Orchestrating an interoperable sovereign federated Multi-vector Energy data space built on open standards and ready for Gaia-X

How To Semantify

This document is issued within the frame and for the purpose of the OMEGA-X project. This project has received funding from the European Union's Horizon Europe Framework Programme under Grant Agreement No. 101069287. The opinions expressed and arguments employed herein do not necessarily reflect the official views of the European Commission.

Objectif:

In OMEGA_X, the data provider (DP) ~~will~~ should semantify their data sets that will be exchanged with the service provider (SP) to enhance interoperability between different stakeholders. This process will transform raw datasets such as ~~the~~ csv or json file into a knowledge graph (KG) serialized in turtle (ttl) or json-ld (.json). The knowledge graph will contain meta-data that describe~~s~~ the information exchanged between DP and SP enabling the SP to understand and interpret the data exchanged. This semantisation is done based on OMEGA-X Common Semantic Data Model (CSDM).[1]

Semantisation using the CSDM:

The CSDM is ~~conceived~~ developped by EDF Team with the ~~assistance~~ support of the SPs and DPs. Workshops have been organized for each use case family (UCF) to extract the information needed to be exchanged. The CSDM documentation can be found in http://www.w3id.org/omega-x and the ontology modules can be found in http://www.w3id.org/omega-x/repository.

~~Used Tool for semantisation:~~

Many methods and tools exist to support dataset semantisation (transformation from raw formats (csv, xml,) into RDF graphs). Figure illustrates an overview of possible approaches.

a mis en forme : Anglais (États-Unis)

a mis en forme : Titre 1, Taquets de tabulation : Pas à 4,02 cm

Code de champ modifié

Code de champ modifié

a mis en forme : Français (France)

[1] https://fz-hannou.github.io/Omega-X/

This document is issued within the frame and for the purpose of the OMEGA-X project. This project has received funding from the European Union's Horizon Europe Framework Programme under Grant Agreement No. 101069287. The opinions expressed and arguments employed herein do not necessarily reflect the official views of the European Commission.

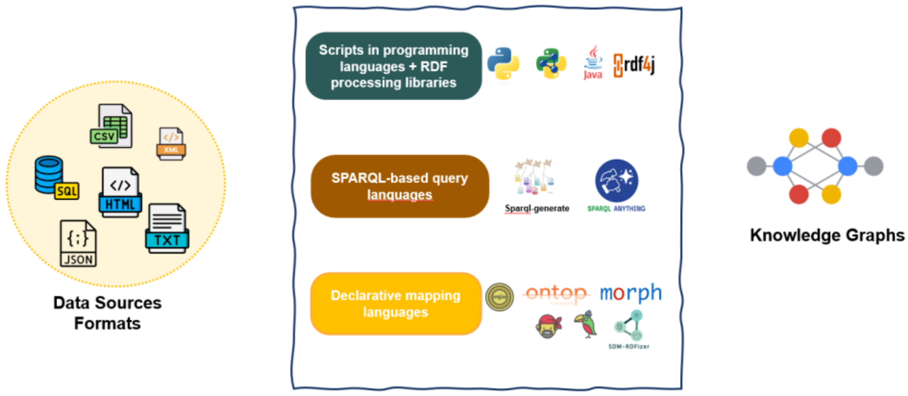

Figure 1Overview of semantisation approaches

In this document, we illustrate semantisation processes conducted in two different pilot sites (Velverde and Maia), using two different approaches: mapping rules and python scripts.

RML Mappings

In this document, we propose to do the semantisation using RML mapping rules that maps the csv file into a turtle KG.
The used tool is SDM-RDFIZER https://github.com/SDM-TIB/SDM-RDFizer. python3 -m pip install rdfizer.

To illustrate how the semantisation done using this tool, we will use a running example of a CSV file from Evora pilte site.
The name of the file is ***combined_production_consumption_574.csv,*** where 574 identify each home management system (HMS) where captured. The structure of the csv file is a s follow:

a mis en forme : Paragraphes solidaires

a mis en forme : Légende, Taquets de tabulation : Pas à 4,02 cm

a mis en forme : Anglais (États-Unis)

a mis en forme : Anglais (États-Unis)

a mis en forme : Titre 2, Taquets de tabulation : Pas à 4,02 cm

Commenté [HFZ1]: Peut être un rappel rapide de RML ? Ou just ele lien à la spec pour voir la syntaxe

Commenté [HFZ2]: Rappeler le UCF

DateUTC	TotalActiveEnergy+	TotalActiveEnergy-	production	consumption
2021-12-31 00:00:00	0.008	0	0	0.008
2021-12-31 00:05:00	0.008	0	0	0.008

The KG of this file will be:

- A general Timeseries that has as evaluation point the HMS, a peridicity of 15 minutes, and a data provider EDP.
- Each column will be a timeseries that has as property the column name, a unit, and data points. It will belong to the general timeseries. Each datapoint will be identified by an incremental ID, and has a timestamp corresponding to DateUTC value, and property value corresponding to the value of the property.

In order to compile this KG the following steps are done.

Steps for semantisation:

~~1.~~ Prepare your csv file.

In general, you have to add the ID (an incremental number) for each row to identify your data points and convert the date to ISO 8601 format (yyyy-mm-ddThh:mm:ss).

In this running example, we will extract the HMS_ID and add it as column to identify the general timeseries.

The script can be found in: https://newrepository.atosresearch.eu/index.php/f/1360600

The new csv file look like, and named *modified_combined_production_consumption_574.csv*.

DateUTC	TotalActiveEnergy+	TotalActiveEnergy-	production	consumpt	TS_id	HMS_id	id
2021-12-31T00:00:00	0.008	0	0	0.008	velverde_combined_production_consumption_574	574	1
2021-12-31T00:05:00	0.008	0	0	0.008	velverde_combined_production_consumption_574	574	2
2021-12-31T00:10:00	0.009	0	0	0.009	velverde_combined_production_consumption_574	574	3
2021-12-31T00:15:00	0.007	0	0	0.007	velverde_combined_production_consumption_574	574	4

~~2.~~1. Prepare your mapping file

In the mapping file, you will add your RML rules to map your csv file into the knoweldge grapgh.

A full resulting KG~~example~~ of the running example mapping file is given https://newrepository.atosresearch.eu/index.php/f/1360600

Here are some hints.

Commenté [HFZ3]: Caption

a mis en forme : Anglais (États-Unis)

a mis en forme : Anglais (États-Unis)

a mis en forme : Titre 3, Sans numérotation ni puces

Code de champ modifié

Commenté [HFZ4]: Caption

Code de champ modifié

A. Namespaces and prefixes

First, set your prefixes. You will add all the namespaces that you will use.

In this running example we are using:

Should be used in all cases

@prefix rr: <http://www.w3.org/ns/r2rml#> .

@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .

@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

@prefix rml: <http://semweb.mmlab.be/ns/rml#> .

@prefix ql: <http://semweb.mmlab.be/ns/ql#> .

CSDM modules (you can add or remove based on your needs)

@prefix qual: <https://w3id.org/omega-x/ontology/Quality/> .

@prefix prop: <https://w3id.org/omega-x/ontology/Property/> .

@prefix ets: <https://w3id.org/omega-x/ontology/EventTimeSeries/> .

@prefix eds: <https://w3id.org/omega-x/ontology/EnergyDataSets/> .

@prefix lec: <https://w3id.org/omega-x/ontology/LocalEnergyCommunities/> .

@prefix role: <https://w3id.org/omega-x/ontology/EnergyRole/>.

External ontologies that are needed for time and units. In ~~general~~general, you will need it.

@prefix unit: <http://qudt.org/vocab/unit/>.

@prefix time: <http://www.w3.org/2006/time#> .

Your namespace for your dataset

@prefix velverde: <https://w3id.org/omega-x/KG/velverde/> .

~~B.~~ Mapping timeseries

In this example, we only have timeseries. However you may have data collections or events. Additional examples can be found in the repository, in

Commenté [HFZ5]: Ajouter la convention de nommage

a mis en forme : Anglais (États-Unis)

a mis en forme : Titre 3, Sans numérotation ni puces

Tooling section.
Each triples in the KG should have a unique URI. Try to choose how you will set your URIs before the mapping.
Here is the example of the general timeseries mapping:

Commenté [HFZ6]: Police paragraphes non homogène

General Timeseries

```
# Map The Excel File to time series for each home management system
<#TimeSeriesProductionConsumptionProfile>
   rml:logicalSource [
       rml:source "/home/lina/SDM-RDFizer/example/LECGamificationService/Velverde_Data/modified_combined_production_consumption_574.csv"; #Link to the input csv file you want to map
       rml:referenceFormulation ql:CSV
   ];
   rr:subjectMap [
       rr:template "velverde:TimeSeries/{TS_id}";  #Identify each TS by the ID we genrated.
   ];
   rr:predicateObjectMap [
         rr:predicate rdf:type ;
         rr:object ets:TimeSeries # say it is a Timeseries
] ;
   rr:predicateObjectMap [
            rr:predicate rdf:type ;
            rr:object eds:EnergyDataSet  # say it is an Energy DataSet to add the exchange context and the evaluation point
] ;
   rr:predicateObjectMap [
       rr:predicate ets:hasStep;
       rr:objectMap [
          rr:datatype time:Duration ;
          rr:constant "PT5M" ; #set the step to 15 minutes
       ];
   ];
   rr:predicateObjectMap [
       rr:predicate eds:includesTechnicalContext;
       rr:objectMap [rr:template "velverde:TechnicalContext/{TS_id}";]
```

Commenté [HFZ7]: Mettre un snipet de code au lieu du texte

This document is issued within the frame and for the purpose of the OMEGA-X project. This project has received funding from the European Union's Horizon Europe Framework Programme under Grant Agreement No. 101069287. The opinions expressed and arguments employed herein do not necessarily reflect the official views of the European Commission.
The dissemination of this document reflects only the author's view, and the European Commission is not responsible for any use that may be made of the information it contains. This deliverable is subject to final acceptance by the European Commission.

```
   ]; #add the technical context for the evaluation point
      rr:predicateObjectMap [
       rr:predicate eds:isExchangedIn;
       rr:objectMap [rr:template "velverde:ExchangeContext/{TS_id}";]
   ]. #add the exchange context to say that EDP is the data provider
```

An example of the evaluation point:

```
<#TEvaluationPoint>
   rml:logicalSource [
       rml:source "/home/lina/SDM-
RDFizer/example/LECGamificationService/Velverde_Data/modified_combine
d_production_consumption_574.csv"; # Link to the input csv file
       rml:referenceFormulation ql:CSV
   ];
    rr:subjectMap [
       rr:template "velverde:EvaluationPoint/{HMS_id}";
   ];
   rr:predicateObjectMap [
   rr:predicate rdf:type ;
   rr:object eds:EvaluationPoint  # Time Series
] ;
   rr:predicateObjectMap [
   rr:predicate rdf:type ;
   rr:object lec:HomeManagementSystem  # Here you can check the UCF
modules to determine the type of your evaluation point.
] .
```

The timeseries associated to each column

```
<#TSActiveEnergyImported>
   rml:logicalSource [
       rml:source "/home/lina/SDM-
RDFizer/example/LECGamificationService/Velverde_Data/modified_combine
d_production_consumption_574.csv"; # Link to the input CSV file
       rml:referenceFormulation ql:CSV
   ];
   rr:subjectMap [
       rr:template
"velverde:TimeSeries/ActiveEnergyImported/{HMS_id}"#Identify each
timeseries
```

This document is issued within the frame and for the purpose of the OMEGA-X project. This project has received funding from the European Union's Horizon Europe Framework Programme under Grant Agreement No. 101069287. The opinions expressed and arguments employed herein do not necessarily reflect the official views of the European Commission.

```
  ];
  rr:predicateObjectMap [
     rr:predicate rdf:type ;
     rr:object ets:TimeSeries  # Time Series
  ];
  rr:predicateObjectMap [
     rr:predicate prop:isAboutProperty;
     rr:objectMap [rr:template
"velverde:Property/ActiveEnergyImported/{HMS_id}";]#Assign a property,
  ] ;
  rr:predicateObjectMap [
     rr:predicate qual:hasAggregationKind ;
     rr:objectMap [rr:template "velverde:AggegationKind/Total";]#If it exists,
precise the aggregation
  ] ;
    rr:predicateObjectMap [
     rr:predicate ets:isElementOf ;
     rr:objectMap [rr:template "velverde:TimeSeries/{TS_id}";] #It should be
element of the general timeseries
  ] ;
   rr:predicateObjectMap [
     rr:predicate prop:hasUnit ; #Assign the unit. Check the qudt ontology
     rr:objectMap [
        rr:constant <http://qudt.org/vocab/unit/KiloW-HR>
     ]
  ] .
```

Commenté [HFZ8]: Snippet de code

Define your property

```
<#PropertyActiveEnergyImported>
  rml:logicalSource [
     rml:source "/home/lina/SDM-
RDFizer/example/LECGamificationService/Velverde_Data/modified_combine
d_production_consumption_574.csv"; # Link to the input csv file
     rml:referenceFormulation ql:CSV
  ];
   rr:subjectMap [
     rr:template "velverde:Property/ActiveEnergyImported/{HMS_id}";
  ];
  rr:predicateObjectMap [
```

This document is issued within the frame and for the purpose of the OMEGA-X project. This project has received funding from the European Union's Horizon Europe Framework Programme under Grant Agreement No. 101069287. The opinions expressed and arguments employed herein do not necessarily reflect the official views of the European Commission.
The dissemination of this document reflects only the author's view, and the European Commission is not responsible for any use that may be made of the information it contains. This deliverable is subject to final acceptance by the European Commission.

```
    rr:predicate rdf:type ;
    rr:object prop:ActiveEnergyImported  # check the property module of the CSDM to find your property
] .
```

Add your data points

```
<#DPActiveEnergyImported>
    rml:logicalSource [
        rml:source "/home/lina/SDM-RDFizer/example/LECGamificationService/Velverde_Data/modified_combined_production_consumption_574.csv"; # Link to the input CSV file
        rml:referenceFormulation ql:CSV
    ];
    rr:subjectMap [
        rr:template "velverde:ActiveEnergyImported/DataPoint_{id}/TimeSeries_{HMS_id}"
    ];
    rr:predicateObjectMap [
        rr:predicate rdf:type ;
        rr:object ets:DataPoint  # Data Point
    ];
    rr:predicateObjectMap [
        rr:predicate ets:dataTime ;
        rr:objectMap [rml:reference "DateUTC" ; rr:datatype xsd:dateTime]
    ] ;# Assign to the Data Point the timestamp from the column DateUTC of your csv
    rr:predicateObjectMap [
        rr:predicate ets:belongsTo;
        rr:objectMap [rr:template "velverde:TimeSeries/ActiveEnergyImported/{HMS_id}"]
    ] ;  # the Data Point belongs to the timeseries you have defined
     rr:predicateObjectMap [
        rr:predicate ets:hasDataValue;
        rr:objectMap [rr:template "velverde:ActiveEnergyImported/PropertyValue/DataPoint_{id}/TimeSeries_{HMS_id}"] #the data point has property value.
    ] .
```

Add your property values

This document is issued within the frame and for the purpose of the OMEGA-X project. This project has received funding from the European Union's Horizon Europe Framework Programme under Grant Agreement No. 101069287. The opinions expressed and arguments employed herein do not necessarily reflect the official views of the European Commission.

```
<#PValueActiveEnergyImported>
   rml:logicalSource [
      rml:source "/home/lina/SDM-RDFizer/example/LECGamificationService/Velverde_Data/modified_combined_production_consumption_574.csv"; # Link to the input CSV file
      rml:referenceFormulation ql:CSV
   ];
   rr:subjectMap [
      rr:template "velverde:ActiveEnergyImported/PropertyValue/DataPoint_{id}/TimeSeries_{HMS_id}"
   ];
   rr:predicateObjectMap [
      rr:predicate rdf:type ;
      rr:object ets:PropertyValue  # Property Value
   ];
   rr:predicateObjectMap [
     rr:predicate ets:value ;
     rr:objectMap [
         rml:reference "TotalActiveEnergy+";
         rr:datatype xsd:double
      ]#assign the value from the corresponding column in your csv
   ] .
```

~~3.~~2. Prepare your config.ini file

Set your main directory, and determine the datasets you want to semantify. In mapping, put the path to the RML mapping file you have generated from the previous step.

```
[default]
main_directory: /home/lina/SDM-RDFizer/example/LECGamificationService/Velverde_Data
#Any path used in Python, unless explicitly defined, is relative to the
#directory in which the code is executed. To avoid confusion, it is
#recommended that the user specifies the absolute path for the main directory.


[datasets]
number_of_datasets: 1
output_folder: ${default:main_directory}/output
remove_duplicate: yes
```

This document is issued within the frame and for the purpose of the OMEGA-X project. This project has received funding from the European Union's Horizon Europe Framework Programme under Grant Agreement No. 101069287. The opinions expressed and arguments employed herein do not necessarily reflect the official views of the European Commission.

```
all_in_one_file: no
name:joinCondition
enrichment: yes
ordered: yes
output_format: turtle


[dataset1]
name: VelverdeProductionConsumptionKG_538
mapping:
${default:main_directory}/generated_mappings/mapping_modified_combined_production_consumption_538.ttl
```

~~4.~~3. Run the rdfizer to get your KG

Commenté [HFZ9]: Mettre le lien du wiki du rdfizer pour des explications exaustives suivant l'OS et docker/source file

Run the command rdfizer –c config.ini
You will get you KG in ttl.

You can use any triple stores or other tools to visualize and query your KG.

Python and RDFlib package

Programming languages such as python offer a powerful and flexible approach to transform raw data format into an RDF knowledge graph, using specialized libraries (RDFlib[2] for python). Using python scripts enable a dynamic alternative for predefined mapping rules and a swift integration into preprocessing pipelines. Beyond easy integration, the script-based semantisation enables scalability and integration with external databases and APIs.

Workflow example

The following paragraphs include a running example of semantisation using python script, applied to MAIA pilot site. The used dataset describes a time series recorder by a smart meter for energy production activities.

Install required packages.

Before running the script, ensure to installl the rdflib package by running :

```
pip install rdflib
```

Figure 2 RDFLib installation command

CSV file preparation

Ensure that your csv file includes the required data for semantisation. Temporal values should be normalized to ISO 8601 format.[3]

A	B	C	D	E	F	G	H	I	J	K	L	M	N
device,tstamp,dstamp,CurrL1_A,CurrL2_A,CurrL3_A,ActPow_kW,AppPow_kVA,ReacPow_kvar,Energy_kWh,VolL1_V,VolL2_V,VolL3_V,THDUL1,THDUL2,THDUL3,THDIL1,THDIL2,THDIL3,cosphi													
TorreLidador,2024-07-09 00:00:23,23:47/ 08.07.2024 (UTC 2),149.21,142.69,156.86,105.97,106.72,755355.0,,,,,,,,,,													

Figure 3Input csv dataset

Python script for transformation

The following steps detail how to build a python script for semantisation.

1. Namespaces definition

[2] https://rdflib.readthedocs.io/en/stable/
[3] https://www.iso.org/fr/iso-8601-date-and-time-format.html

This document is issued within the frame and for the purpose of the OMEGA-X project. This project has received funding from the European Union's Horizon Europe Framework Programme under Grant Agreement No. 101069287. The opinions expressed and arguments employed herein do not necessarily reflect the official views of the European Commission.
The dissemination of this document reflects only the author's view, and the European Commission is not responsible for any use that may be made of the information it contains. This deliverable is subject to final acceptance by the European Commission.

a mis en forme : Anglais (États-Unis)

a mis en forme : Paragraphes solidaires

a mis en forme : Légende, Retrait : Gauche : 0 cm

a mis en forme : Anglais (États-Unis)

a mis en forme : Paragraphes solidaires

a mis en forme : Légende

```
ETS = Namespace("https://w3id.org/omega-x/ontology/EventTimeSeries/")
PROP = Namespace("https://w3id.org/omega-x/ontology/Property/")
MAIA = Namespace("https://w3id.org/omega-x/KG/MaiaPTT/")
UNIT = Namespace("http://qudt.org/schema/qudt/Unit/")
EDS = Namespace("https://w3id.org/omega-x/ontology/EnergyDataset/")
```

a mis en forme : Paragraphes solidaires

Figure 4Namespace definition

2. *Graph creation*

The graph creation step involves the declaration of a new knowledge graph and binding different defined namespaces.

```
# Graph definition and prefixes binding
g = Graph()
g.bind("ets", ETS)
g.bind("prop", PROP)
g.bind("maia", MAIA)
g.bind("eds", EDS)
g.bind("xsd", XSD)
g.bind("unit", UNIT)
```

a mis en forme : Paragraphes solidaires

Figure 5Graph creation step

a mis en forme : Sans numérotation ni puces

3. *Properties mapping*

The input CSV file include values of dynamic properties identified by the CSV header. In this step, a mapping of these column names is achieved to identify corresponding CSDM standard property names, defined in the property taxonomy module[4].

Make sure each column name is properly mapped to a CSDM property. If the property you're looking for is missing, please contact CSDM developers' team.

[4] https://w3id.org/omega-x/ontology/Property

This document is issued within the frame and for the purpose of the OMEGA-X project. This project has received funding from the European Union's Horizon Europe Framework Programme under Grant Agreement No. 101069287. The opinions expressed and arguments employed herein do not necessarily reflect the official views of the European Commission.

```python
# mapping column names to CSDM properties
property_mapping = {
    "CurrL1_A": PROP.CurrentL1,
    "CurrL2_A": PROP.CurrentL2,
    "CurrL3_A": PROP.CurrentL3,
    "ActPow_kW": PROP.ActivePower,
    "AppPow_kVA": PROP.ApparentPower,
    "ReacPow_kvar": PROP.ReactivePower,
    "Energy_kWh": PROP.Energy,
    "VolL1_V": PROP.VoltageL1,
    "VolL2_V": PROP.VoltageL2,
    "VolL3_V": PROP.VoltageL3,
    "THDUL1": PROP.TotalHarmonicDistorsionUL1,
    "THDUL2": PROP.TotalHarmonicDistorsionUL2,
    "THDUL3": PROP.TotalHarmonicDistorsionUL3,
    "THDIL1": PROP.TotalHarmonicDistorsionIL1,
    "THDIL2": PROP.TotalHarmonicDistorsionIL2,
    "THDIL3": PROP.TotalHarmonicDistorsionIL3,
    "cosphi": PROP.CosinusPhi
}
```

Figure 6 Property mapping step

4. *Units mapping*

Each property of your dataset should be associated with a unit of measure. The CSDM property module lists possible units for each property.

a mis en forme : Paragraphes solidaires

a mis en forme : Sans numérotation ni puces

```python
# adding unit names
unit_mapping = {
    "ActPow_kW": UNIT.KiloW,
    "AppPow_kVA": UNIT.KiloVA,
    "ReacPow_kvar": UNIT.kiloV,
    "CurrL1_A": UNIT.A,
    "CurrL2_A": UNIT.A,
    "CurrL3_A": UNIT.A,
    "VolL1_V": UNIT.V,
    "VolL2_V": UNIT.V,
    "VolL3_V": UNIT.V,
    "Energy_kWh": UNIT.KiloWHR,
    "cosphi": UNIT.PERCENT
}
```

Figure 7Unit mapping step

5. *CSV rows processing*

Since the input dataset is produced by one device, recording different properties values, the resulting knowledge graph should contain one timeseries with a unique context (infrastructure element description).

```python
# Main time series instance (device column)
mainTimeSeries = URIRef("https://w3id.org/omega-x/KG/MaiaPTT/TorreLidadorTimeSeries")

g.add((mainTimeSeries, RDF.type, ETS.TimeSeries))
g.add((mainTimeSeries, RDF.type, EDS.EnergyDataset))
g.add((mainTimeSeries, ETS.hasStep, Literal("PT1M", datatype=XSD.duration)))
```

Figure 8Timeseries step

Additionally, each row will be transformed into a dataCollection (see ETS CSDM module [5]). Each datacollection will get two description timestamp: creationTime refers to the digital creation time (from the software), while the collectionTime refers to the real recording time.

a mis en forme : Paragraphes solidaires

a mis en forme : Sans numérotation ni puces

a mis en forme : Paragraphes solidaires

[5] https://fz-hannou.github.io/Omega-X/EventsTimeSeries.html

```python
for row in reader:
    realTime = clean_time(row['tstamp'])
    digitalTime = clean_time(row['dstamp'])
    dataCollection = URIRef(f"https://w3id.org/omega-x/KG/MaiaPTT/DataCollection/{incrementing_id}")

    g.add((dataCollection, RDF.type, ETS.DataCollection))
    g.add((dataCollection, ETS.collectionTime, Literal(realTime, datatype=XSD.dateTime)))
    g.add((dataCollection, ETS.creationTime, Literal(digitalTime, datatype=XSD.dateTime)))
    g.add((dataCollection, ETS.isElementOf, mainTimeSeries))
```

Figure 9 Data Collection step

Then, within a row, cells store values of datapoints representing property values. Each property value is then linked to a property and a unit, and indicates a Literal data value.

6. Serialization

Finally, choose your graph serialization format. Options include turtle (.ttl) and json-ld (.json).

```python
semantify(csvFile)
# The output format is set to: json-ld. It can be changed to turtle. In the latter case, the output file should be .ttl
output = g.serialize(format="json-ld")

with open("TorreLidador_2024_07_09.json", "w") as f:
    f.write(output)
```

Figure 10Serialization step

a mis en forme : Paragraphes solidaires

a mis en forme : Sans numérotation ni puces

a mis en forme : Paragraphe de liste, Sans numérotation ni puces

a mis en forme : Paragraphes solidaires

a mis en forme : Légende, Gauche, Taquets de tabulation : Pas à 4,02 cm