake's, which is very specific for its domain. There are some concepts that seems to coincide perfectly, for example "Time Interval" express the same concept in both the ontologies, on the other hand, the very generalizing concept "Endurant" can only enclose some of the most general concepts of the Pabake ontology as "Object". We can then sum up that DOLCE ontology can nearly enclose the whole Pabake ontology in itself because of the flexibility and non-deep specific domain design of it. Although there are some concepts that remain difficult to express or to enclose in Dolce, as for example, the "Observation Collection". The Sosa ontology seems to be the ontology with the most correspondences with Pabake, the domain of both is very similar and quite specific so we can find a lot of equivalent concepts between the two ontologies.

There are a lot of concepts that we can consider equivalent:

- ObservableProperty - ObservationParameter
- time:instant - TimeInstant
- Observation - Observation

- time:interval - TimeInterval
- Sensor - Sensor
- QuantityValue - Value

These are some of the most representative equivalent classes in the alignment of the two ontologies, we also found few object properties common to both the ontologies which have the same meaning and the same function, and in most cases, also the same name. One example is the object property "makesObservation" from Pabake which is equal to "detects" in Sosa. The Sosa ontology is very suited for the representation of the datas considered in this paper, due to the similarity of the contexts represented in the ontologies we managed to find a lot of equaivalences, and much less subclasses than Dolce. Although not all the concepts are common to both the ontologies, for example Sosa contains the "Actuator" concept which has not been represented in Pabake because it wasn't present in the data.

2.8 Entity Linking

To check the correspondences of concept and classes of our knowledge graph and other external knowledge graphs like DOLCE or SSN/SOSA we used LIMES by [al., 2021]. LIMES is a link discovery framework for the Web of Data.

The program exports a file in which the concepts similar, due to Jaccard index, between two knowledge graphs are listed.We maintained the concepts that have at least a 0.5 Jaccard Index.

DOLCE is a conceptual ontology so there are few correspondences with our ontology, but still the use of patterns helps us to connect DOLCE with PABAKE:

DOLCE	PABAKE
time-interval	TimeInterval
feature	FeatureOfInterest
feature	GeographicalFeature

SSN/SOSA that is very similar to PABAKE, in fact the similarity includes almost all the concept of both the ontologies:

DOLCE	PABAKE
Interval	TimeInterval
Instant	TimeInstant
FeatureOfInterest	FeatureOfInterest
ObservableProperty	ObservableParameter

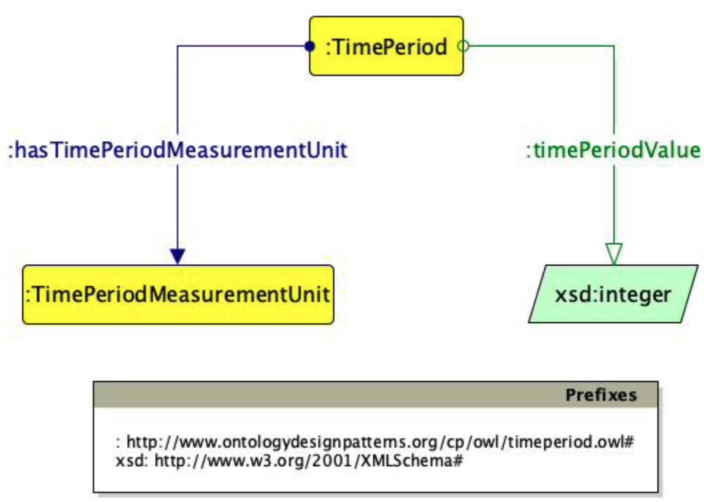
2.9 Docker

Finally the entire project is delivered as a Docker container containing the integration of our data with *LODE* and *LODView* for knowledge graph browsing and human-readable documentation of the ontology production. The *Docker* contains also the a SPARQL endpoint used by other services and useful to query the knowledge graphs, it is included on the *Virtuoso* service.

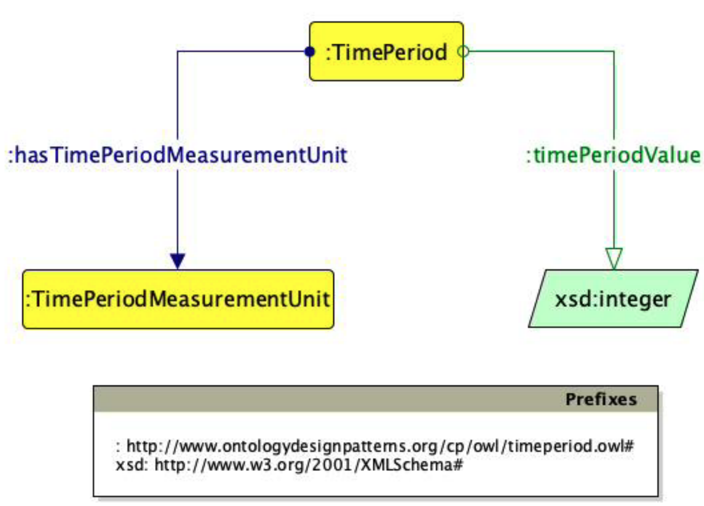
The exposed port for each service is:

- *Virtuoso*: http://localhost:8891/sparql
- *LODE*: http://localhost:9090/lode
- *LODView*: http://localhost:8181/lodview

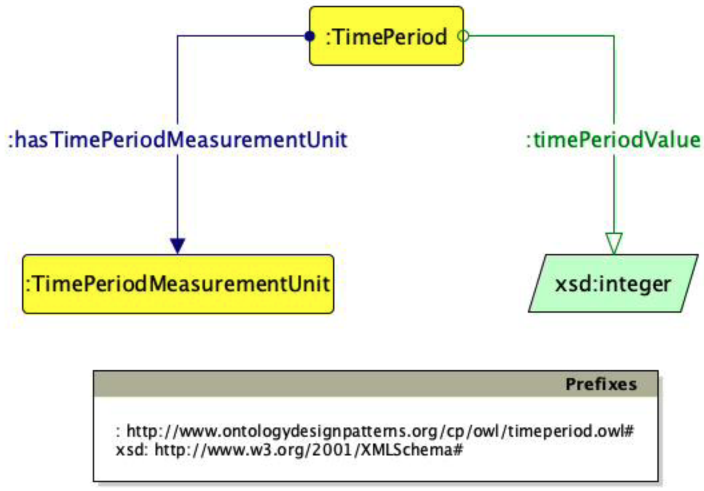
The services needs to be called with the right procedure to fully works, for example the Virtuoso service shows a page in which it is possible to write and execute queries.



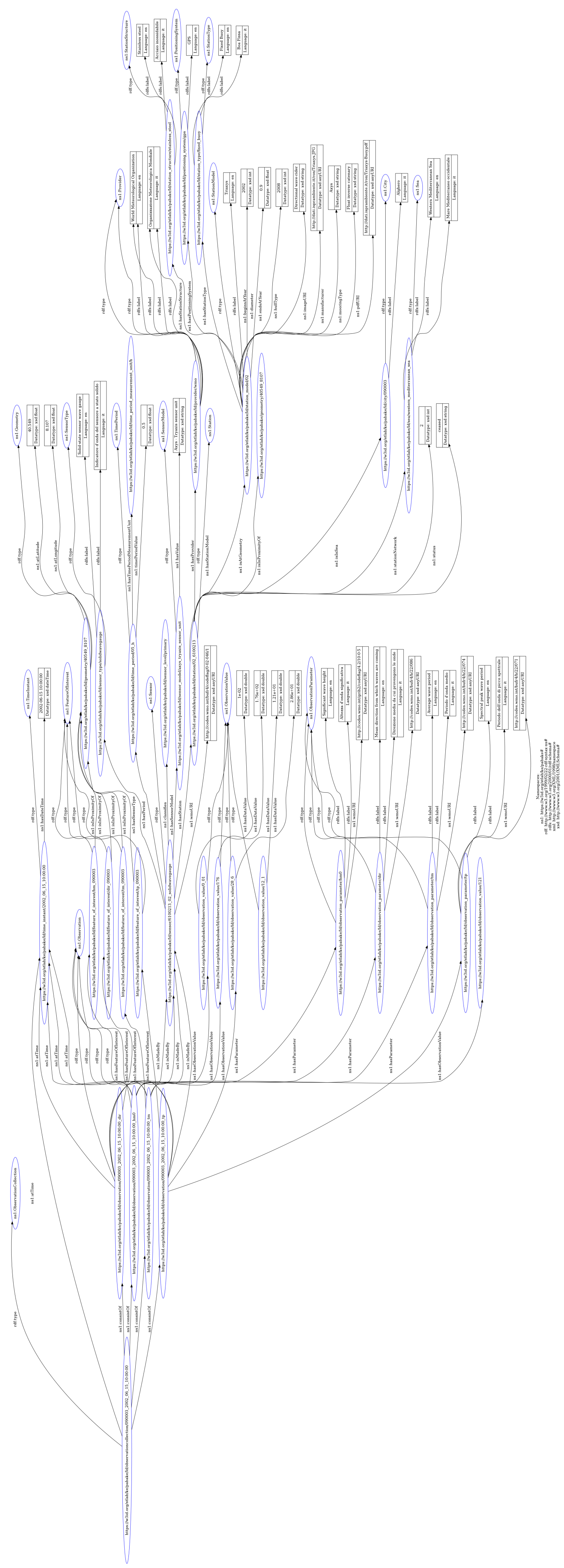
LODE provides a visualization service of the knowledge graphs, so the instances of our ontologies:



LODE provides a visualization service of the ontology, so it needs to be called with the URL of the ontology:


:TimePeriod
:hasTimePeriodMeasurementUnit
:timePeriodValue
:TimePeriodMeasurementUnit
xsd:integer
Prefixes
: http://www.ontologydesignpatterns.org/cp/owl/timeperiod.owl#
xsd: http://www.w3.org/2001/XMLSchema#

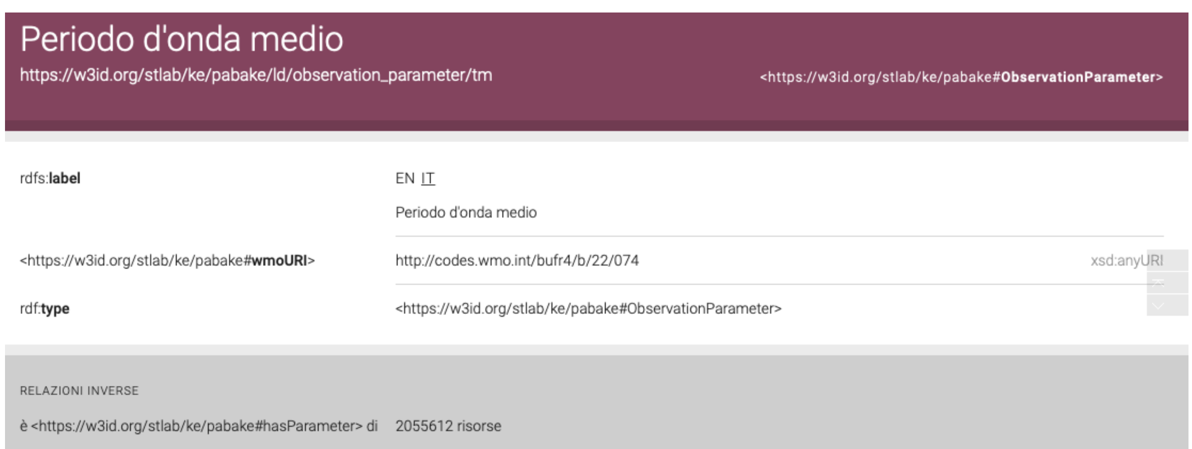
An overview on a small portion of the data in our ontology:



Using the services we can explore the knowledge graph.

For example we can explore the entire Ontology using LODE:

Or we can explore an instance of the ontology using LODView:



3 Conclusion

Our project includes to understand the schema underlying the data source for modelling and constructing a knowledge graph. With the purpose of create a modular ontology to represents the domain knowledge exposed by the linked open dataset provided. Our approach uses different Ontology Design Patterns to exploit the data organization defined by sector experts. The amount of data created is huge, it is possible to explore it with the docker published on GitHub.

To conclude we are satisfied with the organization of the concepts as well as the creation of a project that includes all the relevant analysis, the services implementation and extensions required.

References

A., Gangemi (n.d.). “Ontology Design Patterns for Semantic Web Content”. In: *The Semantic Web – ISWC 2005. ISWC 2005. Lecture Notes in Computer Science, vol 3729. Springer, Berlin, Heidelberg.*

al., AC. Ngonga Ngomo et (2021). “LIMES - A Framework for Link Discovery on the Semantic Web”. In: *KI-Künstliche Intelligenz, German Journal of Artificial Intelligence - Organ des Fachbereichs "Künstliche Intelligenz" der Gesellschaft für Informatik.*

D., Merkel (n.d.). “Docker: lightweight linux containers for consistent development and deployment.” In: *Linux journal. 2014;2014(239):2.*

D. V. Camarda S. Mazzini, A. Antonuccio (2014). “LodView”. In.

E., Presutti V. Daga E. Gangemi A. Blomqvist (2009). “eXtreme Design with Content Ontology Design Patterns.” In.

M.A., Musen (2015). “The Protégé project: A look back and a look forward.” In: *AI Matters. Association of Computing Machinery Specific Interest Group in Artificial Intelligence.*

Peroni S. Shotton D.M., Vitali F. (2012). “Making Ontology Documentation with LODE.” In: *In Proceedings of the I-SEMANTICS 2012 Posters Demonstrations Track, Graz, Austria.*

Z., Auer S. Bizer C. Kobilarov G. Lehmann J Cyganiak R. Ives (2007). “DBpedia: A Nucleus for a Web of Open Data.” In: *The Semantic Web. ISWC 2007, ASWC 2007. Lecture Notes in Computer Science, vol 4825. Springer, Berlin, Heidelberg.*