Focus on what you're looking for rather than how to get it: modules to hide the complexity of SPARQL queries

Yael Tirlet

IRISA Dyliss Team

yael.tirlet@irisa.fr

PEPI IBIS
October 15, 2025

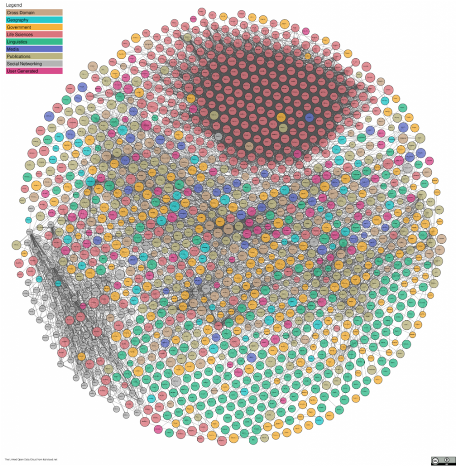
Overview

1. Background: The semantic web to integrate and query life science data

2. Our contribution: Modules to bridge the gap between ”What” and ”How”
   - Our vision
   - Theory
   - Example on Holograph data schema

3. Conclusion

The semantic Web: a unified framework for integrating and querying complementary knowledge bases



The Linked Open Data Cloud from lod-cloud.net

- RDF for data representation
- OWL for knowledge representation
- SPARQL for querying them in a unified way

Many databases in life science are now accessible

Advantages of Semantic Web

Why use semantic web

- Manage knowledge at different precision level
- Query heterogeneous data, backed by a knowledge base
- Interoperability: query multiple knowledge bases together

Why use semantic web

- Manage knowledge at different precision level
- Query heterogeneous data, backed by a knowledge base
- Interoperability: query multiple knowledge bases together

Difficulties

- Intrinsic complexity of Life science
- Technical complexity for querying the model

Why use semantic web

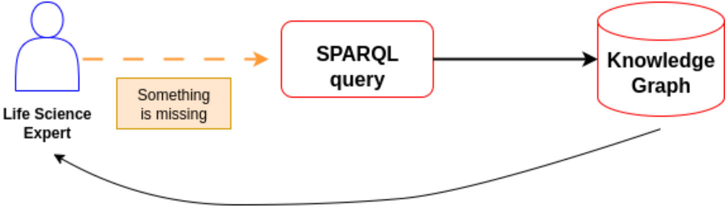
- Manage knowledge at different precision level
- Query heterogeneous data, backed by a knowledge base
- Interoperability: query multiple knowledge bases together

Difficulties

- Intrinsic complexity of Life science
- Technical complexity for querying the model

⇒ **Barrier to the adoption of these resources**

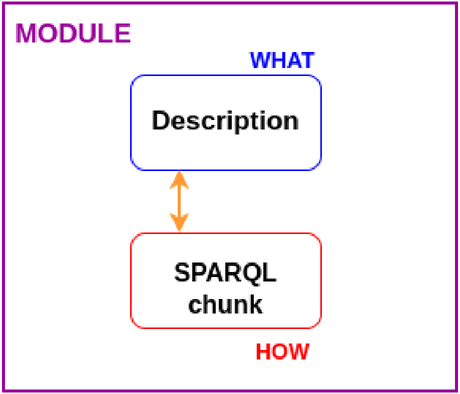
Our vision: The issue



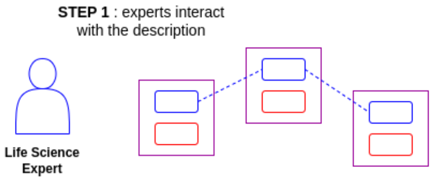
Current bottleneck: interface between

- life Science experts (who focus on what they want)
- high-end engineering systems (which address how to do it reliably)

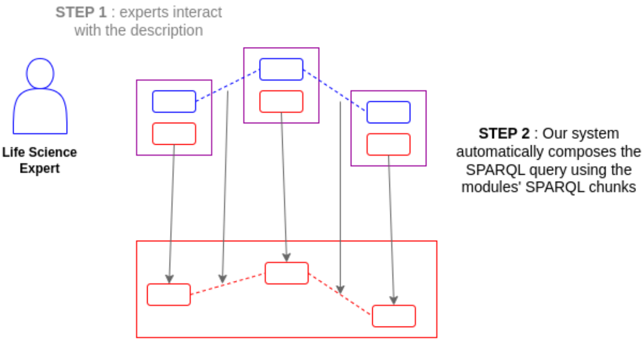
Our vision: Modules providing the missing connection between the what and the how



Our vision: Modules providing the missing connection between the what and the how



Our vision: Modules providing the missing connection between the what and the how

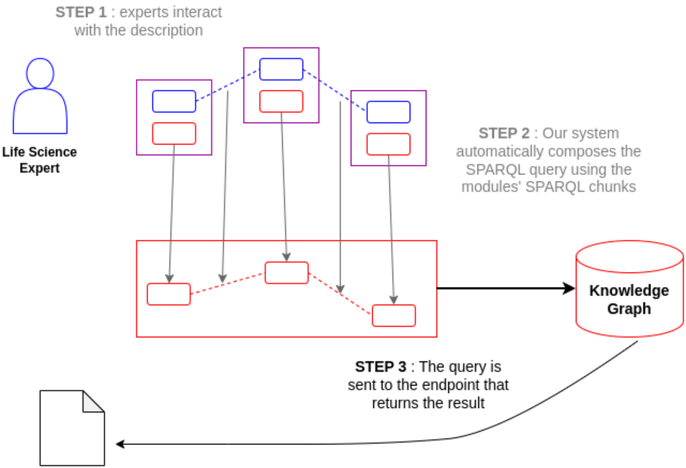


STEP 1 : experts interact with the description

Life Science Expert

STEP 2 : Our system automatically composes the SPARQL query using the modules' SPARQL chunks

Our vision: Modules providing the missing connection between the what and the how

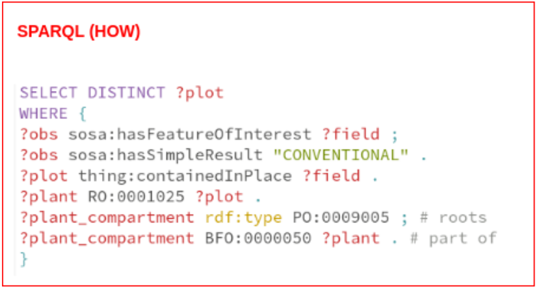


STEP 1 : experts interact with the description

STEP 2 : Our system automatically composes the SPARQL query using the modules' SPARQL chunks

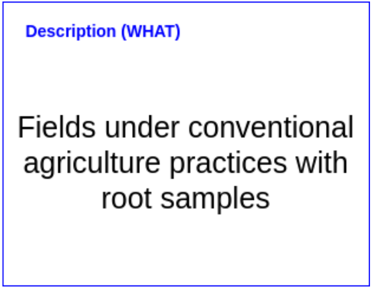
STEP 3 : The query is sent to the endpoint that returns the result

Example



SPARQL (HOW)

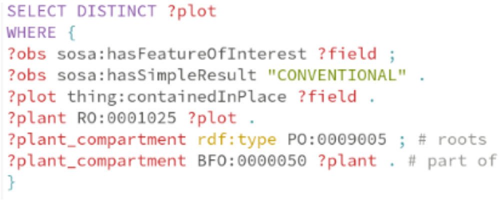
```
SELECT DISTINCT ?plot
WHERE {
?obs sosa:hasFeatureOfInterest ?field ;
?obs sosa:hasSimpleResult "CONVENTIONAL" .
?plot thing:containedInPlace ?field .
?plant RO:0001025 ?plot .
?plant_compartment rdf:type PO:0009005 ; # roots
?plant_compartment BFO:0000050 ?plant . # part of
}
```



Description (WHAT)

Fields under conventional agriculture practices with root samples

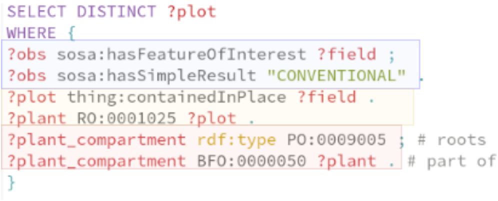
Example



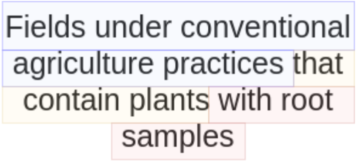
```
SELECT DISTINCT ?plot
WHERE {
?obs sosa:hasFeatureOfInterest ?field ;
?obs sosa:hasSimpleResult "CONVENTIONAL" .
?plot thing:containedInPlace ?field .
?plant RO:0001025 ?plot .
?plant_compartment rdf:type PO:0009005 ; # roots
?plant_compartment BFO:0000050 ?plant . # part of
}
```

Fields under conventional agriculture practices **that contain plants** with root samples

Example

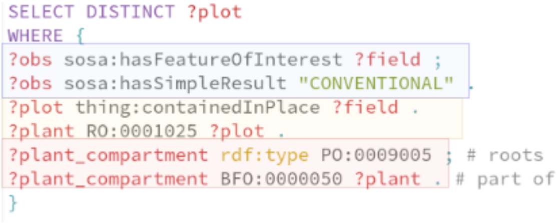


```
SELECT DISTINCT ?plot
WHERE {
?obs sosa:hasFeatureOfInterest ?field ;
?obs sosa:hasSimpleResult "CONVENTIONAL" .
?plot thing:containedInPlace ?field .
?plant RO:0001025 ?plot .
?plant_compartment rdf:type PO:0009005 ; # roots
?plant_compartment BFO:0000050 ?plant . # part of
}
```



Fields under conventional agriculture practices that contain plants with root samples

Example



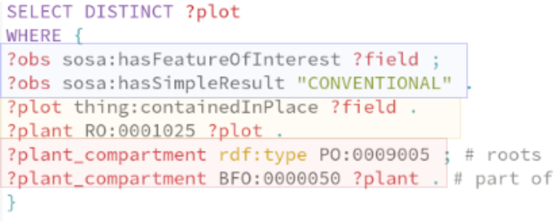
```
SELECT DISTINCT ?plot
WHERE {
?obs sosa:hasFeatureOfInterest ?field ;
?obs sosa:hasSimpleResult "CONVENTIONAL" .
?plot thing:containedInPlace ?field .
?plant RO:0001025 ?plot .
?plant_compartment rdf:type PO:0009005 ; # roots
?plant_compartment BFO:0000050 ?plant . # part of
}
```

Field agriculture type

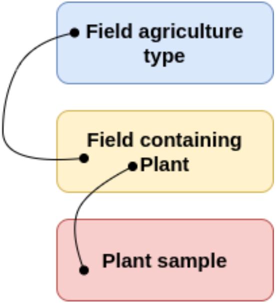
Field containing Plant

Plant sample

Example



```
SELECT DISTINCT ?plot
WHERE {
?obs sosa:hasFeatureOfInterest ?field ;
?obs sosa:hasSimpleResult "CONVENTIONAL" .
?plot thing:containedInPlace ?field .
?plant RO:0001025 ?plot .
?plant_compartment rdf:type PO:0009005 ; # roots
?plant_compartment BFO:0000050 ?plant . # part of
}
```



Theory

What is a module

Definition of a module

- (Component of a) SPARQL query
- Has a semantics in the targeted domain
- With which the user can interact
- Reusable

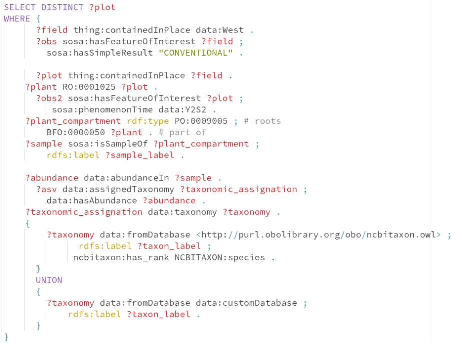
Definition of a Port

A port is a variable with which the user can interact:

- Select it
- Connect to another port
- Fix a value

Holograph

Query 1: SPARQL



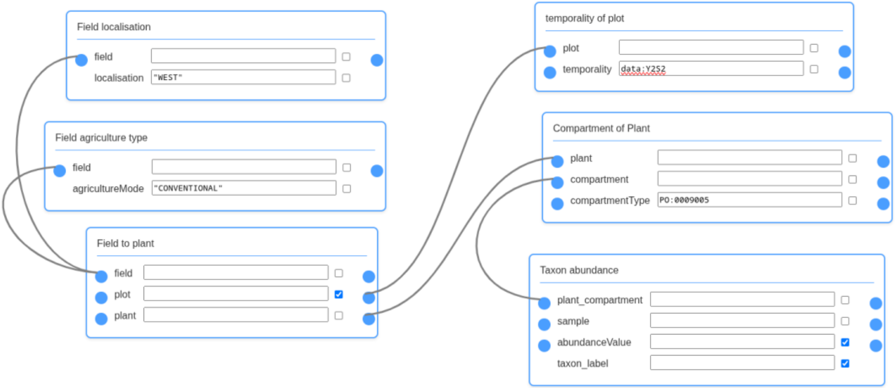
```
SELECT DISTINCT ?plot
WHERE {
      ?field thing:containedInPlace data:West .
      ?obs sosa:hasFeatureOfInterest ?field ;
        sosa:hasSimpleResult "CONVENTIONAL" .

      ?plot thing:containedInPlace ?field .
    ?plant RO:0001025 ?plot .
      ?obs2 sosa:hasFeatureOfInterest ?plot ;
         sosa:phenomenonTime data:Y2S2 .
    ?plant_compartment rdf:type PO:0009005 ; # roots
        BFO:0000050 ?plant . # part of
    ?sample sosa:isSampleOf ?plant_compartment ;
        rdfs:label ?sample_label .

    ?abundance data:abundanceIn ?sample .
      ?asv data:assignedTaxonomy ?taxonomic_assignation ;
        data:hasAbundance ?abundance .
    ?taxonomic_assignation data:taxonomy ?taxonomy .
    {
        ?taxonomy data:fromDatabase <http://purl.obolibrary.org/obo/ncbitaxon.owl> ;
              rdfs:label ?taxon_label ;
             ncbitaxon:has_rank NCBITAXON:species .
      }
      UNION
      {
        ?taxonomy data:fromDatabase data:customDatabase ;
            rdfs:label ?taxon_label .
      }
}
```

Taxa found in root microbiota from fields in western France under conventional agriculture practices in the second sampling of the second year

Very complicated to read for non-SPARQL-litterate people.

Holograph

Query 1: modules



Field localisation

- field □
- localisation "WEST" □

Field agriculture type

- field □
- agricultureMode "CONVENTIONAL" □

Field to plant

- field □
- plot ☑
- plant □

temporality of plot

- plot □
- temporality data:Y2S2 □

Compartment of Plant

- plant □
- compartment □
- compartmentType PO:0009005 □

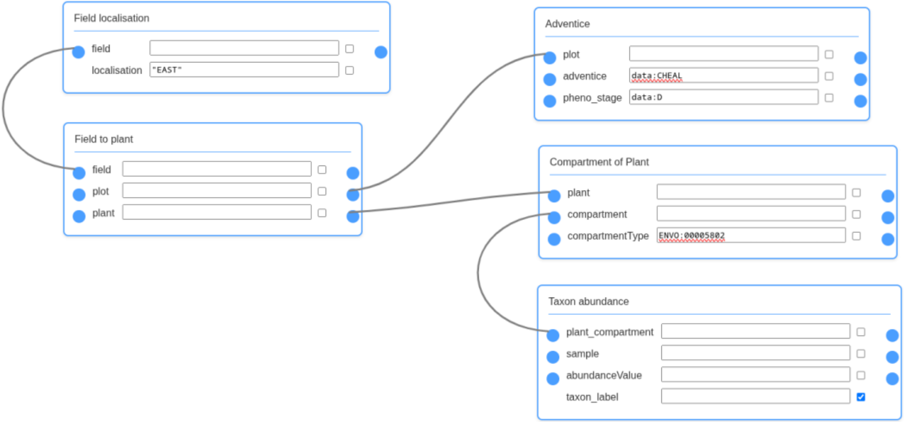
Taxon abundance

- plant_compartment □
- sample □
- abundanceValue ☑
- taxon_label ☑

Module illustration is more understandable for non-SPARQL-litterate people.

Holograph

Query 2: Re-using modules



Field localisation

field □

localisation "EAST" □

Field to plant

field □

plot □

plant □

Adventice

plot □

adventice data:CHEAL □

pheno_stage data:D □

Compartment of Plant

plant □

compartment □

compartmentType ENVO:00005802 □

Taxon abundance

plant_compartment □

sample □

abundanceValue □

taxon_label ☑

Modules can easily be reused for other queries.

Conclusion

We propose a system that enables users to:

- create a library of modules
- simplify the creation of a SPARQL query by assembling modules

What I have not talked about

- How to model a Module in informatics ?
- How to connect Modules together ?
- How to compute the SPARQL query ? (efficiency, optimization...)
- What is a good module ?

$\Rightarrow$ **PhD in progress**

Acknowledgments

Thank you for your attention

Dyliss team, IRISA

University of Rennes

ENDOVIRE ANR

This work was supported by the French National Research Agency, project Endovire ANR-22-CE20-0005