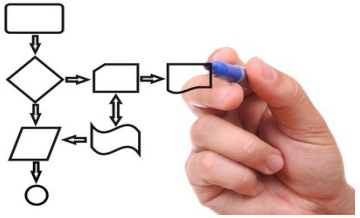
Semantic annotation pipeline & data publication

Functional and conceptual aspects regarding our approach to automate the production of semantic data from relational Databases

Functional ✓

~~Algorithm~~



Conceptual ✓

~~Code~~

Ghislaine Monet
Damien Maurice
Antoine Schellenberger
Yahiaoui Rachid

INTRODUCING THE PROBLEM

What we are talking about ..

To have a tool to ensure the production of semantic data as generic and automated as possible.

Simplify the production of semantic data from existing DBs of the AnaEE-F infra and more.

By using open source semantic tools + some specific Developments

which I would talk about later

INVENTORY OF THE SEMANTIC OPEN SOURCE PROJECT

Before Developing anything

On-The-Fly Translation tools

D2RQ

- **On-the-fly translation allows publishing of RDF from large live databases** ✓
- ~~Not intuitive mapping~~ ~~(Especially for those who handle SQL)~~ ✗
- ~~Fail **last**~~ Errors Mapping Raised at runTime
- ~~No Native GUI ! External Project (AuReli)~~ ✗

ontop

in addition to offering on the fly translations

- **On-the-fly Ontology-based Data Access** ✓
- **Intuitive Mapping (using SQL)** ✓ the most interesting feature provided by ONTOP
- ~~GUI integrated with Protege~~ ✗ ✓
- ~~Support SPARQL 1.0~~ ✗

INVENTORY OF THE SEMANTIC OPEN SOURCE PROJECT

Semantic DataBases..
Two Kinds ...

TripleStore

* Sesame

- Robustness : **K.O**
- Scaling out : **K.O**
- Performance : **Err**

* Sol-RDF

- Robustness : **OK**
- Scaling out : **OK**
- Performance :

REST

* BlazeGraph

- Robustness : **OK**
- Scaling out : **OK ***
- Performance : **OK**

* Corese

- Robustness : **OK**
- Scaling out :
- Performance : **OK**

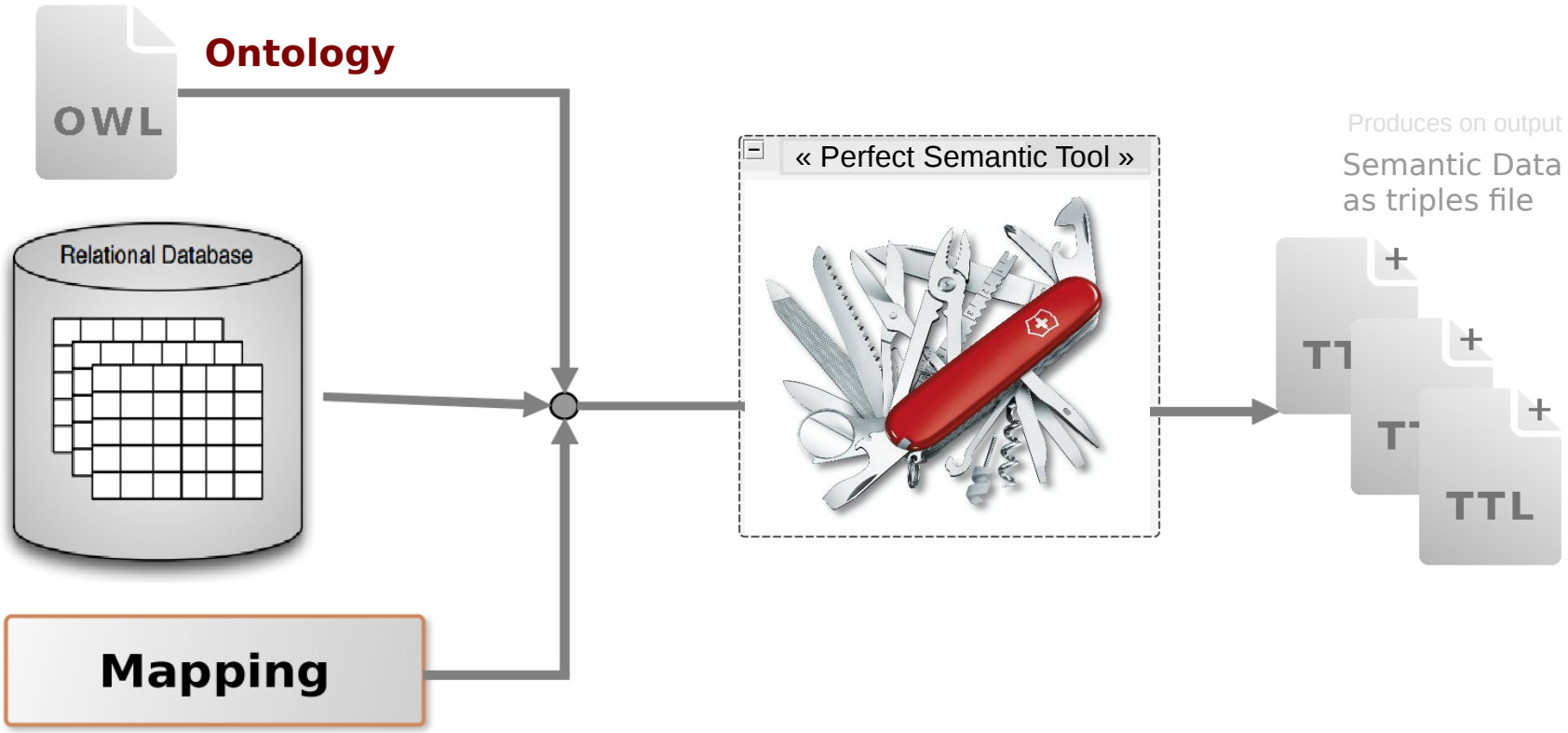
* : Version 1.5.3

Graph Datatabases

Has more generalized structure than a triplestore

what's the perfect semantic tool in our case ?

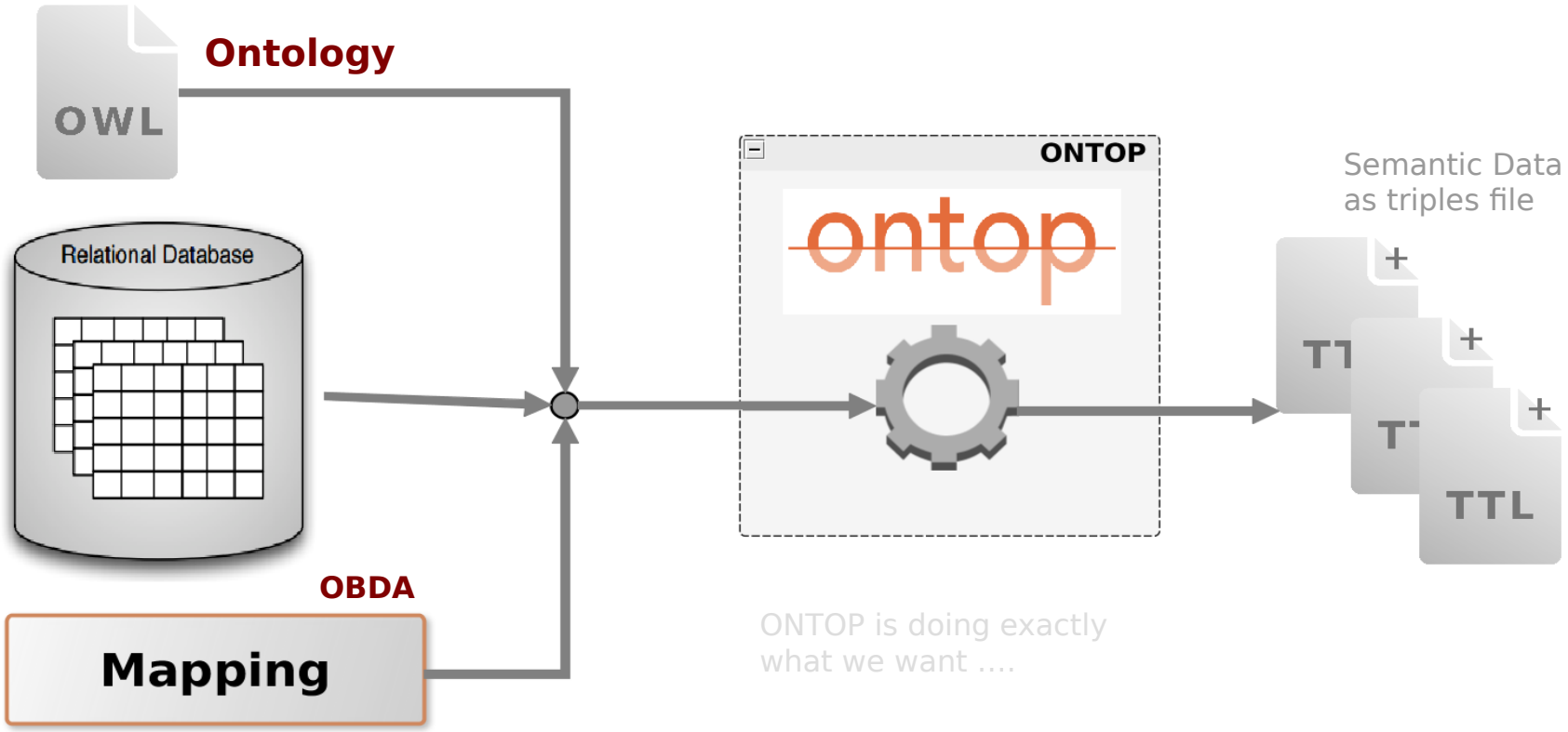
AUTOMATION APPROACH



Something which says how relational data are transformed to semantic data graph

AUTOMATION APPROACH

Good news :-)

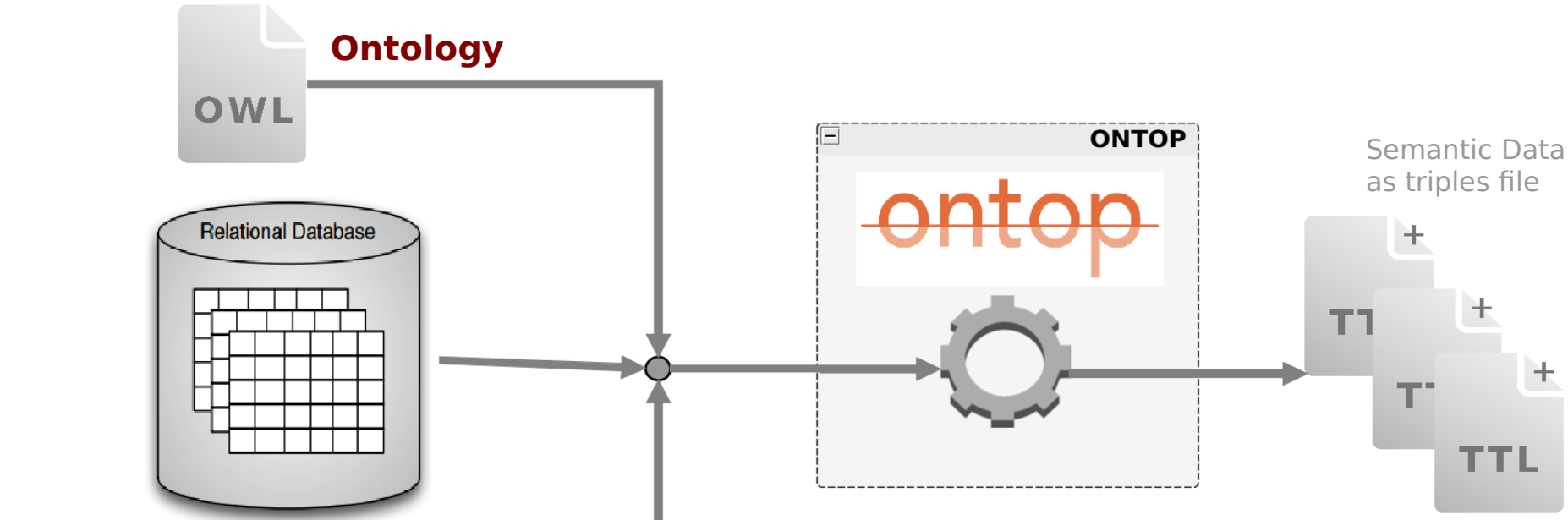


ONTOP is doing exactly what we want ....

Something which says how relational data are transformed to semantic data graph

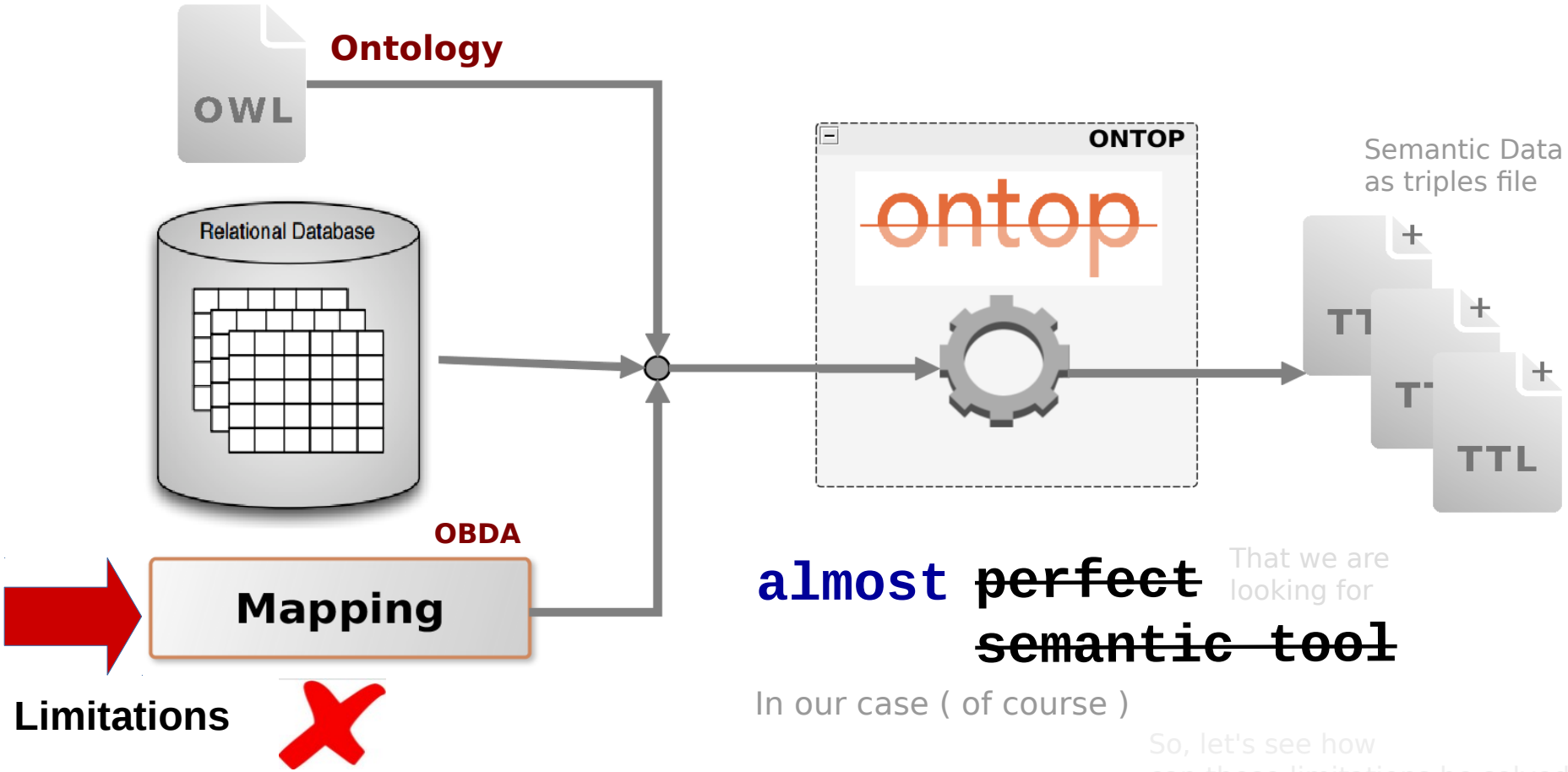
But ??

AUTOMATION APPROACH



Due to these limitations ...

AUTOMATION APPROACH



almost ~~**perfect semantic tool**~~

That we are looking for

In our case (of course)

So, let's see how can these limitations be solved..

Mapping has to be created by hand

AUTOMATION APPROACH

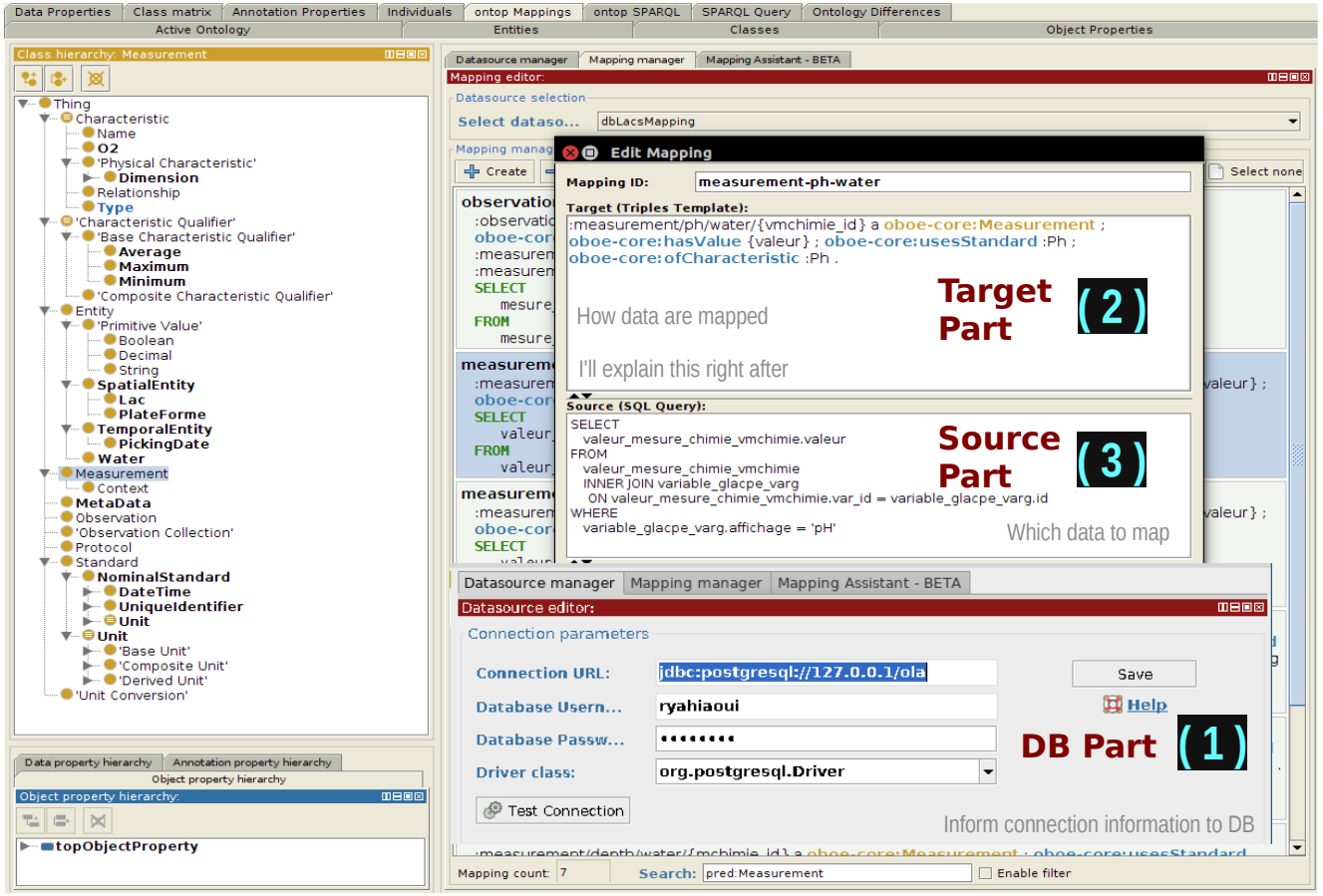
Level of Automation

Ontop From Protegé *

Ontop provides a nice integrated GUI into Protegé to Create mapping files.

Once Protegé opened, we distinguish **3 main parts**

Protegé : open-source Ontology editor



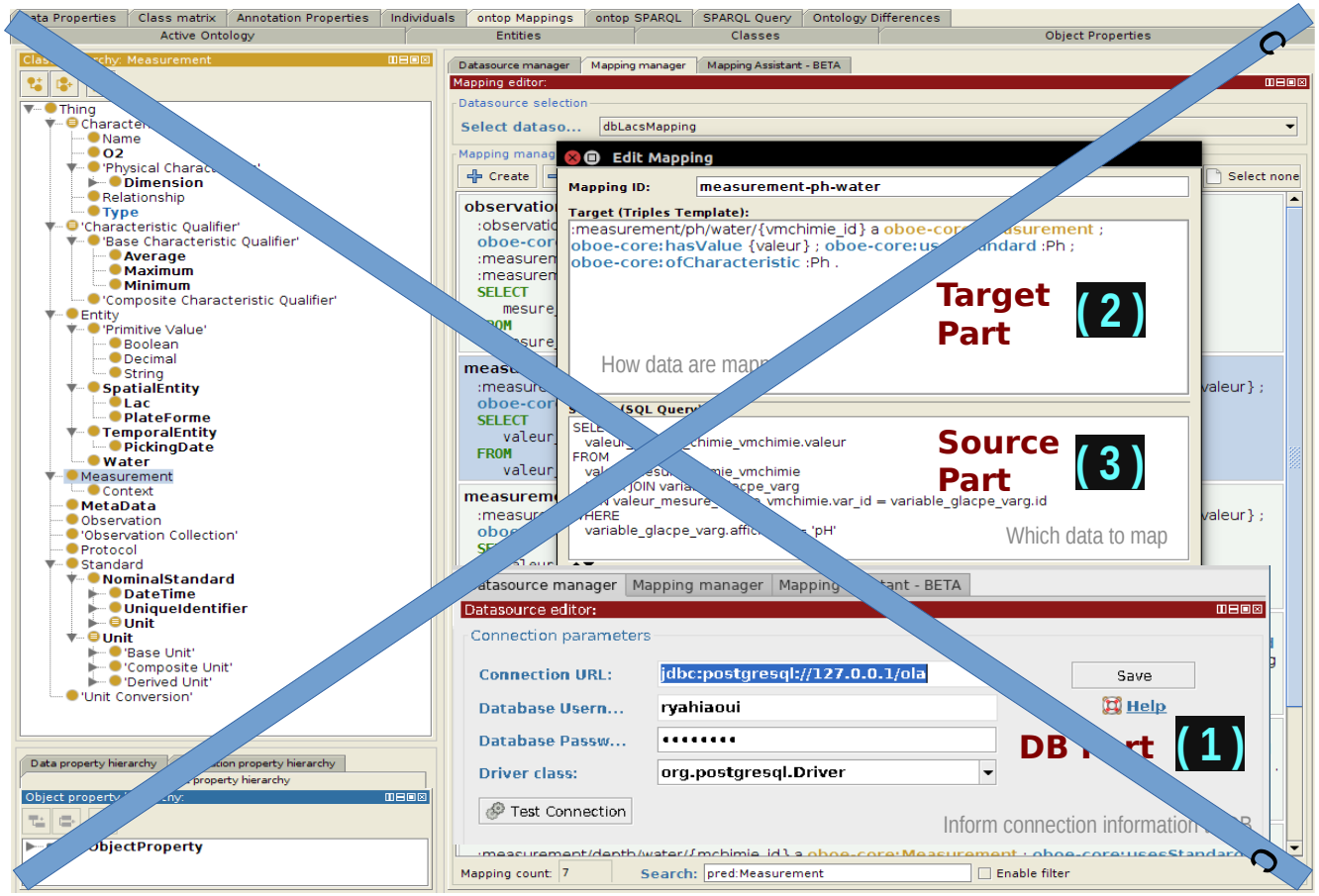
AUTOMATION APPROACH

Level of Automation

Ontop From Protegé *

This approach is

Not an effective solution for automation



Take a look at what there ..

AUTOMATION APPROACH

Level of Automation

Behind the scene ... Ontop manipulates OBDA File (based on R2RML langage)

(R2RML is a W3C recommended RDB-to-RDF mapping langage)

```
[PrefixDeclaration]                                    Interesting thing ..
rdf:  http://www.w3.org/1999/02/22-rdf-syntax-ns#      The 3 parts that was discussed previously
oboe-core: http://ecoinformatics.org/oboe/oboe.1.0/oboe-core.owl#
oboe-temporal: http://ecoinformatics.org/oboe/oboe.1.0/oboe-temporal.owl#
xsd: http://www.w3.org/2001/XMLSchema#
: http://www.anaee france.fr/ontology/anaee-france_ontology#
oboe-standard: http://ecoinformatics.org/oboe/oboe.1.0/oboe-standards.owl#
oboe-characteristics: http://ecoinformatics.org/oboe/oboe.1.0/oboe-characteristics.owl#
oboe-spatial: http://ecoinformatics.org/oboe/oboe.1.0/oboe-spatial.owl#
oboe-standards:          http://ecoinformatics.org/oboe/oboe.1.0/oboe-standards.owl#
rdfs:            http://www.w3.org/2000/01/rdf-schema#

[SourceDeclaration]
sourceUri        dbLacsMapping
connectionUrl    jdbc:postgresql://127.0.0.1/ola?sendBufferSize=5000
username         ryahiaoui
password         yahiaoui                    (1) Inform connection information to DB
driverClass      org.postgresql.Driver

[MappingDeclaration] @collection [[

mappingId        (52) ola_characteristic_depthRelativeToSurface_min
target           :ola/characteristic/depthRelativeToSurface/min a :DepthRelativeToSurface
                 oboe-core:hasQualifier :Minimum .
source           SELECT id from (values ('1')) s(id) ;        (3) Which data to map

]]                                                  How data are mapped (2)
```

how can we automate the generation of this kind of the mapping files

3 Important parts

*** DB Part**

(1) DB Access Informations

*** Target Part**

(2) How data are mapped

Rule : Graphs are composed of nodes, each non terminal node is identified by an URI.

Target Part = Graph +URIs

Based on **turtle** syntax

*** Source Part**

(3) Which data to map

Using SQL Queries

??

Understand how ONTOP works behind the scene.

AUTOMATION APPROACH

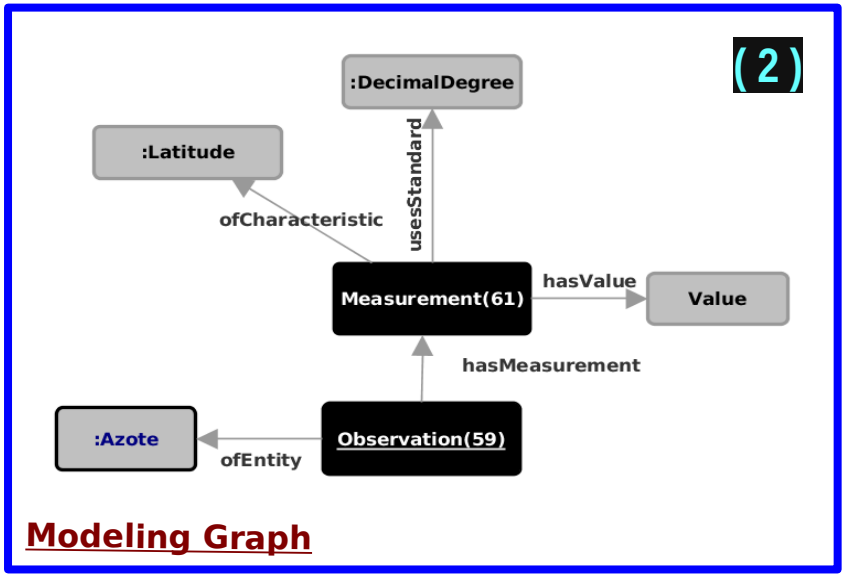
Level of Automation

Piece of Target Part

According to the modeling graphs presented earlier, and according to the rule : **Target Part = Graph +URI**

Modeling graphs can be considered as a piece of the target Part.

What is missing here is something which allows us to uniquely identify each node of the graph, these unique things are URI



Modeling Graph

In semantic approach, the simplest way to modelize is to use Graphs

Why not transform this Modeling Graphs to ODBA mapping file ???

AUTOMATION APPROACH

Level of Automation

Target Part

Connection information to DB (1)

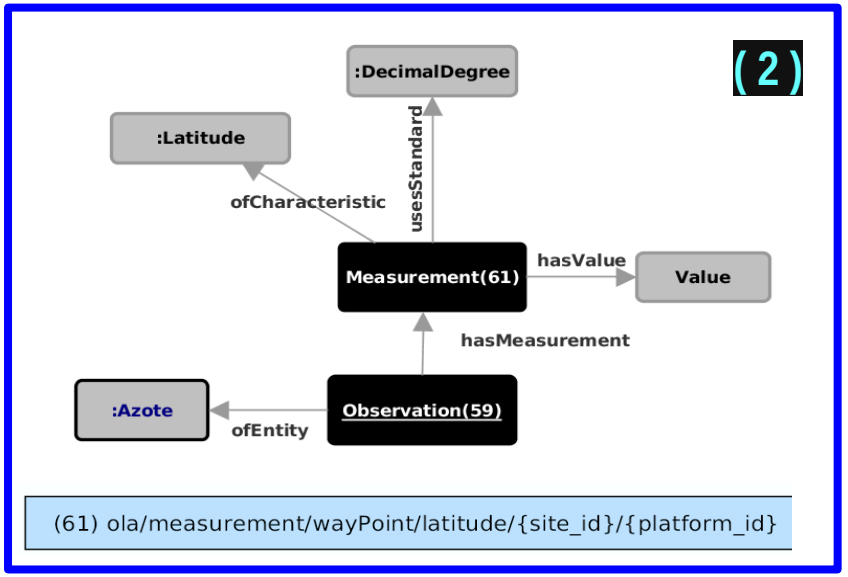
obda-sourceUri : dbLacsMapping

obda-connectionUrl : jdbc:postgresql://127.0.0.1/ola?sendBufferSize=5000

obda-username : ryahiaoui

obda-password : yahiaoui

obda-driverClass : org.postgresql.Driver



yedGen was specifically written to generate obda files from modeling graphs

Assign SQL Query for each non terminal node

Query_(61) : SELECT pla.loc_id AS platform_id, site.id AS site_id, pla.latitude AS latitude (3)
FROM
public.site_glacpe_sit site INNER JOIN public.plateforme_pla pla ON site.id = pla.id

Source Part

This is how was approached the Automation problem

AUTOMATION APPROACH

About Genericity ?

The idea is to generate multiple instances of the same Graph according to different variables described in file (CSV).

Why ? Because these variables has the same Structuration in the data Bases.

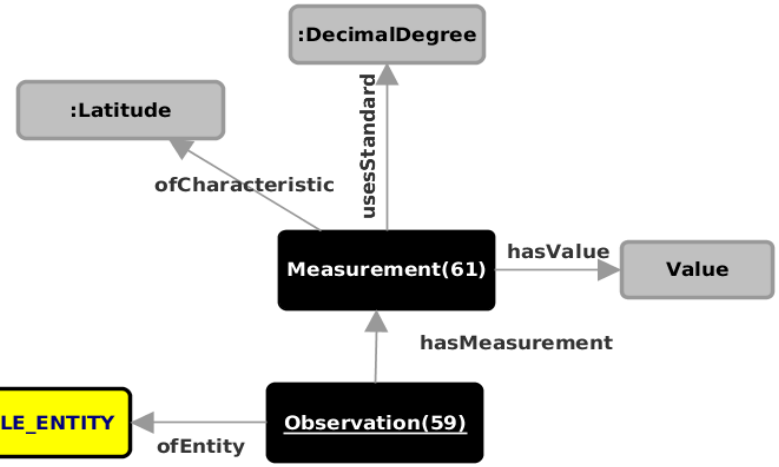
Instead to create one graph per variable, we use a graph type (graph designed for several variables) in order to create instances of this graph

csv file of variable semantic description

	AnaEE Standar	Entity	Context	..
1	cumulative rainfall	cumulative rain		..
2	air carbon dioxide	carbon dioxyde	atmosphere,	..
3	atmospheric air sta	air	atmosphere	..

Apply on VARIABLE_ENTITY each value of the column Enity from CSV , which gives us ...

Graph Type = one graph for several variables



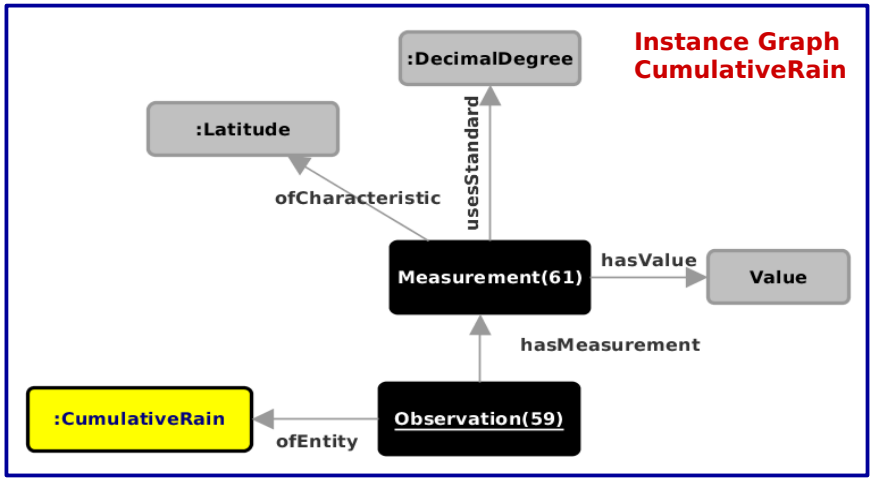
Genericity

The genericity we talk about here concerns the functioning of yedGen

About Genericity ?

csv file of variable semantic description

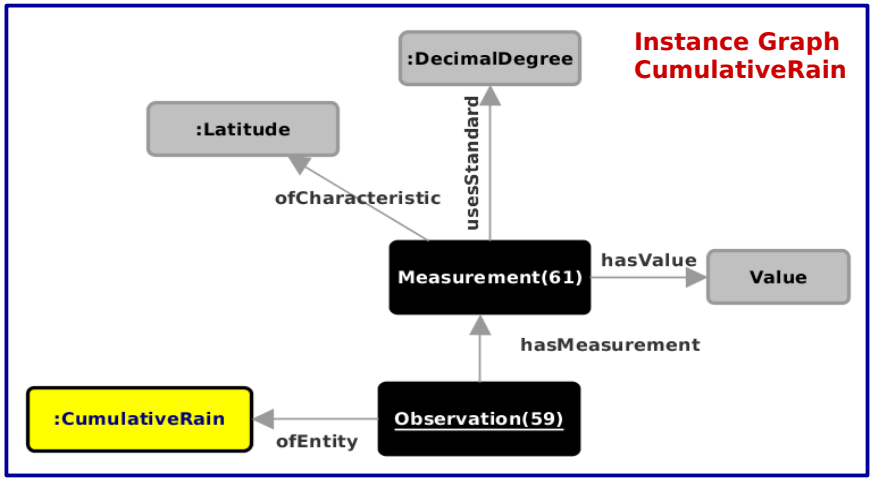
	AnaEE Standar	Entity	Context	..
1	cumulative rainfall	cumulative rain		..
2	air carbon dioxide	carbon dioxyde	atmosphere,	..
3	atmospheric air sta	air	atmosphere	..

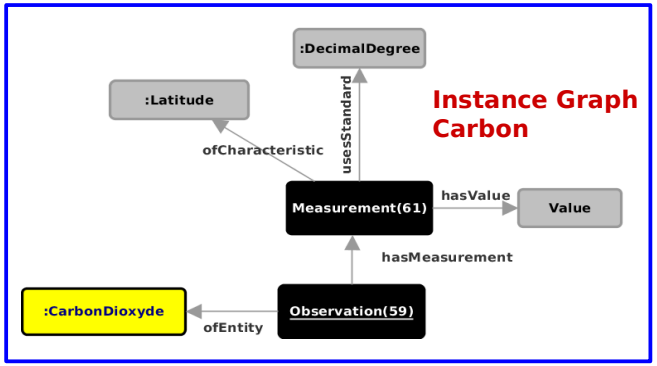


About Genericity ?

csv file of variable semantic description

	AnaEE Standar	Entity	Context	..
1	cumulative rainfall	cumulative rain		..
2	air carbon dioxide	carbon dioxyde	atmosphere,	..
3	atmospheric air sta	air	atmosphere	..

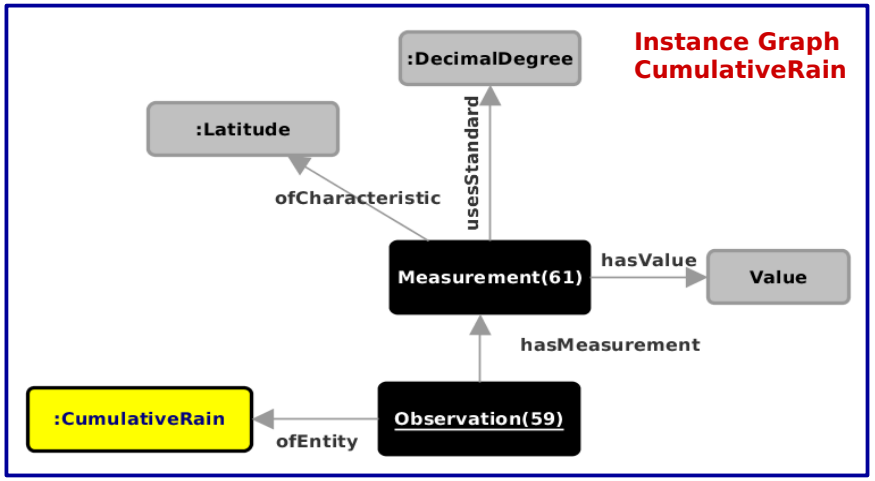


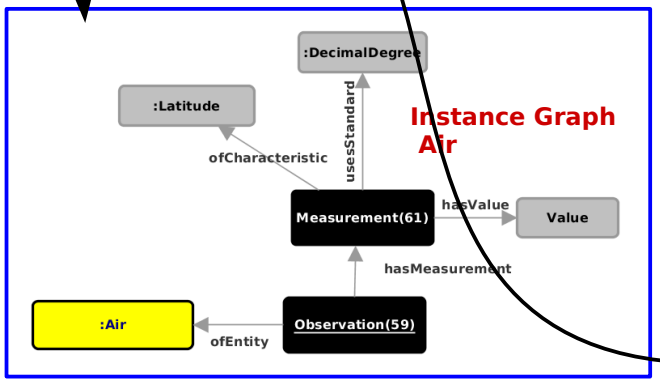


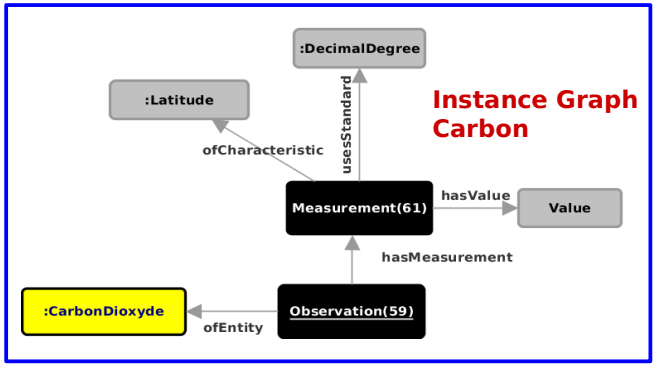
About Genericity ?

csv file of variable semantic description

	AnaEE Standar	Entity	Context	..
1	cumulative rainfall	cumulative rain		..
2	air carbon dioxide	carbon dioxyde	atmosphere,	..
3	atmospheric air sta	air	atmosphere	..







AUTOMATION APPROACH

About Genericity ?

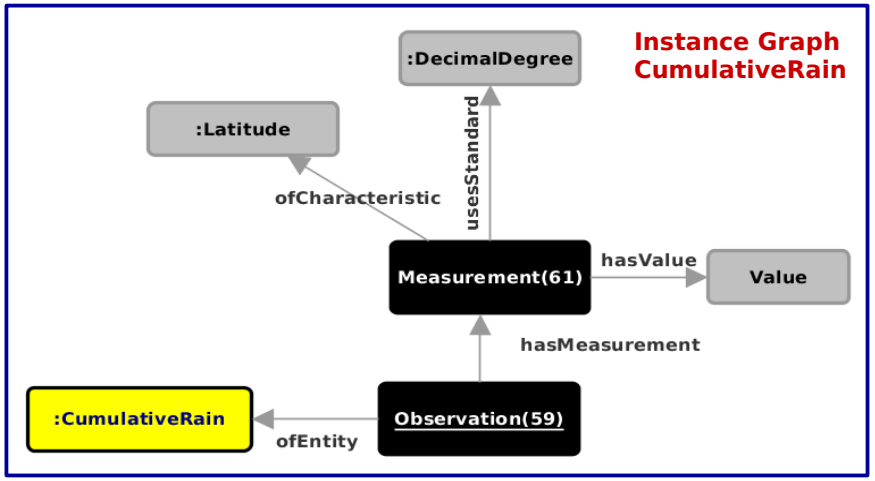
csv file of variable semantic description

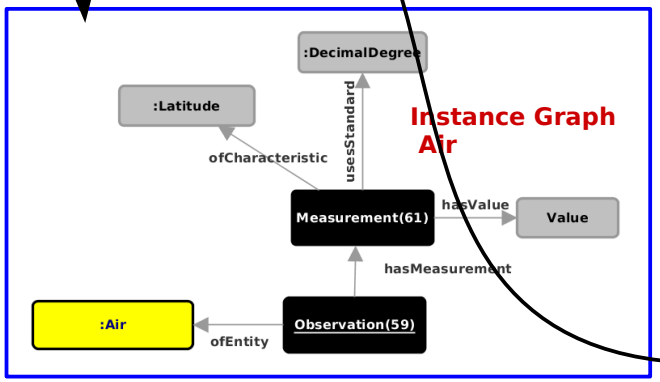
	AnaEE Standar	Entity	Context	..
1	cumulative rainfall	cumulative rain		..
2	air carbon dioxide	carbon dioxyde	atmosphere,	..
3	atmospheric air sta	air	atmosphere	..

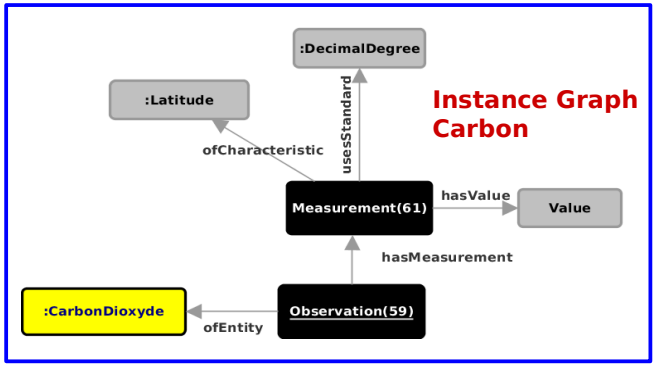
The same process is repeated for each line..

yedGen

This is how was approached the Genericity problem





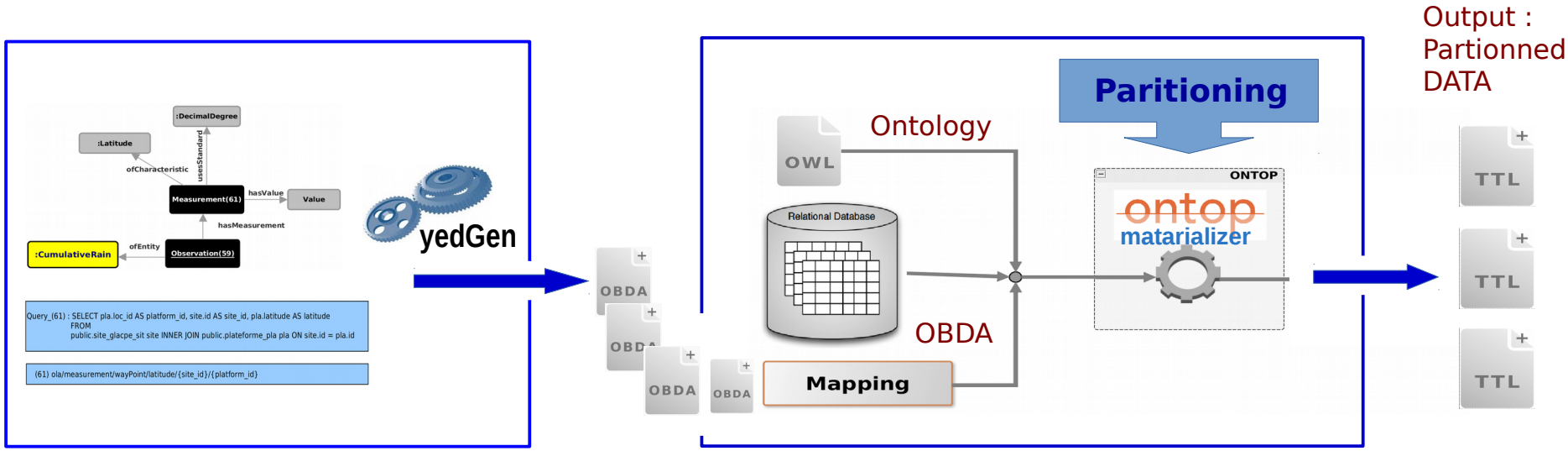


AUTOMATION APPROACH

About Perfomance (on large amounts of data) ?

It sometimes happens that the amount of data to be processed by ONTOP and BlazeGraph exceeds the memory capacity, in this case Outofmemoryerrors are raised.

Solution : **Volume data Partionning** → Processing data by chunk



→ Process « infinite » volume of data

AUTOMATION APPROACH

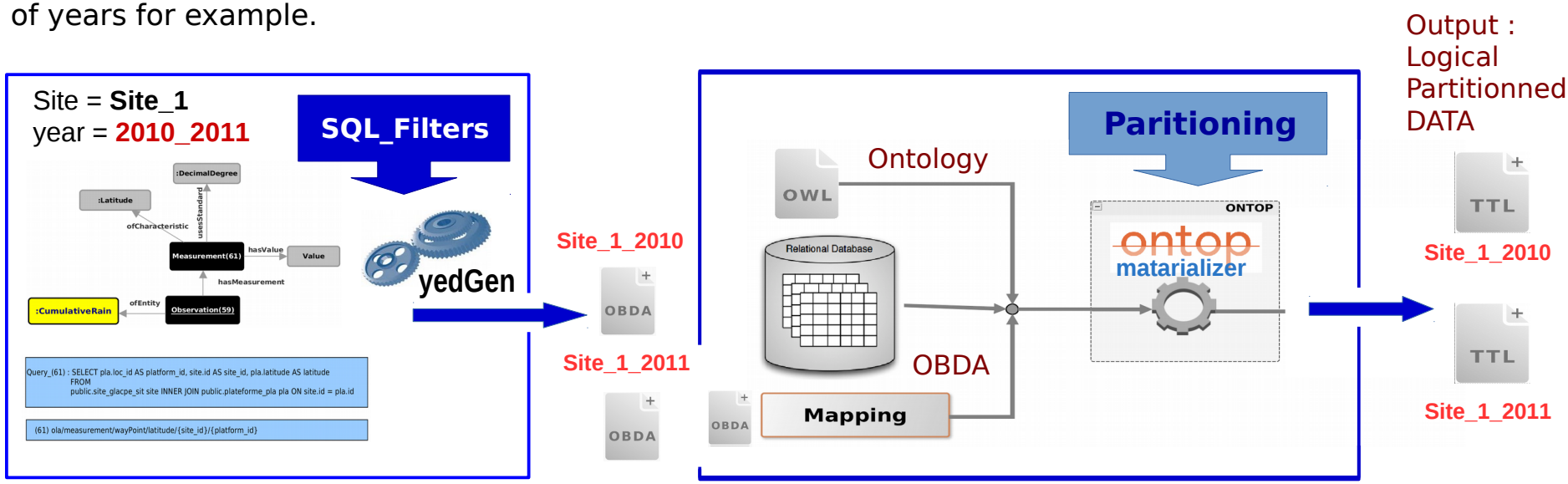
About Perfomance (Filtered Data)

For some use cases, we have to **extract only** the data that users **needs**

Solution : Logical data partionning

Generate Data for a specific variable with specific Site and specific range of years for example.

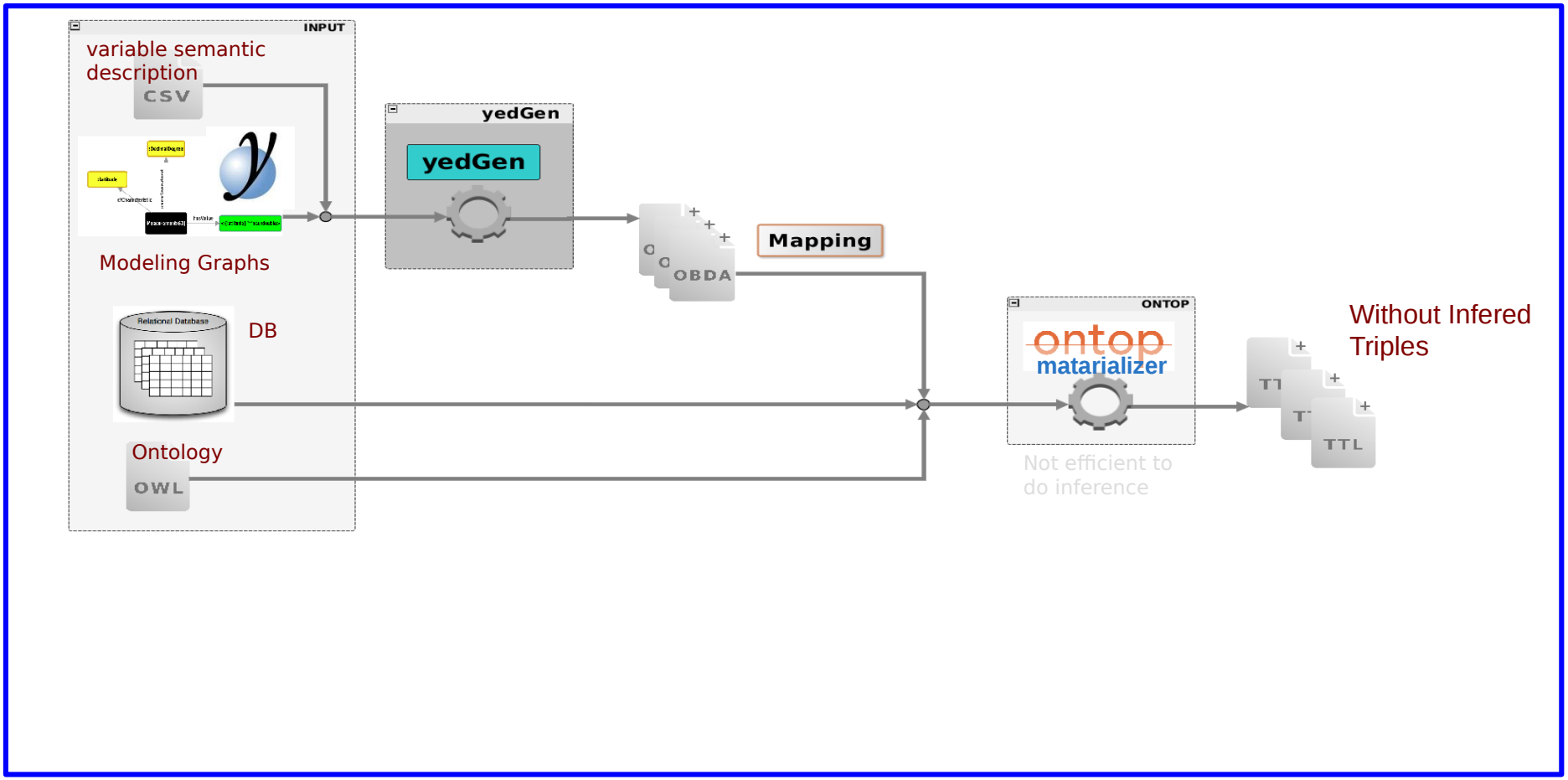
the more you filter, the less you process data, the more you're performant



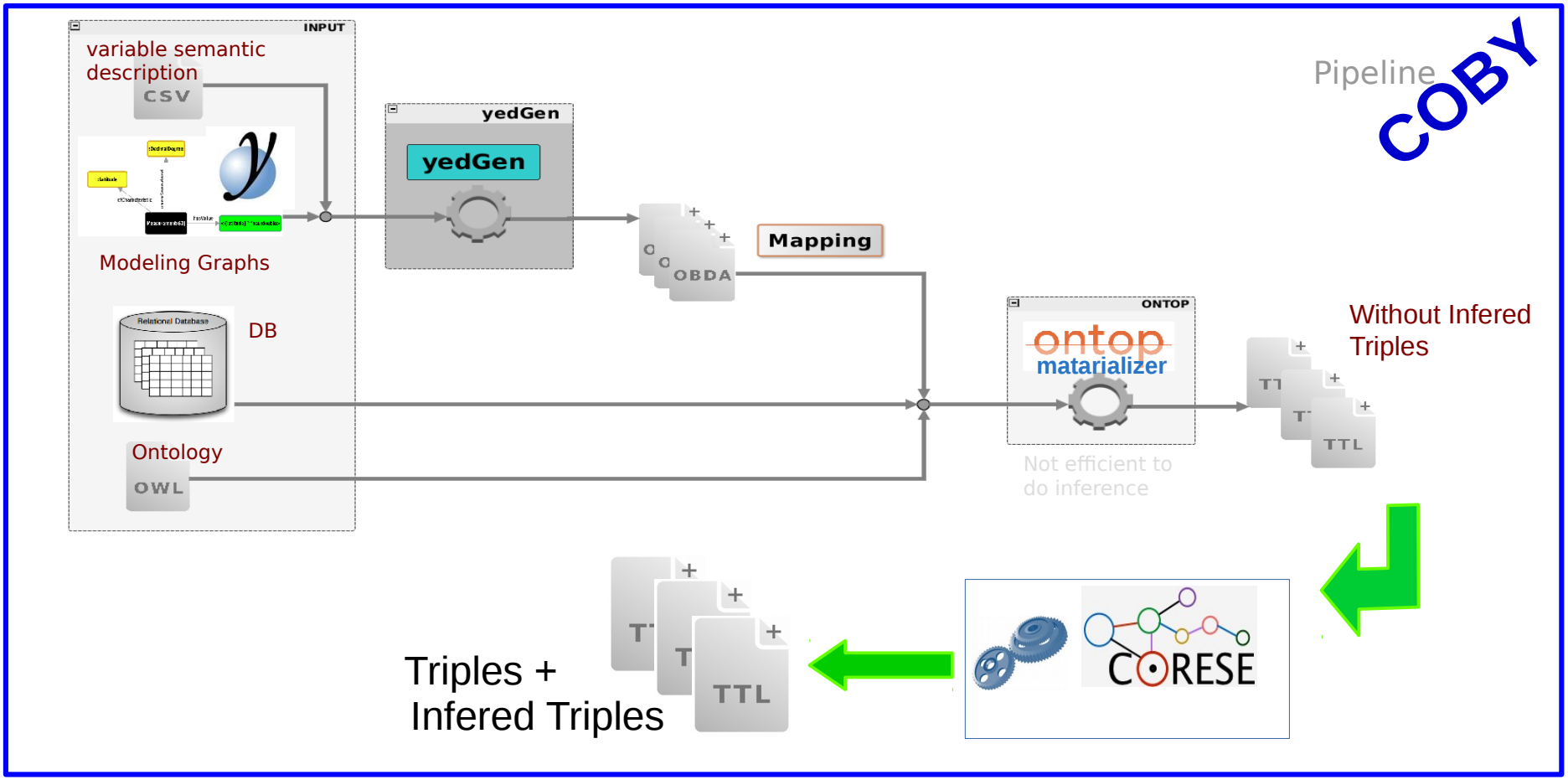
Volume data partitioning & Logical data partitioning can be **combined**

→ Process « infinite » volume of data

AUTOMATION APPROACH



AUTOMATION APPROACH

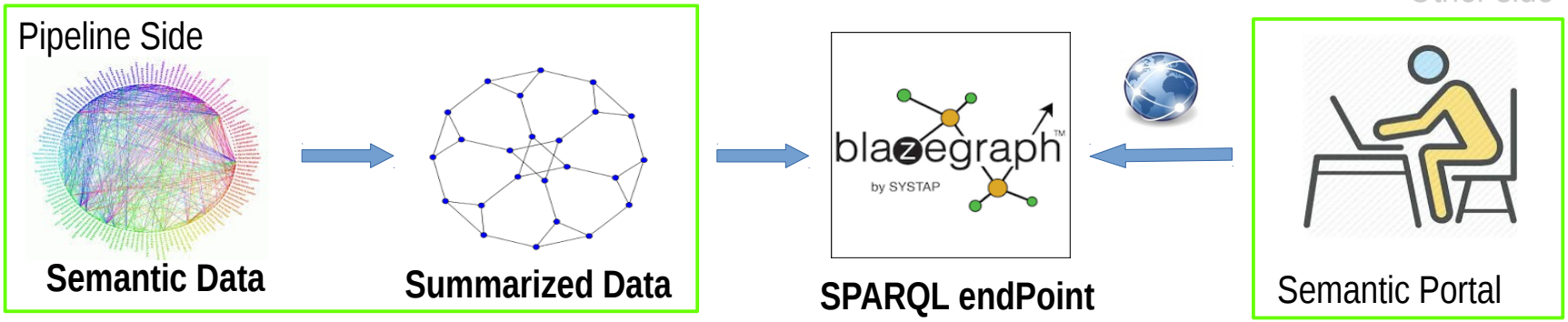


USE CASES

Concrete use cases of the Pipeline ?

1 - Semantic Summarized Data for Portal Use Case 1

Other side



Purpose : Produce a semantic summarized data using the Pipeline and publish it on a specific SPARQL Endpoint (blazegraph) which be consumed by an external entity (AnaEE-F Portal)

2 - Production of netCDF file Use Case 2

Purpose : Produce **Filtered** Semantic data (in TTL Format) that will be used to produce **netCDF** Files

Session for each of these use cases

EVOLUTION

** SOERE Level :

Modeling new Data types

Consists on Creation of new annotation models for other variables stored in relationnal data bases with Yed Graph Editor

** Pipeline Level :

Improve performance by introducing distributed processing [With Docker for example]** We will have **One OBDA File (Mapping) per Docker Container ??)**

Docker : software container platform

Thank You!