



HAL
open science

CANDELA - D2.6 Semantic search v2

Nathalie Aussenac-Gilles, Catherine Comparot, Ba-Huy Tran, Cassia Trojahn dos Santos

► **To cite this version:**

Nathalie Aussenac-Gilles, Catherine Comparot, Ba-Huy Tran, Cassia Trojahn dos Santos. CANDELA - D2.6 Semantic search v2. [Contract] IRIT. 2019. hal-02976647

HAL Id: hal-02976647

<https://hal.science/hal-02976647v1>

Submitted on 23 Oct 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Copernicus Access Platform Intermediate Layers Small Scale Demonstrator

D2.6 Semantic search v2

Document Identification			
Status	Final	Due Date	31/10/2019
Version	2.2	Submission Date	02/06/2020

Related WP	WP2	Document Reference	D2.6
Related Deliverable(s)	D2.6	Dissemination Level (*)	PU
Lead Participant	IRIT-CNRS	Lead Author	Cassia Trojahn
Contributors	See Document Information table	Reviewers	Michelle Aubrun (TAS-FR), Mihai Datcu (DLR)

Keywords:

Knowledge representation, ontologies, semantic data integration, semantic search, image metadata

This document is issued within the frame and for the purpose of the CANDELA project. This project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement No. 776193. 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 document and its content are the property of the CANDELA Consortium. The content of all or parts of this document can be used and distributed provided that the CANDELA project and the document are properly referenced.

Each CANDELA Partner may use this document in conformity with the CANDELA Consortium Grant Agreement provisions.

(*) Dissemination level: **PU**: Public, fully open, e.g. web; **CO**: Confidential, restricted under conditions set out in Model Grant Agreement; **CI**: Classified, **Int** = Internal Working Document, information as referred to in Commission Decision 2001/844/EC.

Document name:	D2.6 Semantic Search v2				Page:	2 of 52	
Reference:	D2.6	Dissemination:	PU	Version:	2.2	Status:	Final

Document Information

List of Contributors	
Name	Partner
Nathalie Aussenac-Gilles	IRIT-CNRS
Catherine Comparot	IRIT-UT2J
Cassia Trojahn	IRIT-UT2J
Ba-Huy Tran	IRIT-CNRS

Document History			
Version	Date	Change editors	Changes
0.1	31/07/2019	Ba-Huy Tran (IRIT-CNRS)	Table of Contents
0.2	01/10/2019	Ba-Huy Tran (IRIT-CNRS)	Completion
0.3	04/10/2019	Nathalie Aussenac-Gilles (IRIT-CNRS)	Revision
0.4	05/10/2019	Ba-Huy Tran (IRIT-CNRS)	Detailed the chapter 3
0.5	27/10/2019	Ba-Huy Tran (IRIT-CNRS)	Final version
0.6	28/10/2019	Juan Alonso (ATOS ES)	Quality Assessment
0.7	30/10/2019	Nathalie Aussenac-Gilles	Additions to section 4
0.8	31/10/2019	Nathalie Aussenac-Gilles	Evolution of section 4 + references
0.9	31/10/2019	Juan Alonso (ATOS ES)	Quality Assessment
1.0	31/10/2019	Jose Lorenzo (ATOS ES)	Coordinator approval for submission
1.1	02/04/2020	Ba-Huy Tran (IRIT-CNRS)	Major revision according to officer requests
1.2	25/04/2020	Ba-Huy Tran (IRIT-CNRS)	Reorganization to focus on prototype new version
1.3	05/05/2020	Cassia Trojahn (IRIT-UT2J)	Proofreading and evolution
1.4	15/05/2020	Nathalie Aussenac-Gilles	Proofreading
2.0	19/05/2020	Ba-Huy Tran	Minor revision
2.1	24/05/2020	Michelle Aubrun (TAS FR)	Review
2.1.1	25/05/2020	Mihai Datcu (DLR)	Review
2.2	01/06/2020	Ba-Huy Tran	Revisions according to reviewers' requests

Document name:	D2.6 Semantic Search v2			Page:	3 of 52
Reference:	D2.6	Dissemination:	PU	Version:	2.2
				Status:	Final

Quality Control		
Role	Who (Partner short name)	Approval Date
Deliverable leader	Cassia Trojahn (IRIT-UT2J)	31/10/2019
Quality manager	Juan Alonso (ATOS ES)	31/10/2019
Project Coordinator	Jose Lorenzo (ATOS ES)	31/10/2019

Document name:	D2.6 Semantic Search v2			Page:	4 of 52		
Reference:	D2.6	Dissemination:	PU	Version:	2.2	Status:	Final

Table of Contents

Document Information.....	3
Table of Contents.....	5
List of Figures.....	7
List of Acronyms	8
Executive Summary	9
1. Introduction	10
1.1 Purpose of the document.....	10
1.2 Relation to other project work.....	10
1.3 Revised submission	11
1.4 Structure of the document.....	11
1.5 Glossary adopted in this document	11
2. Semantic search on heterogeneous data	13
2.1 Datasets selection	13
2.1.1 Sentinel-2 images metadata.....	14
2.1.2 Vegetation index	14
2.1.3 Changes detected by image analysis.....	15
2.1.4 Land cover	15
2.1.5 Geo-located units	16
2.1.6 Weather information.....	16
2.2 Ontology development.....	17
2.2.1 Reused vocabularies.....	17
2.2.2 Data integration models.....	18
2.3 Data integration	23
2.3.1 Data extraction	23
2.3.2 Data transformation.....	23
2.3.3 Load	24
2.4 Semantic search	25
3. Integration to the CANDELA platform.....	27
3.1 Encapsulation of the module	27
3.2 User interface	28
3.3 Demonstration	28
3.3.1 Triplification.....	28
3.3.2 Semantic search	29
3.4 Triplestore capacities	30
3.5 Evolutions of the tools.....	31
4. Interoperability with the other components.....	36
4.1 Complementarity between DLR, CreoDIAS and IRIT semantic approaches.....	36
4.2 Compatibility between building blocks	37

Document name:	D2.6 Semantic Search v2			Page:	5 of 52
Reference:	D2.6	Dissemination:	PU	Version:	2.2
				Status:	Final

6. Conclusion.....	38
References.....	39
Annexes	40
Annex A: Jupyter notebook	40
Annex B: Jupyter notebook - Hackathon	48
Annex C: Collaboration with the other projects.....	51

Document name:	D2.6 Semantic Search v2				Page:	6 of 52	
Reference:	D2.6	Dissemination:	PU	Version:	2.2	Status:	Final

List of Figures

<i>Figure 1: Representation of Sentinel image metadata: the eom model and linked vocabularies</i>	<i>19</i>
<i>Figure 2: Representation of Earth Observation Analysis activities: the eoam model and linked vocabularies.....</i>	<i>20</i>
<i>Figure 3: Representation of territorial data to be mapped to Sentinel images: the tom model and linked vocabularies</i>	<i>22</i>
<i>Figure 4: Representation of weather data: the wom model and linked vocabularies</i>	<i>22</i>
<i>Figure 5: The semantic search web interface</i>	<i>26</i>
<i>Figure 6: Overview of the semantic search modules.....</i>	<i>27</i>
<i>Figure 7: Strabon triplestore performance</i>	<i>31</i>
<i>Figure 8: First version of the integration model representing computed data from Sentinel images ..</i>	<i>33</i>
<i>Figure 9: First version of the semantic search interface.....</i>	<i>35</i>

Document name:	D2.6 Semantic Search v2			Page:	7 of 52	
Reference:	D2.6	Dissemination:	PU	Version:	2.2	Status: Final

List of Acronyms

Abbreviation / acronym	Description
CSV	Comma-Separated Values
D2.6	Deliverable number 2.6 belonging to WP2
DIAS	Data and Information Access Service
EC	European Commission
EOM	Earth Observation Model
ESA	European Space Agency
ETL	Extract, Transform, Load
IRIT	Institut de Recherche en Informatique de Toulouse
LOD	Linked Open Data
NDVI	Normalized Difference Vegetation Index
NIR	Near-InfraRed
OGC	Open Geospatial Consortium
OWL	Web Ontology Language
PNG	Portable Network Graphics
RDF	Resource Description Framework
SOSA	Sensor, Observation, Sample, and Actuator
TAS, TAS FR, TAS IT	Thales Alenia Space, Thales Alenia Space France, Thales Alenia Space Italy
W3C	World Wide Web Consortium
WP	Work Package

Document name:	D2.6 Semantic Search v2			Page:	8 of 52
Reference:	D2.6	Dissemination:	PU	Version:	2.2
				Status:	Final

Executive Summary

This deliverable describes the second version of the semantic search module. As mentioned in the deliverable D2.5 [1], involves a set of tools helping to retrieve images through a semantic description of their content and contextual data. In [1], a prototype of the semantic search module was developed to allow searching for images and their vegetation index variation over time. However, such a version was not integrated to the CANDELA platform. After a complete revision of the proposal, in order to adapt the system on different use-cases and to integrate the additional modules to the platform, a number of changes have been made so far. This deliverable contains a complete description of the semantic search tools with the requirements in terms of datasets, vocabularies, and system architecture. In addition, the use of the module on a real use case is demonstrated. Finally, it details the changes introduced in the system in interaction with WP3 partners in order to properly integrate the tools on the CANDELA platform.

This deliverable has been produced at month 18 of the project in its first version, and strongly modified to produce a version 2.0 at month 24.

Document name:	D2.6 Semantic Search v2			Page:	9 of 52		
Reference:	D2.6	Dissemination:	PU	Version:	2.2	Status:	Final

1. Introduction

The advent of the Copernicus program and its wealth of open data (Earth observation images and their metadata) open many economic perspectives thanks to emerging applications in various fields. These applications can benefit from three types of data: image metadata (such as cloud cover of an image), results of an automatic analysis of the content of images (examples of analyses are calculating vegetation indices or learning changes on images) and geo-localized dated open data (government data, territorial units, meteorological data, etc.) that can be associated to the images themselves. We propose to use semantic web technologies and ontologies to support data integration: the ontology defines a unique vocabulary (made of classes and properties) for the homogeneous representation of meta-data and the heterogeneous data to be linked to images. The ontology and the resulting semantic data are represented using the RDF language.

In the context of the CANDELA project, we proposed to develop a semantic search tool on this semantic dataset to find out either data or Sentinel images according to the rich information that describe them, which means querying the semantic datasets using the types and properties defined in the ontology and the values available in the knowledge base. This task first required to design the ontology, which consists of a set of ontologies two of which are dedicated to space and time. Further tasks dealt with integrating the above-mentioned datasets, which required to develop a process that transforms these datasets into the semantic format. This process populates the ontology and, by doing so, builds a semantic base. The last task aimed at developing an interface that helps to query the RDF base and display the results. The whole process can also be run step by step thanks to Jupyter Notebooks that provide guidelines to carry out the approach.

1.1 Purpose of the document

The first objective of this document is to describe the changes brought to the semantic search module since the first deliverable D2.5 [1]. Evolutions mainly concern the redefinition of the integration vocabularies and the semantic search system architecture, the development of such systems and the adaption of the tools on the CANDELA platform. The second objective of this document is to introduce the context and explain how to operate the semantic search module on the datasets of the vineyard use-case defined in WP1.

1.2 Relation to other project work

Within the CANDELA project, the semantic search module aims at providing semantic capabilities to search for satellite images based on various kinds of information, principally the image metadata, image analysis results and open data. The module is adapted on the use cases defined by WP1 partners and receives results from other WP2 partners. It is integrated in the CANDELA platform by WP3 partners and designed to be compatible with other modules.

Document name:	D2.6 Semantic Search v2			Page:	10 of 52		
Reference:	D2.6	Dissemination:	PU	Version:	2.2	Status:	Final

1.3 Revised submission

Because both the model and the system evolved since the last submission (version 1.0), in this new submission (version 2.0 or higher), all sections have been revised and updated to describe the best of our work.

1.4 Structure of the document

The remaining of this document is structured in 5 major chapters:

- **Chapter 2** presents and illustrates the semantic search module. It is the main chapter describing the selected datasets, presenting the vocabularies that help to integrate these data with image-related metadata and the system components implemented for this integration, and the interface for semantic search.
- **Chapter 3** illustrates how to design a semantic search module for a specific use-case, the CANDELA vineyard use-case defined in WP1.
- **Chapter 4** presents the implementation of this tool on the CANDELA platform.
- **Chapter 5** presents the interoperability of IRIT tools with other components on the CANDELA platform.
- **Chapter 6** presents a preliminary study to initiate the collaboration with other projects, in particular the EOPEN EU project.

1.5 Glossary adopted in this document

Term	Definition
Concept or class	(formal) representation of a group of entities sharing the same properties
Dataset	Set of data, either in a data-base, a file or a triple store. Can be semantic or not
Entity	individual object or a value
(SPARQL) Endpoint	Web service that provides an interface to write SPARQL queries to search a knowledge base stored in a triple store
Instance	Representation of an entity. Instance of a class: means that the entity belongs to the class
Knowledge Base	Formal representation of a domain knowledge. In the semantic web, consists in an ontology (the schema), instances and rules.
Knowledge Graph	One of the ways to represent a knowledge base as a graph where nodes are classes and edges are properties.
Linked Open Data	Data represented in semantic format –RDF– and made available on the web
Module	Small vocabulary or ontology that has a semantic consistency, used to

Document name:	D2.6 Semantic Search v2			Page:	11 of 52
Reference:	D2.6	Dissemination:	PU	Version:	2.2
				Status:	Final

	represent a sub-domain
Ontology	Formal vocabulary (composed of classes and properties) enriched with domain specific rules and respecting good design principles
Open Data	Any database or file made freely available on the web (under licensing conditions)
Property	Labelled link between two resources (values, classes or entities)
Semantic Search	Search engine over semantic graphs
Triple Store	Repository where knowledge graphs can be stored + services for managing the repository
Vocabulary	Set of classes and properties required to represent a domain according specific needs or requirements

Document name:	D2.6 Semantic Search v2			Page:	12 of 52		
Reference:	D2.6	Dissemination:	PU	Version:	2.2	Status:	Final

2. Semantic search on heterogeneous data

By semantic search we mean services to retrieve images through a semantic description of their content (i.e. places, type of vegetation, NDVI), their location and date, or any semantic feature coming from open data and linked to the image context (weather parameters, names of locations, territorial units etc.). A (semantic) query specifies constraints on the data and their values. Results will be not only links towards images but also some data related to their content (i.e. type of vegetation, cities and their population, weather measures).

Semantic search relies on a formal representation of data that can be “located on” (or more generally “linked to”) images thanks to their date and localization. So, a preliminary work to the design of semantic search facilities for a specific use-case is to identify relevant data to be used to search for images, and then to propose a homogeneous representation of this heterogeneous data. We propose a process to design a semantic search module in four stages: dataset selection, ontology development, data integration and semantic search. In this chapter, we present a methodology to develop the semantic search module for core datasets that are Sentinel images. In the next chapter we will take a Candela use-case (the vineyard use-case) as an example to demonstrate how to apply the process on a specific use-case knowing that the same process can be applied on other ones.

2.1 Datasets selection

The first stage is to select or build datasets that are relevant for the considered use-case. The datasets are selected in keeping with the requirements and scenarios of the use-cases proposed by WP partners. Whatever the use-case, we distinguish three categories of datasets of interest:

- Sentinel image metadata: these metadata are available together with the Sentinel EO images on the DIAS.
- Sentinel-image related data: These datasets are extracted from Sentinel images thanks to image processing tools. These datasets are our main subjects of interest. They may be provided by partners of the project (like changes between an image and the previous one on the same area, computed using deep learning techniques; or land cover classification using data mining) or computed by other agencies (like Corine Land Cover).
- Auxiliary datasets: Coming from open data or linked open data (open data already available in semantic format), these datasets can supply contextual information for the use cases, such as weather information. They can also provide information about entities like (geofeatures) based on which the integration will be realized, for instance, cadastral or forest data.

All these datasets are heterogeneous by their content, their structure and their format. They may be open or private (and available only to the project members). The format may be databases, structured files (in JSON format or csv for instance), raster files for compact representation of images or other spatial data or graphs. The homogeneity of the representation can be obtained by using the same format for all datasets. We opted for RDF, the standard format proposed by the W3C for knowledge graphs, which is based on predicate/property/object triples. Nevertheless, turning all data into RDF triples (we call this process triplification) ensures a purely syntactic homogeneity of the representation, but it doesn't guarantee the quality of the integration. Moreover, turning all data

Document name:	D2.6 Semantic Search v2			Page:	13 of 52		
Reference:	D2.6	Dissemination:	PU	Version:	2.2	Status:	Final

from large datasets into graphs could be costly and not relevant. Homogeneity is also necessary at a semantic level, which requires to define and use a single and unifying vocabulary or better an ontology. A vocabulary defines classes (also called concepts) of entities, and their relations (called properties). An ontology includes a vocabulary and domain specific inference rules. We present the core datasets and how to build a knowledge graph out of them using ontologies in the next sections.

Another key feature of the way datasets is used are the pre-processing performed on the data, like averaging the data, checking or turning numeric data into qualitative values (or the reverse) or on the entire dataset, like sampling the data once a day, selecting only a part of the properties because not all properties are relevant, etc...

2.1.1 Sentinel-2 images metadata

Sentinel-2 is an Earth Observation mission from the EU Copernicus Program that systematically acquires satellite imagery at high spatial resolution. The Sentinel data products are available in the Copernicus Data and Information Access Service (DIAS) cloud environments. Each DIAS provides processing resources, tools and complimentary data sources at commercial conditions to further facilitate the access to Sentinel-2 data. In the project, we distinguish two categories of data: Sentinel-2 image metadata and Sentinel tile data.

Among the numerous DIAS that exist, such as Onda¹, Mundi Web Services², Sobloo³, we collect metadata of Sentinel-2 images from the DIAS of our partner Creodias⁴. The image metadata are retrieved from the EO Finder platform⁵. All queries may be executed as simple HTTP-Get calls, by typing the query in a web browser address line, by using any HTTP client, e.g. curl or wget, or from inside of users' program.

The platform returns JSON files storing images metadata records. Each metadata record contains the raster image file name, the cloud cover, the capture time and the corresponding tile. We can specify a large number of parameters to get a set of images, i.e. a maximum value for the cloud cover, an interval of time, an area of interest, the processing method, a used instrument, etc.

2.1.2 Vegetation index

The NDVI vegetation index (Normalized Difference Vegetation Index) is computed using Sentinel-2 images. This index is obtained by processing near-infrared (NIR) and red (R) sensors, which can detect the chlorophyll level. The output of this process gives a matrix of values between -1 and 1 characterizing the NDVI of each pixel of the image. The values between -1 and 0 represent the elements composed of water, values between 0 and 0.25 represent the compound earth elements. In order to associate an index with each pixel of an image, we have classified the indices into four levels whose limits are automatically calculated:

- VeryLow Vegetation Index
- Low Vegetation Index
- Middle Vegetation Index
- High Vegetation Index

¹ <https://www.onda-dias.eu/>

² <https://mundiwebservices.com/>

³ <https://sobloo.eu/>

⁴ <https://creodias.eu/>

⁵ <http://finder.creodias.eu>

Document name:	D2.6 Semantic Search v2			Page:	14 of 52		
Reference:	D2.6	Dissemination:	PU	Version:	2.2	Status:	Final

To get a representation of these categories as percentage, the number of pixels in each category must be calculated in relation to the total number of pixels in the desired zone. The zone could be a cadastral parcel, a village, or an entire Sentinel tile. This information is collected in public data sources from governments that provide territorial units nomenclature and descriptions with their labels and geometry on Earth. The geometry generally comes as a polygon in vector format. The percentages are evaluated in the triplification process implemented in a NoteBook as presented in Annexe A.

2.1.3 Changes detected by image analysis

TASF and TASI have proposed to develop an application for change detection in order to identify various kinds of changes between two consecutive images available on the Creodias EO Finder platform. TASF tools apply deep learning algorithms on Sentinel-2 images while TASI tools apply “classical” machine learning algorithms on Sentinel-1 images. These tools output generic change detection maps for each pair of time-consecutive Sentinel images that represent exactly the same field of view (footprint). The output map is a GeoTIFF raster with the same size and geo-reference as the images on which it has been calculated. Pixel values of the resulting map represent the probability (between 0 and 1) that a change has occurred between the acquisitions of the two input images.

We have classified the output value into four categories whose limits are automatically calculated based on the min and max value of the map:

- VeryLow Change
- Low Change
- Middle Change
- High Change

Using the same computation method as for NDVI, the number of pixels in each category is calculated in relation to the total number of pixels in the desired zone to represent these categories as percentages. This process is part of the pipeline run when a change dataset is given as input.

2.1.4 Land cover

DLR implemented a system allowing the prototyping of EO applications by applying interactive Data Mining functions to satellite images as described in D2.1 [2] and D2.2 [3] and Data Fusion functions presented in D2.7 [4] and D2.8 [5]. These functions search for patterns or structures in sets of EO image products, and generate semantic annotations at image patch level. In other words, these functions make explicit the semantics of EO images. The methods are based on Active Learning. They support the discovery of objects or structures in images, and classify the land cover into one of the categories provided by the end user contributing to the active learning. A hierarchical semantic annotation scheme is defined with 3 levels but only the top two levels are used in case of Sentinel-2. The EO image semantic annotation results, relevant EO image metadata and patch features are stored in the Data Mining Data Base, a MonetDB database. Following the project requirements, DLR has implemented functions to export the annotation results into raster files so that the output be compatible with the input of the triplification tools that we propose. Pixel values of the raster represent the EO image annotation labels given by the users. During the triplification process, the semantic classes in the DLR annotation scheme will be aligned with other land cover classification classes.

Document name:	D2.6 Semantic Search v2			Page:	15 of 52
Reference:	D2.6	Dissemination:	PU	Version:	2.2
				Status:	Final

In addition to DLR semantic data, several open datasets provide land cover information for different spatial extent and in different detail levels. They could be used depending on the use case in general and on the size of territorial units of interest. We considered in particular the following three datasets:

- Global Land Cover Share dataset⁶: The worldwide dataset was published by FAO in 2014, it is
- provided in raster format as a TIFF file. The value of each pixel is an integer that represents the identifier of the most prevalent land cover class for the area that is covered by the pixel.
- CESBIO dataset⁷ is another potential land cover source. The dataset is updated more frequently but is only available for France. It is provided in both raster and vector format as a TIFF file.
- Corine Land cover dataset⁸ is rather the most standard one using the Corine Land cover vocabulary. The most recent version (2018) contains land cover information for 39 European countries.

2.1.5 Geo-located units

Geo-located units are used as features of interest with spatial information. Their footprint is one of the key input parameters to compute values from the information available in image raster files. Geo-located units define territorial areas of interest for the end user. Their geometry is used in the triplification process to compute index values on larger areas than pixels. So the data source is identified and selected by the user depending on his use case. For example, the French administrative units and cadastral dataset can be used for the vineyard use case because they provide the geometry of each cultivated land parcel, in particular vineyards. Both of these datasets are available on a French government open data website⁹. The forest dataset from Forest Data Bank¹⁰ published by a Polish agency about Forest Management identifies all the forests in Poland. We proposed to use it for the forest use case.

2.1.6 Weather information

Weather data act as contextual information that can explain some change and detail the situation or phenomenon; for instance, the temperature and weather during the period when some parcels highly changed could contribute to explain this change. Historical weather dataset can provide such information. For France, it is possible to retrieve precise weather data from MeteoFrance website¹¹; however for Europe, precise information is not easily accessible due to the heterogeneity of the sources from country to country.

International historical weather data can be retrieved from ECAD website (European Climate Assessment & Dataset project¹²). The ECA dataset contains a series of daily observations captured at 19090 meteorological stations throughout Europe and the Mediterranean area. The predefined

⁶ <http://www.fao.org/geonetwork/srv/en/main.home?uuid=ba4526fd-cdbf-4028-a1bd-5a559c4bff38>

⁷ <http://osr-cesbio.ups-tlse.fr/~oso/>

⁸ <https://www.data.gouv.fr/en/datasets/corine-land-cover-occupation-des-sols-en-france>

⁹ <https://www.data.gouv.fr/en/>

¹⁰ <https://www.bdl.lasy.gov.pl/portal/en>

¹¹ <http://www.meteofrance.com/>

¹² <https://www.ecad.eu/>

Document name:	D2.6 Semantic Search v2			Page:	16 of 52
Reference:	D2.6	Dissemination:	PU	Version:	2.2
				Status:	Final

subsets of ECA data are created on a regular basis for easy bulk download. These are compressed files; each of them contains separate text files storing observations of a particular weather measure of a station.

2.2 Ontology development

The second subtask to be carried out is to design an appropriate ontology that will provide the right concepts and properties to represent the data and their relations. Once this ontology is available, the third stage is to build the semantic representation of a dataset, that we will call a knowledge base or knowledge repository. This task requires to assign one of the classes of the vocabulary to each data, and to store the relations between the data thanks to properties. Two types of properties are of major importance in the case of Earth Observation data and information: the geometry or location of the observations, and their date of capture.

To integrate the selected heterogeneous datasets, we propose to rely on semantic data integration techniques. Semantic data integration is the process of combining data from heterogeneous sources and consolidating them into meaningful and valuable information through the use of semantic technology. In this way, heterogeneous data is expressed, stored and accessed in the same way. In addition, auxiliary data can be “located on” (or more generally “linked to”) images thanks to their temporal and spatial features. For instance, meteorological measures (humidity, temperature, pressure) are dated and geo-localized as they are attached to meteorological stations or to villages. It is possible to know which part of an image is documented by this data through the topological relation between the village and the image footprint (or tile) and the temporal relation between the meteorological measure sample and the image capture date.

2.2.1 Reused vocabularies

To make the integration ontology compliant with existing standards, we have reused five vocabularies: GeoSPARQL, OWL-Time, SOSA, GeoDCAT (version 2), PROV-O.

1. **GeoSPARQL**, an OGC standard, defines an ontology for the representation of features, spatial relations and functions. While alternative vocabularies exist, such as GeorDF which allows for representing simple data like latitude, longitude, and altitude as properties of points (using WGS84 as reference datum), and GeoOWL which allows for expressing spatial objects (lines, rectangles, polygons), we opted for GeoSPARQL because it offers good reasoning capabilities to compare geometries and then establish relations (such as *contains*, *touches*, *overlaps*, etc.) between them. The *geo:Feature* class represents any entity having a spatial component. This spatial component is described as a *geometry* (point, polygon, etc.), instance of the *geo:Geometry* class, and related to its feature via the property *geo:hasGeometry*.
2. **OWL-Time**¹³ is a W3C standard ontology for representing the temporal component of data. The *time:TemporalEntity* class represents any entity having a temporal dimension, i.e. a start date (*time:hasBeginning* property) and an end date (*time:hasEnd* property), and thus a duration (*time:hasDuration* property). Temporal entities can be linked with binary relations (such as *meets*, *overlaps*, *during*) coming from the Allen's interval algebra. They are used for spatio-temporal reasoning. OWL-Time provides a generic property called *time:hasTime* which may be used to associate a temporal entity to anything (such as a calculated NDVI).

¹³ <https://www.w3.org/TR/owl-time/>

Document name:	D2.6 Semantic Search v2			Page:	17 of 52
Reference:	D2.6	Dissemination:	PU	Version:	2.2
				Status:	Final

3. **SOSA**¹⁴ (Sensor, Observation, Sample, and Actuator) is a light-weight but self-contained core ontology representing elementary classes and properties of the ontology SSN¹⁵ (Semantic Sensor Network). SOSA describes sensors, their observations and their procedures. It has been largely adopted in a range of applications, and more recently, satellite imagery. In the SOSA vocabulary, an observation (*sosa:Observation*) is considered as a sensor activity providing an estimation of a property value using a given procedure. It allows us to describe the related features of interest and the observed properties as well. SOSA reuses OWL-Time to date observations (*sosa:phenomenonTime*). We have hence adopted SOSA for describing image metadata and meteorological observations as respectively, *Earth observations* (*eom* module) and *meteorological observations* (*wfo* module). We chose to specialize SOSA in order to better type the instances of these concepts, although the trend in domains largely adopting SOSA, such as IoT, is to avoid this kind of construction and to directly use SOSA as main vocabulary. Thanks to the specialization, we identify better the various kinds of data described with the ontology (e.g., image metadata, meteorological observations, for instance). We reused the version 2 of SOSA. SOSA is considered as an updated OWL implementation that covers most of Observations and Measurements (O&M) vocabulary.
 - **DCAT**¹⁶ (Data Catalog Vocabulary) enables to describe datasets and data services in a catalog using a standard model and vocabulary that facilitates the consumption and aggregation of metadata from multiple catalogs. In DCAT, a catalog (*dcat:Catalog*) is a dataset in which each individual item is a metadata record describing some resource; the scope of the catalog is collections of metadata about datasets or data services. A dataset (*dcat:Dataset*) is a collection of data, published or curated by a single agent. The second version of DCAT has been recently published taking in account various properties introduced in GeoDCAT-AP, an application of DCAT version 1 applied in Europe. We have applied the same restriction proposed in DCAT-AP (for DCAT version 1) on this version of DCAT.
 - **PROV-O**¹⁷ (Provenance ontology) defines a data model, serializations, and definitions to support the interchange of provenance information on the Web. In PROV-O vocabulary, an entity (*prov-o:Entity*) captures a thing in the world in a particular state. The entity was derived from some other entity, and was generated by an activity (*prov-o:Activity*) that used other entities. In CANDELA semantic search, the use of PROV-O vocabulary has two purposes:
 - o It is firstly used to keep track of the EO analysis activities of the partners that produced. EO analyses are *prov-o:Activity* that use Sentinel products, each of them being a *prov-o:Entity*, and generate raster file output, which is also a *prov-o:Entity*.
 - o Next, it is dedicated to document of the semantic indexation process (triplification). The process consumes a raster and a vector file, each of them is a *prov-o:Entity*, and generates observations. The latter represent calculated values, such as % of NDVI, change level or land cover, on each geo-features of the vector.

2.2.2 Data integration models

The information contained in the semantic dataset used for semantic search is represented using the classes defined in the ontology playing the role of an integration model. We built this ontology by

¹⁴ <https://www.w3.org/TR/vocab-ssn/>

¹⁵ <https://www.w3.org/TR/vocab-ssn/>

¹⁶ <https://inspire.ec.europa.eu/good-practice/geodcat-ap>

¹⁷ <https://www.w3.org/TR/prov-o/>

Document name:	D2.6 Semantic Search v2			Page:	18 of 52
Reference:	D2.6	Dissemination:	PU	Version:	2.2
				Status:	Final

reusing and extending the ontologies presented in section 2.2.1. This ontology is quite complex and composed of various modules, which are not all relevant for each dataset. Only a part of them are used for each of the datasets to be transformed into RDF triples. In the following, instead of presenting the ontology as a whole, we chose to present views on the ontology, that we call models, made of the classes and properties dedicated to describe each of the input dataset. If a new dataset would be used to add contextual information to Earth Observation images, then one of these models could be reused and adapted to get the right model for this new dataset.

Sentinel images metadata

The *eom* (Earth Observation model) model in Fig. 1 is the part of the ontology dedicated to represents metadata of Sentinel images. It is based on classes and properties defined in the SOSA, GeoSPARQL and PROV-O vocabularies. It mainly describes the *product* (an image) as a *result* of an *observation*. Each observation is associated with a temporal information -its date of capture- (*sosa:resultTime*) and a spatial information -the area being captured- (either by the *geometry* property of the *eom:Product* or through the *sosa:hasFeatureOfInterest* relation).

To reduce the cost of image indexing, we assume that images are mapped to a grid made of tiles, such as the grid provided by ESA for Sentinel-2 Single-Title (S2ST) images. So the model proposes a vocabulary to represent *grids* and *tiles*, which are spatial entities. So the *Tile* class specializes the *geo:Feature* class. As mentioned earlier, they could be dated thanks to the *sosa:resultTime* property if needed (if a new grid was defined by ESA). The model has been revised to take into account both Sentinel-1 and Sentinel 2 images metadata.

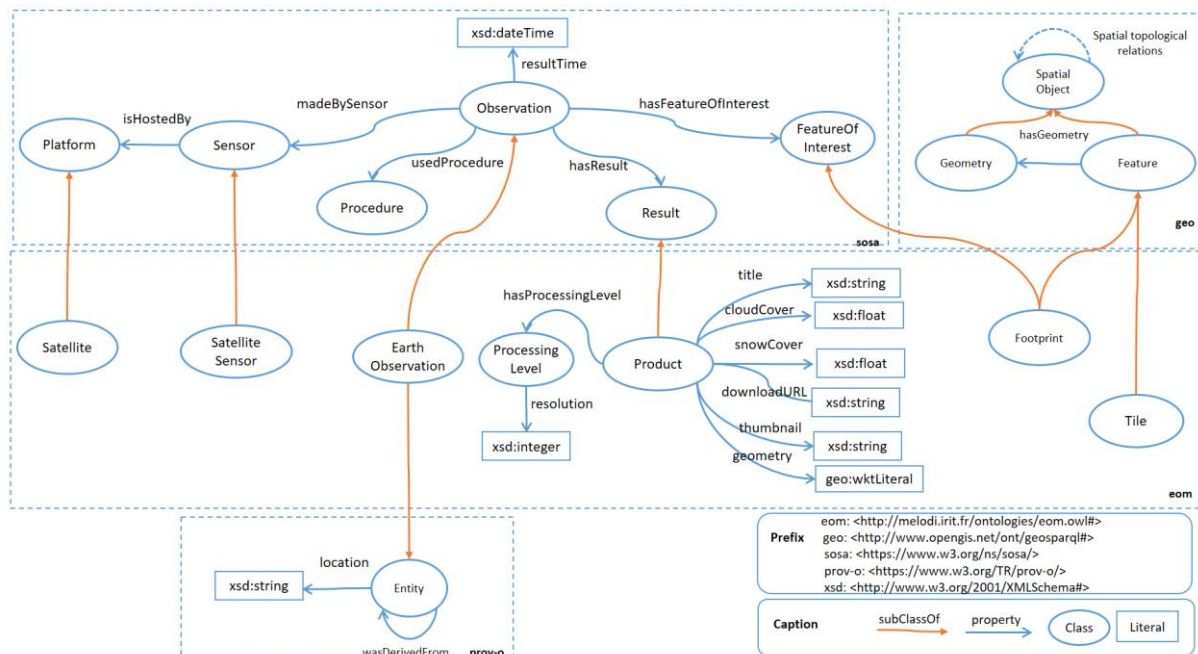


Figure 1: Representation of Sentinel image metadata: the eom model and linked vocabularies

Document name:	D2.6 Semantic Search v2			Page:	19 of 52
Reference:	D2.6	Dissemination:	PU	Version:	2.2
				Status:	Final

EO analysis activities model

Sentinel images are consumed by analyses defined by the project partners. For instance, they are the input data of various machine learning algorithms that identify changes between two images. The ontology provides a vocabulary, the *eoam* model (EO Analysis Model), to describe these datasets and to provide information about the way they were generated (Figure 2). The *eoam* model reuses the DCAT and PROV-O vocabularies, with links to *eom* and *tom* model of our ontology. The *EOAnalysis* class represents an analysis activity that takes as input one or several Sentinel images (*eom:Product*, as defined in Fig. 1), and is carried out by one of the project partners (*wasAssociatedWith* an *Agent*). The activity generates a *dataset* whose *distribution* is a *RasterFile*. Some of these classes come from the DCAT vocabulary dedicated to represent datasets information. We use DCAT both for the dataset generated by the analyses and for the dataset of the territorial units of interest (coming from open data) that provides Geo-Feature vector files. The model is represented by Figure 2.

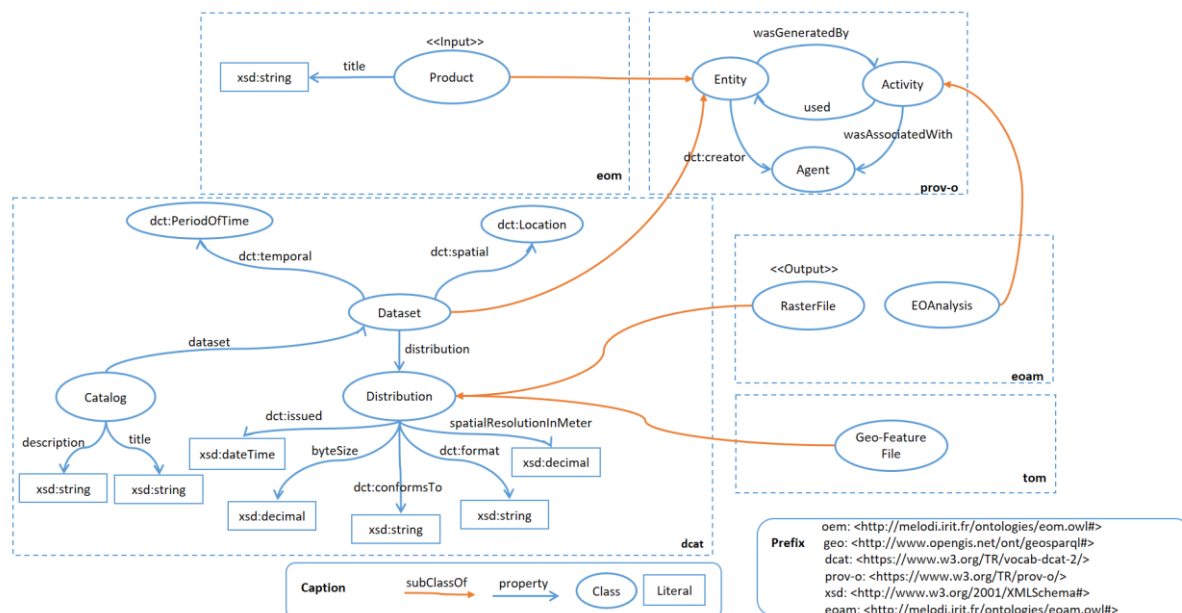


Figure 2: Representation of Earth Observation Analysis activities: the eoam model and linked vocabularies

Territorial observation model

The *tom* model (for territorial observation model) (Figure 3) acts as the main integration model. We generalized it recently to take as input any output from EO analysis activity as long as it comes in raster files. The *tom* model reuses SOSA, GeoSPARQL, OWL-Time as well as DCAT and PROV-O vocabularies. The key role of this model can be explained as follows: given a property of interest evaluated on each pixel of a raster file, the user wants to know the value of this property on each territorial unit presenting a footprint on Earth.

Document name:	D2.6 Semantic Search v2			Page:	20 of 52		
Reference:	D2.6	Dissemination:	PU	Version:	2.2	Status:	Final

The property could be for instance the land cover (Corine, Cesbio or DLR), detected changes (from TASF or TASI modules) or NDVI. It is important to have in mind that a human would give qualitative values to these properties, that we call indicators. Examples of land cover indicators are Wheat, MixedForest, ConiferousForest, etc.; examples of change level indicators are LowChange, MiddleChange. Each land cover and more generally has its own list or hierarchy of indicators. But image analysis is not able to provide a unique and certain value to this property. Moreover, on larger areas like a French village, various land covers may be present with various proportions. So in the source datasets, for each pixel of an image, the property has not a unique value, but rather a probabilistic estimation of each of its possible values. For instance, the land cover on a pixel could be wheat by 80%, forest by 2%, oat by 10% etc...

This model has evolved and the current definition is able to manage any property with its own set of indicators. The triplification module is able to process any “property” raster providing probabilistic values for each indicator on each pixel.

Examples of territorial units of interest are administrative units (village, county, ...), cadastral parcels, agricultural parcels or forest units. During the triplification process, a value (in percentage) is assigned to each indicator of the property for each territorial unit and represented in RDF format using the *tom* model.

In *tom*, we represent the input files of the triplification (raster file as a *eoam:RasterFile* and dataset of territorial units as a *GeoFeatureFile*) as well as its output: the RDF triples representing values assigned to each indicator on each territorial unit of the input dataset. The raster file stores the values of a particular property (land cover or change) on a given area (*tom:GeoFeature*). The same property is calculated for a territorial unit and stored in the RDF datastore. The property is represented using the *GeoFeatureObservationPropertyType* class; each indicator of the property is an instance of the *GeoFeatureObservationProperty* class. A *GeoFeatureObservationCollection* represents the evaluation of all the property indicators on a given territorial unit (represented with the class *tom:GeoFeature*) whereas *GeoFeatureObservation* describes the evaluation of a single indicator. The evaluation is represented using the *tom:GeoFeatureObservation* class. It produces (*sosa:hasSimpleResult*) the percent value of each indicator (*tom:GeoFeatureObservableProperty*) on a given territory (*tom:GeoFeature*), for instance, 40% of MixedForest and 60% of ConiferousForest. All the triples resulting from these “observations” form a *dcat:Dataset*. We keep track of how it has been generated using the PROV-O vocabulary: it is linked with *prov-o:wasDerivedFrom* to a RasterFile and a GeoFeature vector file.

Document name:	D2.6 Semantic Search v2			Page:	21 of 52
Reference:	D2.6	Dissemination:	PU	Version:	2.2
				Status:	Final

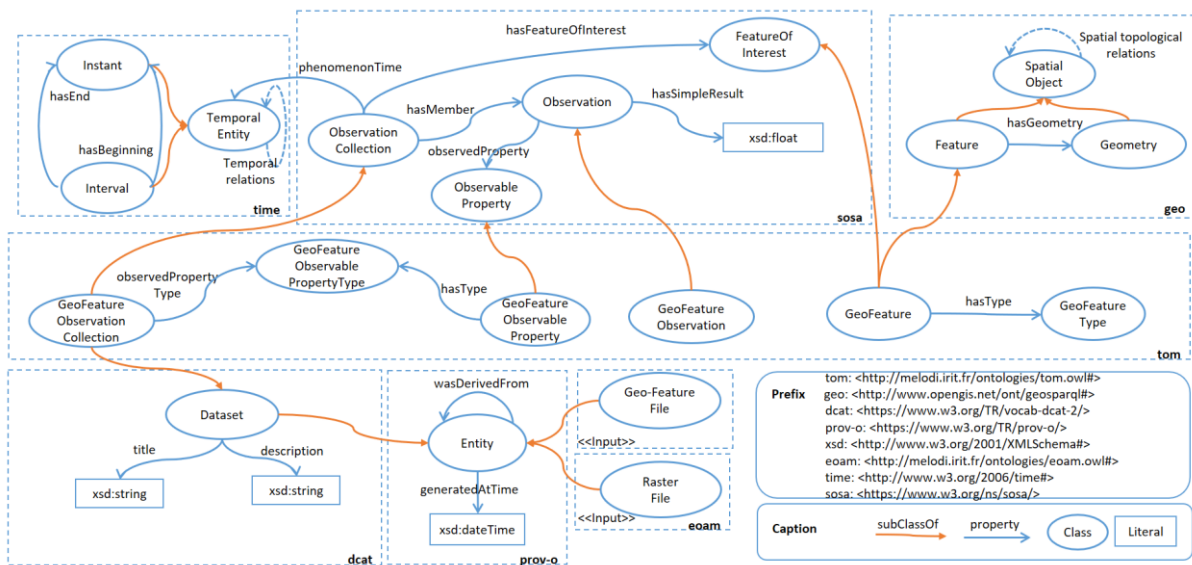


Figure 3: Representation of territorial data to be mapped to Sentinel images: the tom model and linked vocabularies

Weather observation model

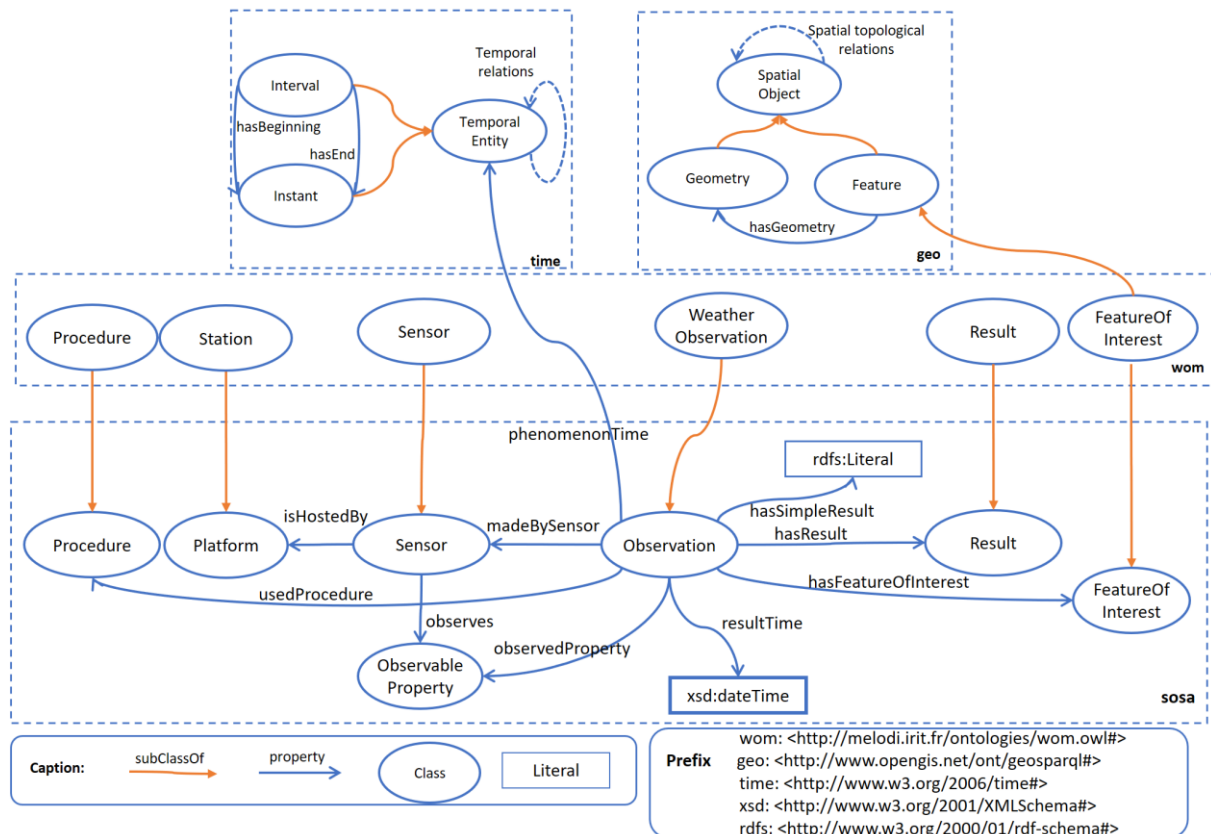


Figure 4: Representation of weather data: the wom model and linked vocabularies

Document name:	D2.6 Semantic Search v2	Page:	22 of 52
Reference:	D2.6	Dissemination:	PU
		Version:	2.2
		Status:	Final

Weather observations are described with the weather observation model (*wom*) which is based on SOSA, GeoSPARQL and OWL-Time vocabularies as shown in Figure 4. In the *wom* frame the *wom:Station* (weather station) class specializes *sosa:Platform* while *wom:FeatureOfInterest* specializes both *sosa:FeatureOfInterest* and *geo:Feature*.

2.3 Data integration

The third stage is to integrate the heterogeneous data from the various datasets by generating RDF triples based on the ontology developed in the previous stage. The triplification aims at building a semantic representation of selected data from the datasets, that we will call *a knowledge base* or *knowledge repository*. In the following sections, we describe each step of the process from raw data to semantic data, this process being known as ETL (extract, transform, load).

2.3.1 Data extraction

The first step of the data integration process is to identify and access the desired data from data sources (or datasets). When possible, we advise to avoid downloading the data from the datasets and storing useless information. For example, among all weather report data, only the weather forecast measures of a period are relevant to the use-case; or only the Sentinel image metadata concerning an area and a period of time is useful for the analysis, not the image itself. This step helps to limit the quantity of data to be processed and managed. It improves the whole system performance.

For Candela use-cases, auxiliary datasets are open data available in different formats: CSV, JSON, GeoJSON, GeoTIFF, Shape files... In the first version of semantic search, the datasets were extracted and loaded into MongoDB databases so that they could be retrieved in JSON format for further processing.

Several additional operations can be performed in this step to extract desired data in desired data type from raw values. These operations depend on the data source, the user needs and the integration model defined for the data source. They may be one of the big challenges of this stage. For example, hourly weather forecast measures can be aggregated to daily measures that are the average of all measures collected by the same sensor during one day. Spatial mask can be used to compute the NDVI or change detection (raster data) of a cadastral parcel or a village (vector data). Or specially, geo-localized data using different spatial reference systems must be converted to the same spatial reference system to be compared and to discover topological relations.

2.3.2 Data transformation

Extracted data is then transformed to RDF format. Given a dataset, once the relevant data has been calculated from the raw data (if needed), transforming the data to RDF format requires to select the appropriate model (among those defined in section 2) and then to assign one of the classes of the vocabulary to each data, and to store the relations between the data thanks to properties. Two types of properties are of major importance in the case of Earth Observation data and images: the geometry or location of the observations, and their date of capture. In this step, data from whatever

Document name:	D2.6 Semantic Search v2			Page:	23 of 52
Reference:	D2.6	Dissemination:	PU	Version:	2.2
				Status:	Final

sources are linked to images. This relation is based on the notion of ‘interest’ to select the relevant data, on a spatial dimension to link the data according to their location, and the date or temporal feature to link data according to their date. Given a set of data that satisfy a set of criteria of interest, thanks to the spatial and temporal links, one can retrieve the images that show the Earth at the time and location when and where the data has been collected or computed. It is important to note that temporal and spatial properties are stored, but temporal and spatial relations between entities from various sources are not necessarily stored in the knowledge base. They are evaluated at the time of querying the knowledge base.

We defined a mapping template and a processing mechanism implemented as a Python module. The latter contains functions helping to perform more sophisticated operations that are not possible in alternative approaches such as RML (RDF Mapping Language). Output of this step are files containing RDF triples (semantic data) in N-triples format.

The mapping template is hand written based on the developed ontology. We manually adapted it to define templates for each data source. They make explicit the mappings between the models presented in section 2.2 and the data source schemas. In other words, the template is an explicit writing of how entities of some category in the schema of the data source will be represented with the ontology classes and properties. Below is an extract of the mapping template used for Sentinel image metadata. It is necessary to make a template for each dataset with the same formalism.

```

@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>.
@prefix owl: <http://www.w3.org/2002/07/owl#>.
@prefix xsd: <http://www.w3.org/2001/XMLSchema#>.
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>.
@prefix sosa: <http://www.w3.org/ns/sosa/>.
@prefix eom: <http://melodi.irit.fr/ontologies/eom.owl#>.
instance a eom:EarthObservation.
instance sosa:resultTime valueToLiteral($.properties.startDate, dateTime).
instance sosa:hasResult instance_Result.
instance sosa:hasFeatureOfInterest valueToInstance($.properties.tile, Tile).
instance sosa:madeBySensor valueToInstance($.properties.sensor, Sensor/s).
instance _Result eom:fileName valueToLiteral($.properties.services.download.url, string).
instance _Result eom:fileSize valueToLiteral($.properties.services.download.size, integer).
instance _Result eom:type valueToLiteral($.properties.productType, string).
instance _Result eom:indentifier valueToLiteral($.properties.productIdentifier, string).
instance _Result eom:cloudCover valueToLiteral($.properties.cloudCover, decimal).
instance _Result eom:snowCover valueToLiteral($.properties.snowCover, decimal).
instance _Result eom:title valueToLiteral($.properties.title, string).

```

2.3.3 Load

The final step is to load the transformed data to a semantic data store, called triple-store. Strabon, a geospatial triple-store is chosen for the management of CANDELA semantic data because this platform can perform inferences on spatial relations among entities. Moreover, it provides an end-point from which data can be accessed through REST API and SPARQL, a protocol to query semantic data.

Document name:	D2.6 Semantic Search v2			Page:	24 of 52
Reference:	D2.6	Dissemination:	PU	Version:	2.2
				Status:	Final

2.4 Semantic search

The fourth stage is to design a search interface adapted to each use-case. The interface will take into account the kind of data that is part of a query. It should provide a relevant way to express the user's interest, the searched area and the time period.

Semantic data stored in Strabon can be accessed by several means:

- Strabon end-point web interface;
- SPARQL service: Users can make use of semantic tools and libraries to query the store, for instance Jena (Java framework), SPARQLWrapper or RDFLib (Python libraries);
- Strabon client java library.

In any case, it is necessary to formulate semantic queries. Unfortunately, these queries are not always straightforward for non-expert users, especially when they have no knowledge about the ontological models or SPARQL protocol. As a consequence, we provide to the user two possibilities to easily perform semantic search:

- Jupyter notebooks: Notebooks with prepared scripts and semantic queries have been developed and uploaded on the platform (see Annex A).
- Web interface: A web interface has been developed in order to simplify the query that takes into account three dimensions: spatial, temporal and data-specific.

The web interface

The interface of the semantic search module is composed of five zones as numbered in Figure 5:

1. The temporal dimension: Users must select a period of time.
2. The semantic (data-specific) dimension: Users can choose for example a level of change, a level of NDVI index or a land cover type.
3. The spatial dimension: User draws a zone of interest on the map.
4. Use cases area management: Users can add areas of new use cases. All areas are displayed on the map for identification.
5. The result of the semantic search is displayed to the right in tabular form.

The query results can be downloaded in two forms, either as a PNG image by right-clicking on the map or as a CSV file by clicking on the export button.

Document name:	D2.6 Semantic Search v2			Page:	25 of 52
Reference:	D2.6	Dissemination:	PU	Version:	2.2
				Status:	Final

When

From To

What

- Change_DL
- High_Change_DL
- Low_Change_DL
- Middle_Change_DL
- VeryLow_Change_DL
- + LC_CESBIO17
- + LC_Corine
- + LC_DM
- + LC_LU_FR
- + NDVI

Where

Clear Draw

Go

Help Zones

Parcel 471100000M0028

Weather (GOURDON - Distance:92502.59375m) Export CSV

Date	Measure	Value
2017-04-01T12:00:00	Humidity	8.9
2017-04-01T12:00:00	Max_Temperature	14.2 Cel
2017-04-01T12:00:00	Mean_Temperature	10.6 Cel
2017-04-01T12:00:00	Min_Temperature	7.1 Cel
2017-04-01T12:00:00	Precipitation	4.4 mm
2017-04-01T12:00:00	WindSpeed	1.8 m/s
2017-04-02T12:00:00	Humidity	8.2
2017-04-02T12:00:00	Max_Temperature	15.3 Cel
2017-04-02T12:00:00	Mean_Temperature	11.2 Cel
2017-04-02T12:00:00	Min_Temperature	7.2 Cel

Change_DL Export CSV

Property	Start date	End date	Value	Thumbnail	Sentinel image
VeryLow_Change_DL	2017-04-19T00:00:00	2017-04-29T00:00:00	0.0		http://melodi.int.fr/resource/Product/S2A_MSIL2A_20170429T105651_N1...
VeryLow_Change_DL	2017-04-19T00:00:00	2017-04-29T00:00:00	0.0		http://melodi.int.fr/resource/Product/S2A_MSIL2A_20170419T105621_N1...
Low_Change_DL	2017-04-19T00:00:00	2017-04-29T00:00:00	0.0		http://melodi.int.fr/resource/Product/S2A_MSIL2A_20170429T105651_N1...
Low_Change_DL	2017-04-19T00:00:00	2017-04-29T00:00:00	0.0		http://melodi.int.fr/resource/Product/S2A_MSIL2A_20170419T105621_N1...
Middle_Change_DL	2017-04-19T00:00:00	2017-04-29T00:00:00	0.0		http://melodi.int.fr/resource/Product/S2A_MSIL2A_20170429T105651_N1...

LC_CESBIO17 Export CSV

Property	Start date	End date	Value	Thumbnail	Sentinel image
culture ete	2017-01-01T00:00:00	2017-12-31T00:00:00	0.0		http://melodi.int.fr/resource/Product/CESBIO

LC_Corine Export CSV

Property	Start date	End date	Value	Thumbnail	Sentinel image
Non-irrigated arable land	2017-01-01T00:00:00	2019-12-31T00:00:00	0.0		http://melodi.int.fr/resource/Product/CLC
Pastures	2017-01-01T00:00:00	2019-12-31T00:00:00	0.0		http://melodi.int.fr/resource/Product/CLC

LC_DM Export CSV

Property	Start date	End date	Value	Thumbnail	Sentinel image
Mixed urban areas	2017-04-09T00:00:00	2017-04-09T00:00:00	0.0		http://melodi.int.fr/resource/Product/0
Mixed urban areas	2017-04-09T00:00:00	2017-04-09T00:00:00	0.0		http://melodi.int.fr/resource/Product/S2A_MSIL1C_20170409T105651_N0204_F...

LC_LU_FR Export CSV

Property	Start date	End date	Value	Thumbnail	Sentinel image
Tournefol	2017-01-01T00:00:00	2017-12-31T00:00:00	0.0		http://melodi.int.fr/resource/Product/RPG
Divers	2017-01-01T00:00:00	2017-12-31T00:00:00	0.0		http://melodi.int.fr/resource/Product/RPG

NDVI Export CSV

Property	Start date	End date	Value	Thumbnail	Sentinel image
VeryLow_NDVI	2017-04-19T00:00:00	2017-04-19T00:00:00	0.0		http://melodi.int.fr/resource/Product/S2A_MSIL2A_20170419T105621_N0204_F...
Low_NDVI	2017-04-19T00:00:00	2017-04-19T00:00:00	0.0		http://melodi.int.fr/resource/Product/S2A_MSIL2A_20170419T105621_N0204_F...
Middle_NDVI	2017-04-19T00:00:00	2017-04-19T00:00:00	0.0		http://melodi.int.fr/resource/Product/S2A_MSIL2A_20170419T105621_N0204_F...
VeryLow_NDVI	2017-04-29T00:00:00	2017-04-29T00:00:00	0.0		http://melodi.int.fr/resource/Product/S2A_MSIL2A_20170429T105651_N0205_F...
Low_NDVI	2017-04-29T00:00:00	2017-04-29T00:00:00	0.0		http://melodi.int.fr/resource/Product/S2A_MSIL2A_20170429T105651_N0205_F...

Figure 5: The semantic search web interface

Document name:	D2.6 Semantic Search v2	Page:	26 of 52
Reference:	D2.6	Dissemination:	PU
		Version:	2.2
		Status:	Final

3. Integration to the CANDELA platform

In this project, the processing tools must be integrated to the CANDELA cloud platform to be tested by experts and to make them accessible to the public. The process is done through Docker technologies with the help of ATOS France from WP3. This chapter presents the encapsulation of IRIT tools in Docker files and the adaptation of these tools to the CANDELA platform.

3.1 Encapsulation of the module

Docker¹⁸ is an open-source tool that automates the deployment of an application inside a software container. The key benefit of Docker is that it allows users to package an application with all of its dependencies into a standardized unit for software development. As containers have no high overhead, they enable more efficient usage of the underlying system and resources.

Docker images must be developed to integrate the tools on the CANDELA platform. The common way to make Docker images is through Docker Files, that are simple text files with instructions on how to build these images. Eventually, Docker Compose¹⁹ can be used to run multiple containers as a single service as in the case of the semantic search module.

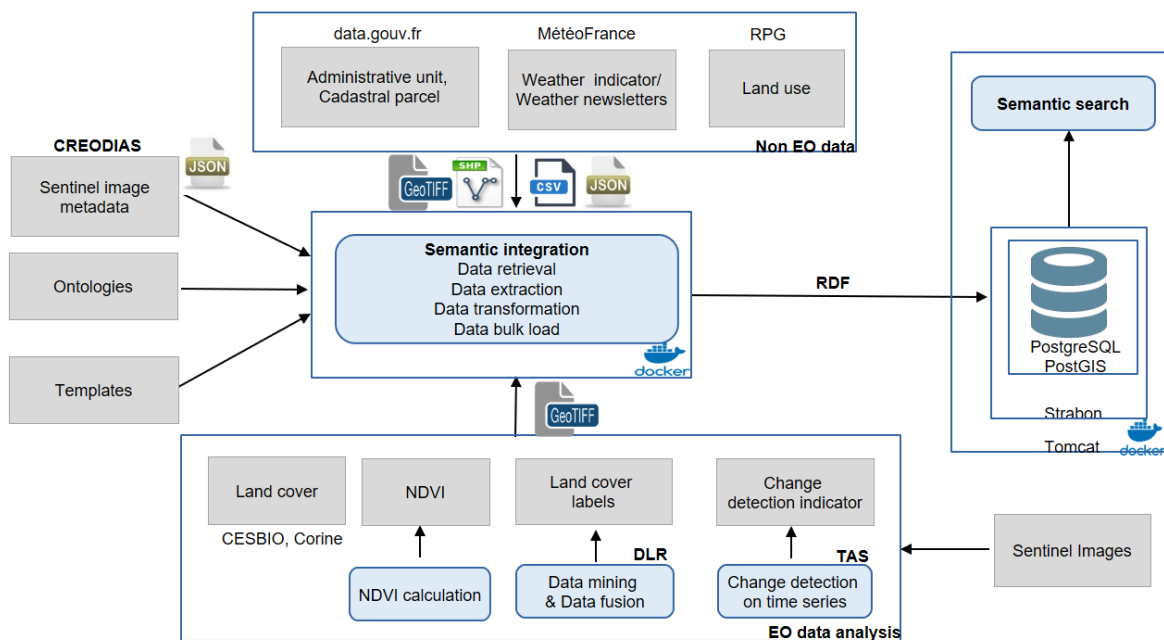


Figure 6: Overview of the semantic search modules

IRIT tools are implemented in two docker images as shown in Figure 6:

¹⁸ <https://www.docker.com/>

¹⁹ <https://docs.docker.com/compose/>

Document name:	D2.6 Semantic Search v2			Page:	27 of 52
Reference:	D2.6	Dissemination:	PU	Version:	2.2
				Status:	Final

- **Triplification docker:** This docker contains scripts and mapping templates that are used for data extraction, data transformation and triple-store bulk load. They depend on several packages, such as GDAL, GeoPandas, Rasterio for spatial processing; or RDFLib for semantic processing.
- **Semantic search docker:** This docker is used to deploy the Strabon triple-store that hosts the knowledge-base and a web interface for semantic search. To use the docker, it is necessary to have a PostgreSQL server with a DBMS) beforehand as it is the underlying component of the triple-store. The DBMS is deployed through a (third) docker²⁰. To facilitate the deployment of the module (as service), a docker-compose script is provided.

3.2 User interface

As stated above, there are two different modules, triplification and semantic search. The use of dockers is different for each of them:

- Triplification: The docker is composed of python scripts. They can be launched as services through docker containers.
- Semantic search: The docker contains web applications. Users can interact with the interface to search for desired information.

Since running a docker container is quite a complex task for people without basic knowledge of this technology, IRIT has created several Jupyter Notebook documents to demonstrate how to use the tools (see Annex A). Thanks to the Notebooks, a user can not only perform the triplification process, but he can also query the triple-store with his own semantic queries formulated in SPARQL language. Each cell in the Notebook guides the user on how to perform a stage of the process. In each cell, the usage of the statements as well as the list and role of function parameters is described in detail. A modification of the SPARQLWrapper library was done to make it compatible with the GeoJSON format returned by the Strabon triplestore.

3.3 Demonstration

We describe here how to apply the proposed tools on the Candela vineyard use-case, knowing that the same methodology can be applied on other ones. The objective of the use case is to retrieve changes in vineyard vegetation that are due to natural hazards such as frost or hail. These meteorological events can cause significant loss in vineyards and reduce wine productions. For example, in 2017, vintners reported widespread damage in the Bordeaux area, some of them losing their entire 2017 crop.

3.3.1 Triplification

For this use case, the cadastral dataset²¹ is used for the generation of territorial unit representations. The dataset can be retrieved based on the village or department that the parcels belong to. Each unit has a geometry in vector format. Other datasets used for this use case are change files obtained by processing pairs of Sentinel images from the project partners (DLR, TASF and TASI), NDVI and land

²⁰ <https://hub.docker.com/r/kartoza/postgis>

²¹ <https://cadastre.data.gouv.fr/datasets/cadastre-etalab>

Document name:	D2.6 Semantic Search v2			Page:	28 of 52
Reference:	D2.6	Dissemination:	PU	Version:	2.2
				Status:	Final

cover files coming from several open data sources (Corine Land Cover and CESBIO). These are raster files. The cadastral vector file along with available rasters over the same area and period are the input data of the triplification process. This process can be run by a simple command line such as the one bellow (where only the NDVI raster file is used and where territorial units are parcels available in the file given as “feature_file”):

```
semanticlib.Triplify( output_folder='/home/jovyan/work/data',
    feature_file='/home/jovyan/work/data/cadastre-33036-parcelles.json',
    feature_type='Parcel',
    raster_file='/home/jovyan/work/data/NDVI_S2A_MSIL2A_20170429T105651_N0205_R094_T30TYQ_20170429T110525.tif )
```

Vector and raster file are checked to guarantee that they provide enough metadata for the triplification or to check that they are on the same area for the masking process.

3.3.2 Semantic search

The semantic search on the integrated data can be done through the web interface or scripts using SPARQL protocol or HTTP requests. Although the web interface is easy and convenient to use, it is not flexible and cannot cover all analysis requirements. Scripts and HTTP requests fit better to call for the services from a software program.

- **Using semantic search interface:** The interface is available here²². Please refer to section 2.4 as it also represents the result of a search on this use case data. To perform the search, the month of April, 2017 is selected for temporal filtering. The semantic filter is set by selecting a change value: “Middle_change_DL” (classed as middle change by deep learning algorithm). For spatial filtering, the user can draw an area (near Bordeaux) on the map. After completing the spatial filtering, the geofeatures (here parcels) satisfying the constraints (changed and may be affected by the frost) are displayed on the map. When a parcel is chosen, all available information about the parcel at the given period is retrieved thanks to spatio-temporal relations between entities. As shown in the Figure 5, there is weather information; land cover information coming from CESBIO, Corine, and DLR; EO image semantics (e.g. land cover classes) and NDVI information. Each source is also associated with one or two Sentinel image products that were used to generate it.
- **Using endpoint interface:** The Strabon endpoint interface is available here²³ where users can query the triplestore. Such method makes the data analysis process easier since it provides a more flexible way to exploit the knowledge base. However, as said, this task requires basic knowledge about semantic web technologies as well as mastering the classes and properties used in the integration ontology. For example, the following query is used to retrieve observations with three-dimensional filters (spatial, temporal and semantic):

```
Select ?wkt ?foi ?props
WHERE
{
    ?observation sosa:observedProperty ?props.
```

²² <http://platform.candela-h2020.eu/semsearch/>

²³ <http://platform.candela-h2020.eu/semsearch/ep/>

Document name:	D2.6 Semantic Search v2			Page:	29 of 52
Reference:	D2.6	Dissemination:	PU	Version:	2.2
				Status:	Final

```

?observation sosa:hasSimpleResult ?value.
FILTER (?value>=0.5).
?obsCol sosa:hasMember ?observation.
?obsCol sosa:hasFeatureOfInterest ?foi.
?foi geo:hasGeometry/geo:asWKT ?wkt.
FILTER (geof:sfContains("POLYGON((-0.457 45.125, 0.936 45.085, 0.869 44.099, -0.500 44.138, -0.457 45.125))"^^geo:wktLiteral, ?wkt)).
?obsCol sosa:phenomenonTime ?timeInterval.
?timeInterval time:hasBeginning/time:inXSDDateTime ?dti1.
?timeInterval time:hasEnd/time:inXSDDateTime ?dti2.
FILTER(?dti1 <= "2017-05-01-T00:00:00"^^xsd:dateTime && ?dti2>="2017-04-01-T00:00:00"^^xsd:dateTime).
?foi tom:hasType/tom:name ?type_name.
}

```

- **Using scripts:** Users can send queries to the triplestore not only by its endpoint but also through SPARQL libraries or HTTP requests. For example, the above query can be sent using a python script.

```

from SPARQLWrapper import SPARQLWrapper, JSON
sparql = SPARQLWrapper("http://platform.candela-h2020.eu/semsearch/ep/Query")
sparql.setQuery(" ")
sparql.setReturnFormat(JSON)
results = sparql.query().convert()

```

Please refer to Annex B for results obtained with such scripts implemented on the platform.

3.4 Triplestore capacities

The Strabon triplestore can scale up to 500 million triples [7]. Since about 25 triples are required to represent a change detection result on a geofeature, up to now, the triplestore has been able to manage triplified data. A hybrid architecture with multiple triplestores could be considered in case of saturation.

A triplestore performance test was performed based on the proposed models. Strabon was deployed on a IRIT CentOS 7 server with 6 cores CPU, 12 GB of RAM and 50GB of SAS disk. Test scripts were developed on the Candela platform that generated artificial observations on all cadastral parcels of 20 villages inside the T30TYQ tile. These observations were then imported to the triplestore on the IRIT server. There are a total of 97,579,936 triples representing 3,903,800 (random) observations on 501,750 parcels.

We calculated the response time of a GeoSPARQL query with three dimensional filters to evaluate the triplestore performance. As different factors can impact this indicator, mostly the network connection, the query has been launched five times and the best response time of their run was reported. The results of the test are reported in Figure 7. The maximum response time of such a query is about 10 seconds. That is acceptable considering the modest resources of the server.

Document name:	D2.6 Semantic Search v2			Page:	30 of 52
Reference:	D2.6	Dissemination:	PU	Version:	2.2
				Status:	Final

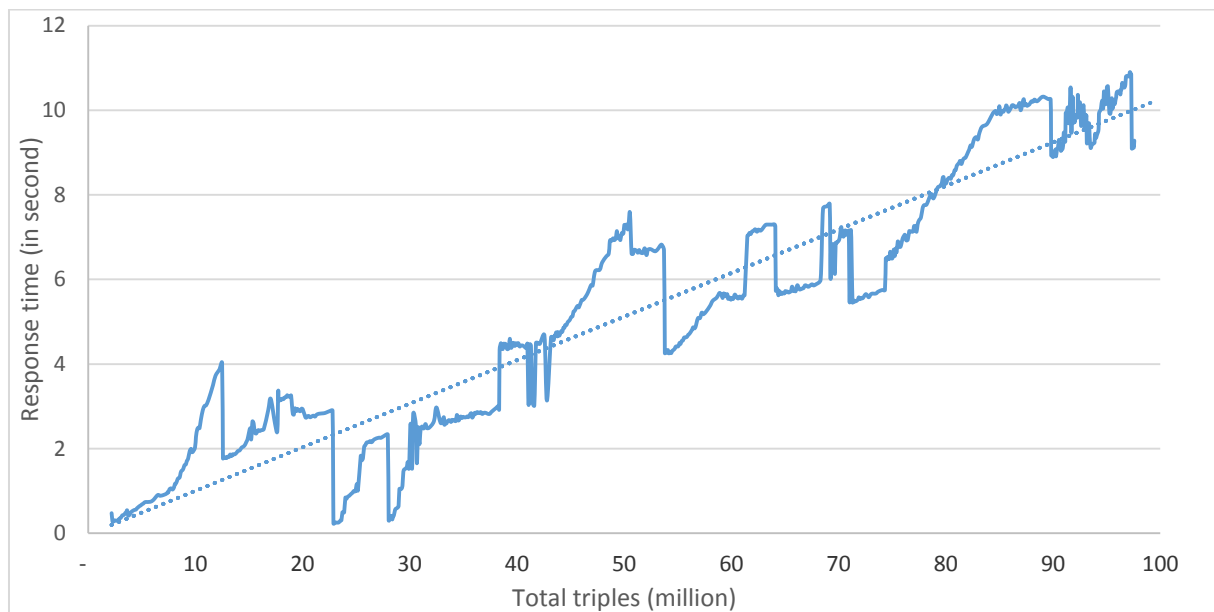


Figure 7: Strabon triplestore performance

3.5 Evolutions of the tools

The second version of the semantic search has a number of considerable changes:

- **Replacement of the triple-store:** Virtuoso, the triple-store used in the first version of the deliverable suffers from poor performances and limitations, mostly in spatial relations discovery. Although the Virtuoso triple-store, enhanced with geospatial, is functional, it cannot deal with large amounts of spatial entities and it does not support comparison between two polygons (only either point-point or polygon-point can be compared). As a consequence, we carried out a comparative study of all current open-source geospatial triple-stores and selected the one that best meet our needs. In the second version of the semantic search tool, Virtuoso is replaced by Strabon that is used in many related-EO-data projects such as H2020 Copernicus App Lab²⁴, H2020 project Big Data Europe²⁵ or Melodies²⁶.
- **Exclusion of the MongoDB database:** The initial module used a set of scripts to retrieve or read the dataset and store them in MongoDB. Another set of scripts was used next to query the data from the database and rewrite it in semantic format. Hence, these scripts require that the MongoDB server must be launched and ready for data storing/selection in advance. From now on, the DBMS has been excluded from the system. Only one script is used for both data retrieval/read and data triplification. There are several benefits coming from the exclusion, mainly:
 - Avoid data redundancy
 - Need less resources
 - Reduce system complexity, mostly when deploying it on the platform

However, the tracking of triplified data and the data selection becomes more difficult without the help of the DBMS.

²⁴ <https://www.app-lab.eu/>

²⁵ <https://www.big-data-europe.eu/>

²⁶ <https://cordis.europa.eu/project/rcn/110495/>

Document name:	D2.6 Semantic Search v2			Page:	31 of 52
Reference:	D2.6	Dissemination:	PU	Version:	2.2
				Status:	Final

- **Web application interface:** A web interface has been implemented to facilitate the semantic search process. The search result and intermediate data can also be visualized by either a diagram or a map.

There are also several major updates since the last submission of the deliverable, which are summed up below:

Integrated datasets

At the beginning, an integration model and system had first been developed for the vineyard use case, knowing that the same methodology can be applied on other ones. In this use case, we had linked together six open datasets available on the web. Among them several datasets were considered too specific for the general usage of the CANDELA platform. The specific datasets listed below were removed or replaced from the list when the tools have been generalized:

- **French administrative unit:** The GeoZones dataset²⁷ had been selected because it provides more details about administrative units. In addition to some basic information (like id, code, name, geometry) the dataset provides information such as the area, the population and especially the links to open data websites that are very useful to interlink data. Administrative units are now considered as a type of geofeature and the URI of the dataset must be provided by the user, depending on the territorial division system of the country of interest.
- **Weather report measures:** The previous dataset came from MeteoFrance website²⁸. The main information of an observation was: the atmosphere, the wind speed, the temperature, date, time. The details about the dataset are available here²⁹. The dataset has been replaced by the ECAD dataset which can provide international weather data.
- **Weather stations:** The dataset is available on MeteoFrance website³⁰. It is a GeoJSON file containing information about 62 weather stations of MeteoFrance: ID, longitude, latitude and altitude. The dataset is no longer needed since the related weather report dataset has been replaced by ECAD.
- **Weather events:** The dataset was retrieved as part of the Weather forecast bulletins (newsletter) that are available on MeteoFrance website³¹. It was extracted by a docker originally developed for the SparkInClimate project³². As several libraries used by the docker are not available anymore, we made several updates. A new version of the docker is now accessible from here³³. The docker generates a json file containing weather events described in the bulletins, which are: title, topics, facts, date and location. The dataset has been removed because it is not general enough.
- **Cadastral parcels:** Along with administrative units, the cadastral data are available from the French government data website³⁴. They exist either in GeoJSON format or shapefile. The dataset indicates the identification and the localization of cadastral parcels in the villages. The dataset

²⁷ <https://www.data.gouv.fr/en/datasets/geozones/>

²⁸ https://donneespubliques.meteofrance.fr/?fond=produit&id_produit=90&id_rubrique=32

²⁹ https://donneespubliques.meteofrance.fr/client/document/doc_parametres_synop_168.pdf

³⁰ https://donneespubliques.meteofrance.fr/donnees_libres/Txt/Synop/postesSynop.json

³¹ https://donneespubliques.meteofrance.fr/?fond=produit&id_produit=129&id_rubrique=29

³² <https://bitbucket.org/sparkindata-irit/sparkinclimate/src/master/>

³³ <https://hub.docker.com/r/h2020candela/forecast-bulletin>

³⁴ <https://cadastre.data.gouv.fr/datasets/cadastre-etalab>

Document name:	D2.6 Semantic Search v2			Page:	32 of 52
Reference:	D2.6	Dissemination:	PU	Version:	2.2
				Status:	Final

has been removed. However, we recommend users to use the dataset as preference for French cadastral data.

All these datasets have their own format, and some require to be pre-processed to select a subset of the data, or to compute new data which are more relevant for the use-case, like average values or daily sampling. We illustrated how this data is represented in an RDF graph using the EO ontology. Then we had also proposed the semantic search interface for the use case. A copy of the system is deployed on IRIT server³⁵.

Integration models

To make the integration vocabulary compliant with existing standards, we had reused various existing vocabularies or models. Each model was composed of a generic part with classes and properties reused from existing vocabularies and a specific part dedicated to data to be integrated. The generic part relied on five vocabularies: GeoSPARQL, OWL-Time, SOSA, DCAT, PROV-O.

Any index associated with an image can be represented in a similar way: a new vocabulary module needs to be added. We illustrate the process with three such properties: the land cover, which is available as an open data, the change detection results computed by CANDELA partners and the vegetation index (NDVI); All three types of data are computed thanks to image processing. These features are represented thanks to the *lci*, *change* and *ndvi* frames in Figure 8. We associated these data to EOFeatures that could be cadastral parcels, administrative units, or tiles. They are geo-localized using the geometry of the EOFeature which specializes geo:Feature.

In case a new dataset had to be taken into account for a use case, the model would need to be changed and detailed with the classes required to represent this data (as subclasses of EOComputedData).

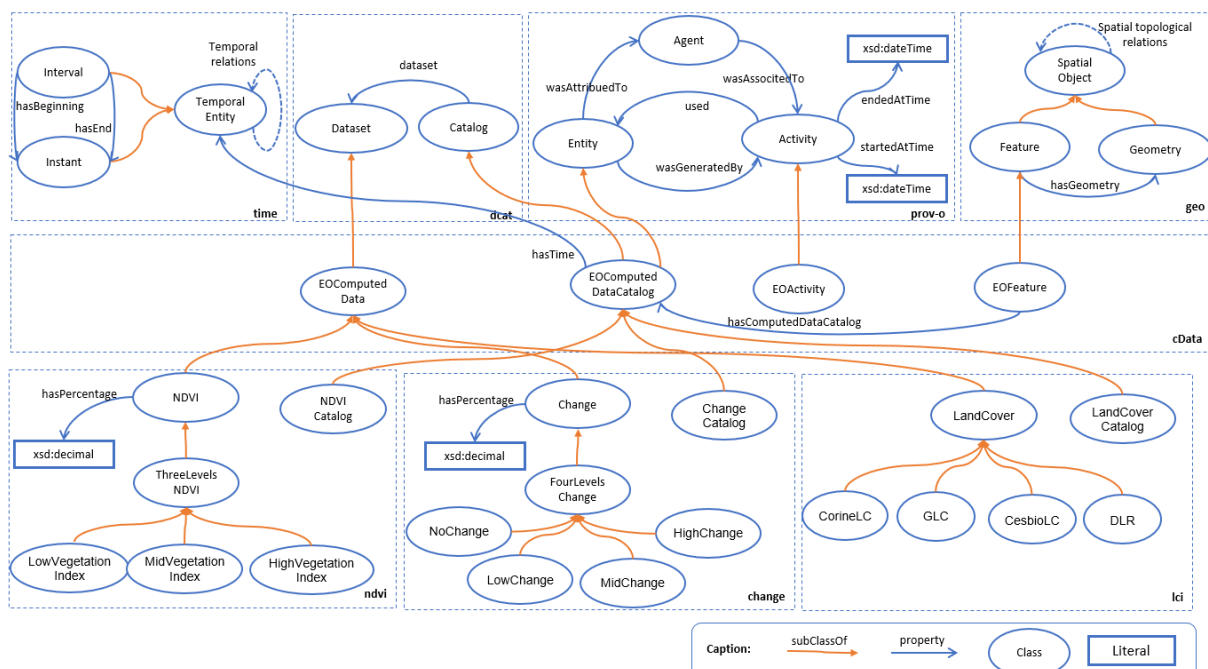


Figure 8: First version of the integration model representing computed data from Sentinel images

³⁵ <http://melodi.irit.fr/semantic-search/>

Document name:	D2.6 Semantic Search v2			Page:	33 of 52		
Reference:	D2.6	Dissemination:	PU	Version:	2.2	Status:	Final

The model has been generalized so that each new dataset will be integrated without modifying the ontology. The new dataset provides a set of observation collections that are associated with specific observable property types. The latter represent the type of the new source data, for instance: NDVI or Change. In this way, no new vocabulary will be needed when integrating the new dataset.

Triplification process

At the beginning, for each open data set, a python script was required to select and combine data from the dataset, and a template had to be defined to represent this data as RDF linked entities of the appropriate type from the integration ontology. Thanks to the generalization, the process may receive whatever vector file and raster file as input provided they describe the same area at the same time. In this way, new datasets will be integrated using the same triplification template and scripts; no implementation effort will be needed.

Semantic search

The interface of semantic search was composed of five zones as numbered in Figure 9. As the interface has been generalized, two components were removed:

- Cartographic data (4): Cartographic data had been integrated to provide additional information about the studied area (and so facilitate the drawing of the zone). This data was divided into two categories: WMS services coming from IGN, or Cesbios and Geofeatures coming from the knowledge-base, which could be an administrative unit, a weather station or a Sentinel-2 tile. Several cartographic data were integrated through WMS service to provide additional information about entities over time. The information would be useful when identifying a potential area (for example, based on the type of land cover) on the map or when comparing it with the query results (for example, comparing the land cover classification and the land cover in the past years). Also, thanks to the data visualization service, we did not need to manage additional information in the triple-store, but certainly this was not a very practical solution. This component is no longer available because it is only applicable for France and the free WMS service is very limited.
- Diagram visualization (5): The visualization represents all the query results in a simple manner using the ChartJS framework. There are two Y-axes, one on the left to represent weather forecast measures and one on the right for computed data (i.e. change level, NDVI...). The X-axis is the time-line; it is used also to provide a quick-look on Sentinel images. Each type of data is represented by a curve/line and can be displayed or hidden by clicking on its name on the top. This component is not available anymore since no adaptation was made for the new dataset and we concentrated on Jupyter notebook development.

Document name:	D2.6 Semantic Search v2			Page:	34 of 52		
Reference:	D2.6	Dissemination:	PU	Version:	2.2	Status:	Final

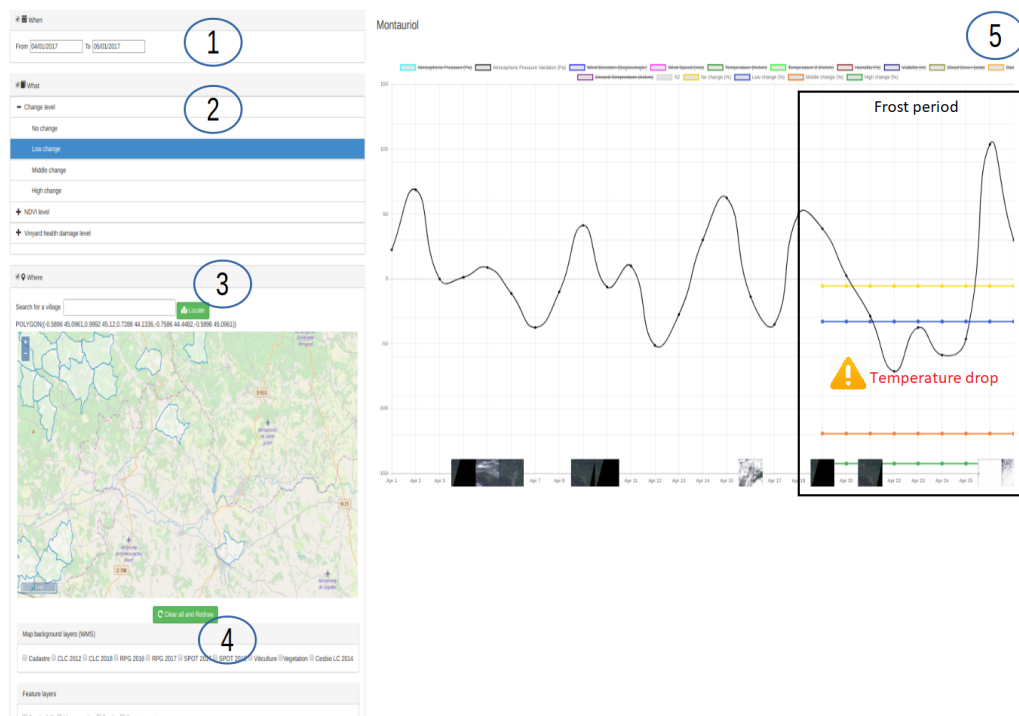


Figure 9: First version of the semantic search interface

Document name:	D2.6 Semantic Search v2			Page:	35 of 52
Reference:	D2.6	Dissemination:	PU	Version:	2.2
				Status:	Final

4. Interoperability with the other components

Conceptually, the building blocks from IRIT, TAS and DLR play a complementary role in the process of producing and querying data repositories where images and changes identified on images are either enriched with EO image semantics labels extracted by supervised Active Machine Learning (Data Mining) in the case of DLR, either linked to “contextual” data collected from the web, reusing open data-bases (like statistical data, territorial units, land cover information, or meteorological measures). All this data being heterogeneous, the first stage is to represent it in various knowledge graphs with a single vocabulary (the integration ontology) that provides the data types in a uniform and coherent way and with a standard format (RDFS-OWL). This is the role of the Triplification module. Then these knowledge graphs can be queried and linked to collect images and contextual knowledge that satisfy some search criteria. The connection between data, whatever their origin, relies on a homogeneous description of their temporal and spatial features. Querying and linking is the role of the Semantic Search module.

So IRIT tools process data that are the output of other tasks (TAS detected changes and DLR EO semantic annotations e.g., land cover classes) as well as open data. Compatibility between the different building blocks is explained in section 4.2. However, IRIT modules are not the only components of the CANDELA platform that deal with semantic approaches. Thus we will first present the various CANDELA semantic tools with their divergences in section 4.1.

4.1 Complementarity between DLR, CreoDIAS and IRIT semantic approaches

Three complementary semantic approaches are carried out in CANDELA, each of them providing RDF datasets to be integrated in the triplestore used for semantic search purposes:

- CreoDIAS provides a semantic representation of EO images metadata. This resource was not yet available when IRIT experimented data integration for the vineyard use case. However, CreDIAS API is used to retrieve Sentinel image metadata that does not exist in the semantic store.
- DLR calls semantic annotation the labelling process of pixel clusters with Land cover classes which is presented in D2.7 [3] and D2.8 [4]. The EO image content is semantically labelled in an Active [Machine] Learning process, where the user provides positive and negative training examples that are analysed. According to the desired meaning, the system suggests sorted image patches as most relevant or ambiguous. The process is iterative, fast, requires only few training samples, thus verifiable. The extracted EO image semantics is adapted and defined by the user/application. The semantic labels are stored in the Data Mining Data Base jointly with relevant EO products metadata and the features of the image patches. The Data Mining Data Base can be exploited by Multi-knowledge queries in SQL. The EO semantic labels are exported as a raster file in a forma designed based on IRIT requirements for further merging with non-EO semantics using RDF based methods. The Data Mining tool operates on Big EO Data, it was demonstrated for Sentinel-1 and Sentinel-2 images covering up to 1 Mkm².
- In IRIT tools, semantic datasets are the RDF graphs built by the triplification module from image metadata, eventually RDF metadata produced by CreoDIAS, DLR annotations of the images as well as other data coming from open sources. This data is said to be semantic because their nature and properties are explicitly represented using the *eom* ontology classes and properties.

Document name:	D2.6 Semantic Search v2			Page:	36 of 52		
Reference:	D2.6	Dissemination:	PU	Version:	2.2	Status:	Final

Some parts of the RDF repository can be selected for a specific use case so that users can query desired information through the Semantic Search module.

4.2 Compatibility between building blocks

In CANDELA, we propose to store an RDF representation of EO image meta-data, change detection resulting from TAS module D2.3 [6], EO image semantics e.g., land cover labels provided by DLR module D2.7 [4] and D2.8 [5] and any open data that could be relevant to describe, document or provide context to these images. The task is realized by the Triplification module thanks to a unique integration vocabulary (a data schema or ontology). Triplification builds an RDF representation from each dataset.

The module has recently been generalized to be compatible with other building blocks. The tool receives whatever vector file (shapefile or geojson) and raster file (GeoTIFF) as input. The raster format acts as a standard for integrating datasets:

- The change detection results of TAS France and Italy are already in raster format.
- The land cover labels annotated by DLR are stored in MonetDB. To make the results compatible with IRIT tools, DLR has developed a module to export the annotated images into raster files.
- Other open datasets, such as Cesbio or Corine land cover, also exist in raster format and can feed the process.

To be compatible with our tool, these rasters must fulfil several conditions:

- They must be a geoTiff raster with a single band
- The Coordinate Reference System and the affine georeferencing transform must be updated
- They must include metadata

Several tests were made and showed that these modules are fully compatible through raster files.

The semantic representation of datasets is computed based on spatial features of interest (or territorial units). A feature of interest may be a village, a forest or a cadastral parcel or any territorial unit provided in a vector file. Thanks to the spatial representation, either images or partner results can be linked to spatial open data that refer to the same place or area at the time when the image was captured. In order to save memory, these links are not stored but they are set at query time.

As said in section 3, the Semantic Search module provides a user-friendly interface to query the RDF graphs built for a use case with the Triplification module. The interface generates SPARQL queries that represent the search criteria expressed through the interface and returns as output the corresponding triples and a map visualization. The output may propose a list of images that satisfy the query criteria and that, in turn, may be given as input to the TAS change detection or DLR Data mining / data fusion modules for detailed image semantic annotation.

Document name:	D2.6 Semantic Search v2			Page:	37 of 52		
Reference:	D2.6	Dissemination:	PU	Version:	2.2	Status:	Final

6. Conclusion

Task 2.4 of the CANDELA project aims at demonstrating that the integration of EO data from heterogeneous sources with satellite image metadata can be a means to promote the use of these images, and that it can be achieved thanks to semantic web technologies. The current deliverable 2.6 presents the datasets, the corresponding models, and the modules that IRIT developed to demonstrate this. It details the methodology that was followed to implement a semantic repository in relation with Earth Observation images, and search facilities on this data. The main result is a framework to integrate and search data with spatial and temporal and semantic features.

The results presented in this document concern the version V2 of the semantic search system. Version V2 upgrades V1 in various ways, all of them aiming at more generality and easier adaptation to new datasets. Version V2 proposes solutions to challenges raised by version V1: the selection of relevant data sources for the vineyard use-case, the adaptation of the integration data model and the implementation of an intuitive search interface to support searching the semantic repositories. The system has been deployed on the CANDELA platform thanks to Docker technology.

As future work, we plan to apply the full approach to the big data use-case. This can be done by making a pipeline combining various modules developed by the project, including the change detection (from TASF and TASI), and the NDVI computation.

Document name:	D2.6 Semantic Search v2			Page:	38 of 52		
Reference:	D2.6	Dissemination:	PU	Version:	2.2	Status:	Final

References

- [1] C. Trojahn et al. (2018), D2.5 Semantic Search v1, Deliverable of the CANDELA project, [Online]. Available: <http://www.candela-h2020.eu/content/semantic-search-v1>
- [2] O. Dumitriu, M. Daitu, W. Yao. (2018), D2.1 Data Mining v1, Deliverable of the CANDELA project, [Online]. Available: <http://candela-h2020.eu/content/data-mining-v1>
- [3] O. Dumitru, M. Daitu, W. Yao. (2019), D2.2 Data Mining v2, Deliverable of the CANDELA project, [Online]. Available: <http://candela-h2020.eu/content/data-mining-v2>
- [4] O. Dumitru, M. Daitu, W. Yao. (2018), D2.7 Data Fusion v1, Deliverable of the CANDELA project, [Online]. Available: <http://candela-h2020.eu/content/data-fusion-v1>.
- [5] O. Dumitru, M. Daitu, W. Yao. (2019), D2.8 Data Fusion v2, Deliverable of the CANDELA project, [Online]. Available: [2http://candela-h2020.eu/content/data-fusion-v1](http://candela-h2020.eu/content/data-fusion-v1).
- [6] M. Aubrun et al. (2018), D2.3 Deep Learning v1, Deliverable of the CANDELA project, [Online]. Available: <http://www.candela-h2020.eu/content/deep-learning-v1>
- [5] Kyzirakos, K., M. Karpathiotakis, K. M. (2012). Strabon: A semantic geospatial DBMS. In The Semantic Web ISWC 2012, Berlin, pp. 295–311. Springer.

References to tools, programming languages, datasets and other data resources are given as end of page footnotes.

Document name:	D2.6 Semantic Search v2			Page:	39 of 52	
Reference:	D2.6	Dissemination:	PU	Version:	2.2	Status: Final

Annexes

Annex A: Jupyter notebook

IRIT has provided general functions for triplification and semantic search tools. Several examples are demonstrated by two Jupyter notebooks. Users can choose (and combine) several functions depending on their need. The basic idea is:

- For the triplification: Users should prepare at least one raster file (with metadata included) (or one raster per dataset) and a vector file (describing territorial units like parcels or forests).
 - We provide support to get raster files containing the following kinds of data:
 - + For the change detection and land cover, users can run the notebook provided by TASF/TASI or DLR.
 - + For the NDVI, IRIT has implemented functions to directly generate NDVI raster from Sentinel image. They are in semanticlib.py and demonstrated by the Triplification notebook.
 - + For open data like CESBIO or Corine Land Cover: Users can download these rasters from open data and add metadata to them before processing. IRIT has put a prepared sample of each dataset on the platform.

The metadata of raster files must be defined as follows.

Attribute	Description	Example
Start_date	Start date of the validity period. Format: YYYYMMDD	20170409
End_date	End date of the validity period. Format: YYYYMMDD Is optional if Start_date = End_date	20170429
Category	Category of the dataset: <ul style="list-style-type: none"> - 'NDVI' for the NDVI - 'Change_xx' for change - 'LC_xx' for land cover 	NDVI Change_DL LC_DM LC_Cesbio
Product_id1	Title of the first Sentinel image used in the EO analysis process. Is optional for open data, such as Cesbio or Corine land cover	S2A_MSIL2A_20170409T105651_N0204_R094_T30TYQ_20170409T110529.SAFE
Product_id2	Title of the second Sentinel image used in the EO analysis process. Is optional if only one image was used, (i.e. NDVI raster)	S2A_MSIL2A_20170409T105651_N0204_R094_T30TYQ_20170409T

Document name:	D2.6 Semantic Search v2			Page:	40 of 52
Reference:	D2.6	Dissemination:	PU	Version:	2.2
				Status:	Final

		110529.SAFE
--	--	-------------

User can update the raster metadata by functions of rasterio, or GDAL, for example:

```
with rasterio.open("result_L2A_T30TYQ_20170419T105621.tif", mode='r+') as src:
    src.update_tags(Category='Change_DL',
                    Start_Date='20170419',
                    End_Date='20170429')
```

o Features vector file:

+ For the vineyard use case: Users can download parcel data from data.gov.fr or elsewhere. There are also some samples on the platform.

+ For the forest use case: you can use the shape file from Forest Data Bank or the GeoJSON file converted by IRIT.

- For semantic search: Users can combine different functions depending on their purpose. Basic python knowledge may be required to declare variables or loop through arrays.

The figures below are screenshots of a Jupyter notebook example allowing to triplify cadastral data and change detection results, upload triplified data to the semantic base and perform semantic queries to discover the integrated data.

These documents will be accessible on CANDELA platform in the folder *public/IRIT*.

Document name:	D2.6 Semantic Search v2			Page:	41 of 52		
Reference:	D2.6	Dissemination:	PU	Version:	2.2	Status:	Final

Triplication

Compute the NDVI of Sentinel image and triplify these information at parcel level

```
In [1]: import os
import semanticlib
import creodias_lib as creodias
```

Use creodias_lib to search for desired Sentinel images. Create a folder that contains Sentinel-2 data of interest.

```
In [2]: import creodias_lib as creodias
""" Search Sentinel2 data in creodias
:param satellite: Sentinel2
:param productType: type of the product
:param name : string defining the area to search
        box : bounding box defining the area to search (west, south, east, north)
:param startDate : beginning of the period
:param completionDate: end of the period
:param cloudCover (optional): filter to select images without clouds """
results = creodias.search('Sentinel2',
                          productType='L2A',
                          name='Aquitaine',
                          startDate='2017-04-01',
                          completionDate='2017-04-30',
                          cloudCover='[0,20]',
                          status='all')

""" Filter the selection according to a character sequence
:param : the selection
:param : the character sequence
"""
filtered_res = creodias.filter_product(results, ['.*T30TYQ'])
creodias.display_products_identifier(filtered_res)
```

Search for results...
38 products found

Product Identifier
/eodata/Sentinel-2/MSI/L2A/2017/04/29/S2A_MSIL2A_20170429T105651_N0205_R094_T30TYQ_20170429T110525.SAFE
/eodata/Sentinel-2/MSI/L2A/2017/04/19/S2A_MSIL2A_20170419T105621_N0204_R094_T30TYQ_20170419T110601.SAFE
/eodata/Sentinel-2/MSI/L2A/2017/04/09/S2A_MSIL2A_20170409T105651_N0204_R094_T30TYQ_20170409T110529.SAFE
/eodata/Sentinel-2/MSI/L2A/2017/04/06/S2A_MSIL2A_20170406T105021_N0204_R051_T30TYQ_20170406T105317.SAFE

```
In [3]: link_folder = './links'
""" Delete symbolic links inside the destination directory
:param destination: folder where to delete symbolic links (the folder must exist)
"""
creodias.delete_links_in_folder(link_folder)

""" Create symbolic links to the Sentinel-2 data products
:param : the selection
:param destination: folder where to create symbolic links
"""
creodias.create_links(filtered_res, destination=link_folder)
```

Compute the NDVI

```
In [4]: """ Compute the NDVI from image
:param product_title: Title of Sentinel image
:param links_folder: Creodias links folder
:param output_folder: Output folder (default: .) """
image = filtered_res[3]
title = image['title']
ndvi=semanticlib.compute_ndvi(product_title=title,
                              links_folder=link_folder,
                              output_folder='/home/jovyan/work/data')

/home/jovyan/work/IRIT/semanticlib.py:369: RuntimeWarning: invalid value encountered in true_divide
(np.float16(nir) + np.float16(red))
```

Document name:	D2.6 Semantic Search v2			Page:	42 of 52
Reference:	D2.6	Dissemination:	PU	Version:	2.2
				Status:	Final

```
In [5]: """
:param output_folder: Output folder
:param feature_file: Vector file (parcel, forest), in geojson format
:param feature_type: Type of feature: parcel, forest, village...
:param raster_file: The NDVI file """

semanticlib.Triplify( output_folder='/home/jovyan/work/data',
                      feature_file='/home/jovyan/work/data/cadastre-33036-parcelles.json',
                      feature_type='Parcel',
                      raster_file=ndvi,
                      )
```

Out[5]: <owslib.wps.WPSExecution at 0x7fd33e8e4c90>

Semantic search

```
In [1]: import semanticlib
```

Semantic search

Search units according to geofeature or area or weather data

Provide a DataFrame with the available types of geofeature

```
In [2]: """ Identify the different types of geofeature
"""
feature_types=semanticlib.query_feature_type()
```

	type
0	Parcel

Provide a DataFrame with the units of interest

```
In [3]: """ Search features belonging to a type
:param DataFrame that contains a column with the id of a geofeature
:param line_id: Row that corresponds to the desired geofeature (default value: 0)
:param limit: max number of features returned (default value: all)
"""
features=semanticlib.query_feature_by_type(feature_types,
                                           line_num=0,
                                           limit=10)
```

	id	geometry
0	47001000AD0158	POLYGON ((0.62938 44.21007, 0.62941 44.20998, ...
1	47001000AD0159	POLYGON ((0.62941 44.20998, 0.62938 44.21007, ...
2	47001000AD0488	POLYGON ((0.63269 44.21247, 0.63282 44.21246, ...
3	47001000AD0422	POLYGON ((0.63954 44.21244, 0.63953 44.21256, ...
4	47001000AD0490	POLYGON ((0.63474 44.21224, 0.63481 44.21198, ...
5	47001000AD0002	POLYGON ((0.63107 44.21349, 0.63079 44.21363, ...
6	47001000AD0461	POLYGON ((0.63193 44.21449, 0.63192 44.21449, ...
7	47001000AD0458	POLYGON ((0.63235 44.21341, 0.63233 44.21340, ...
8	33165000AB0111	POLYGON ((-0.44813 44.82264, -0.44797 44.82282, ...
9	33165000AB0062	POLYGON ((-0.44615 44.82450, -0.44552 44.82435, ...

Search features within an area

```
In [4]: """ Search features within an area (and)
:param geometry: the area to search in WGS84 geometry
:param feature_type: Desired type of feature (default value: all)
:param limit: max number of features returned (default value: all)
"""
features2=semanticlib.query_feature_by_type_and_zone(geometry="POLYGON((-0.457 45.125, 0.936 45.085, 0.869 44.099, -0.500 44.138, -0.457 45.125))",
                                                    feature_type="Parcel",
                                                    limit=10)
```

Document name:	D2.6 Semantic Search v2			Page:	43 of 52	
Reference:	D2.6	Dissemination:	PU	Version:	2.2	Status: Final

	id	type	geometry
0	47001000AD0158	Parcel	POLYGON ((0.62938 44.21007, 0.62941 44.20998, ...
1	47001000AD0159	Parcel	POLYGON ((0.62941 44.20998, 0.62938 44.21007, ...
2	47001000AD0488	Parcel	POLYGON ((0.63269 44.21247, 0.63282 44.21246, ...
3	47001000AD0422	Parcel	POLYGON ((0.63954 44.21244, 0.63953 44.21256, ...
4	47001000AD0490	Parcel	POLYGON ((0.63474 44.21224, 0.63481 44.21198, ...
5	47001000AD0002	Parcel	POLYGON ((0.63107 44.21349, 0.63079 44.21363, ...
6	47001000AD0461	Parcel	POLYGON ((0.63193 44.21449, 0.63192 44.21449, ...
7	47001000AD0458	Parcel	POLYGON ((0.63235 44.21341, 0.63233 44.21340, ...
8	33165000AB0111	Parcel	POLYGON ((-0.44813 44.82264, -0.44797 44.82282, ...
9	33165000AB0062	Parcel	POLYGON ((-0.44615 44.82450, -0.44552 44.82435, ...

Search weather station from which weather observations were made

```
In [5]: """ Search geolocated weather stations that respect a weather data criteria
:param measure: Name of the desired weather data
:param condition: Condition to respect
:param start_date: beginning of the period (default value: "2017-04-01")
:param end_date: end of the period (default value: "2017-05-01")
:param geom: area (or list of areas) to search in WGS84 geometry (default value: None --> for all the Earth)
:param feat: Name of the desired feature (default value: all)
:param limit: max number of units returned (default value: all)
"""
stations=semanticlib.query_station_by_meteo(measure='Min_Temperature',
condition="<0",
start_date="2017-04-01",
end_date="2017-05-01",
geometry="all",
limit=10)
```

	station_name	date	value	geometry
0	TORUN	2017-04-24T12:00:00Z	-0.5	POINT (18.59556 53.04167)
1	TORUN	2017-04-21T12:00:00Z	-0.9	POINT (18.59556 53.04167)
2	TORUN	2017-04-19T12:00:00Z	-1.6	POINT (18.59556 53.04167)
3	TORUN	2017-04-17T12:00:00Z	-1.1	POINT (18.59556 53.04167)
4	GOURDON	2017-04-28T12:00:00Z	-1.8	POINT (1.39667 44.74500)
5	GOURDON	2017-04-27T12:00:00Z	-2.0	POINT (1.39667 44.74500)
6	GOURDON	2017-04-21T12:00:00Z	-2.0	POINT (1.39667 44.74500)

Search features near a weather station

```
In [7]: """ Search geolocated weather stations that respect a weather data criteria
:param distance: Maximum distance (in meter) from the station
:param stations: Stations dataframe
:param line_num: Line number
:param feature_type: Desired type of feature (default value: all)
:param limit: max number of units returned (default value: all)
"""
feature3=semanticlib.query_feature_by_station(distance=100000,
stations=stations,
line_num=4,
feature_type="all",
limit=10)
```

	id	type	geometry
0	24140000AT0023	Parcel	POLYGON ((0.54832 44.83146, 0.54952 44.83302, ...
1	24140000AT0042	Parcel	POLYGON ((0.54849 44.83526, 0.54851 44.83532, ...
2	24140000AT0034	Parcel	POLYGON ((0.55032 44.83343, 0.55071 44.83363, ...
3	24140000AT0026	Parcel	POLYGON ((0.55337 44.83386, 0.55285 44.83419, ...
4	24140000AT0037	Parcel	POLYGON ((0.55196 44.83606, 0.55183 44.83617, ...
5	24140000AT0030	Parcel	POLYGON ((0.55285 44.83419, 0.55476 44.83505, ...
6	24140000AT0018	Parcel	POLYGON ((0.54795 44.83412, 0.54764 44.83409, ...
7	24140000AT0065	Parcel	POLYGON ((0.55016 44.83356, 0.55205 44.83468, ...
8	24140000AT0039	Parcel	POLYGON ((0.55150 44.83651, 0.54949 44.83571, ...
9	24140000AT0038	Parcel	POLYGON ((0.55115 44.83693, 0.55129 44.83679, ...

Document name:	D2.6 Semantic Search v2			Page:	44 of 52
Reference:	D2.6	Dissemination:	PU	Version:	2.2
				Status:	Final

Search Sentinel data product according to an unit

In [8]: `df = features2 #DataFrame that contains a column with the geometry of the feature (weather station, parcel or forest)`
`row = 0 #Row that corresponds to the desired unit`

```
import creodias_lib as creodias
product = creodias.search('Sentinel2',
                          status='all',
                          cloudCover='[0,20]',
                          startDate='2017-07-30',
                          completionDate='2017-09-30',
                          productType='L2A',
                          geometry=df.at[row,'geometry'])
creodias.display_products_identifier(product)
```

Search for results...
 5 products found

Product Identifier
/eodata/Sentinel-2/MSI/L2A/2017/08/29/S2B_MSIL2A_20170829T105019_N0205_R051_T31TCK_20170829T105633.SAFE
/eodata/Sentinel-2/MSI/L2A/2017/08/29/S2B_MSIL2A_20170829T105019_N0205_R051_T31TCJ_20170829T105633.SAFE
/eodata/Sentinel-2/MSI/L2A/2017/08/19/S2B_MSIL2A_20170819T105029_N0205_R051_T31TCK_20170819T105403.SAFE
/eodata/Sentinel-2/MSI/L2A/2017/08/14/S2A_MSIL2A_20170814T105031_N0205_R051_T30TYQ_20170814T105517.SAFE
/eodata/Sentinel-2/MSI/L2A/2017/08/04/S2A_MSIL2A_20170804T105031_N0205_R051_T30TYQ_20170804T105328.SAFE

Search observations according to types of Observation

Provide a DataFrame with the different types of observation and their levels

In [9]: `""" Identify the different types of observation and their Levels """`
`obs_prop=semanticlib.query_observation_type()`

	type_name	prop_name
0	NDVI	VeryLow_NDVI
1	NDVI	Low_NDVI
2	NDVI	Middle_NDVI
3	NDVI	High_NDVI
4	Change_DL	VeryLow_Change_DL
5	Change_DL	Low_Change_DL
6	Change_DL	Middle_Change_DL
7	Change_DL	High_Change_DL

Provide a DataFrame with the observations of interest

In [10]: `""" Search observations with the desired property in a defined period and area
:param obs_type: Name of the desired type of observation
:param obs_prop: Name of the desired property (default value: all)
:param min_value: Minimal percentage of pixels inside an unit that correspond to the property selected (default value: 0.5)
:param start_date: beginning of the period (default value: "2017-04-01")
:param end_date: end of the period (default value: "2017-05-01")
:param geom: area (or list of areas) to search in WGS84 geometry (default value: None --> for all the Earth)
:param feature_type: Name of the desired feature (default value: all)
:param limit: max number of units returned (default value: all)
"""`
`observations=semanticlib.query_observation(obs_type='Change_DL',
 obs_prop='High_Change_DL',
 min_value=0.5,
 start_date="2017-04-01",
 end_date="2017-05-01",
 geom="POLYGON((-0.457 45.125, 0.936 45.085, 0.869 44.099, -0.500 44.138, -0.457 45.125))",
 feature_type='Parcel',
 limit=10)`

	feature	type	start_date	end_date	prop_name	value	obsCol	geometry
0	33315000ZB0023	Parcel	2017-04-19	2017-04-29	High_Change_DL	0.78	GFObservationCollection/Change_DL_Parcel_33315...	POLYGON ((-0.10909 45.07562, -0.10903 45.07558...
1	33315000ZB0134	Parcel	2017-04-19	2017-04-29	High_Change_DL	0.80	GFObservationCollection/Change_DL_Parcel_33315...	POLYGON ((-0.11958 45.07581, -0.11954 45.07581...
2	33315000ZB0026	Parcel	2017-04-19	2017-04-29	High_Change_DL	1.00	GFObservationCollection/Change_DL_Parcel_33315...	POLYGON ((-0.10774 45.07727, -0.10545 45.07785...
3	33315000ZB0140	Parcel	2017-04-19	2017-04-29	High_Change_DL	0.99	GFObservationCollection/Change_DL_Parcel_33315...	POLYGON ((-0.11619 45.07831, -0.11689 45.07865...

Document name:	D2.6 Semantic Search v2			Page:	45 of 52		
Reference:	D2.6	Dissemination:	PU	Version:	2.2	Status:	Final

4	33315000ZB0024	Parcel	2017-04-19	2017-04-29	High_Change_DL	0.92	GFObservationCollection/Change_DL_Parcel_33315...	POLYGON ((-0.10855 45.07541, -0.10795 45.07555...
5	33315000ZB0015	Parcel	2017-04-19	2017-04-29	High_Change_DL	0.94	GFObservationCollection/Change_DL_Parcel_33315...	POLYGON ((-0.10779 45.07741, -0.10548 45.07826...
6	333150000E0195	Parcel	2017-04-19	2017-04-29	High_Change_DL	0.50	GFObservationCollection/Change_DL_Parcel_33315...	POLYGON ((-0.09372 45.06718, -0.09364 45.06714...
7	333150000E0458	Parcel	2017-04-19	2017-04-29	High_Change_DL	1.00	GFObservationCollection/Change_DL_Parcel_33315...	POLYGON ((-0.09481 45.06646, -0.09474 45.06644...
8	333150000E0511	Parcel	2017-04-19	2017-04-29	High_Change_DL	0.50	GFObservationCollection/Change_DL_Parcel_33315...	POLYGON ((-0.09906 45.06866, -0.09879 45.06863...
9	333150000E0241	Parcel	2017-04-19	2017-04-29	High_Change_DL	0.58	GFObservationCollection/Change_DL_Parcel_33315...	POLYGON ((-0.09432 45.06525, -0.09449 45.06508...

```
In [12]: """ Search observations with the desired property in a defined period and feature
:param obs_type: Name of the desired type of observation
:param obs_prop: Name of the desired property (default value: all)
:param min_value: Minimal percentage of pixels inside an unit that correspond to the property selected (default value: 0.5)
:param start_date: beginning of the period (default value: "2017-04-01")
:param end_date: end of the period (default value: "2017-05-01")
:param feature_id: ID of the desired feature
:param limit: max number of units returned (default value: all)
"""
observations=semanticlib.query_observation_by_feature(obs_type='Change_DL',
                                                    obs_prop='all',
                                                    min_value=0.5,
                                                    start_date="2017-04-01",
                                                    end_date="2017-05-01",
                                                    feature_id='24140000AT0038',
                                                    limit=10)
```

	feature	start_date	end_date	prop_name	value	obsCol
0	24140000AT0038	2017-04-19T00:00:00	2017-04-29T00:00:00	VeryLow_Change_DL	0.98	GFObservationCollection/Change_DL_Parcel_24140...

Search Sentinel data product according to an Observation

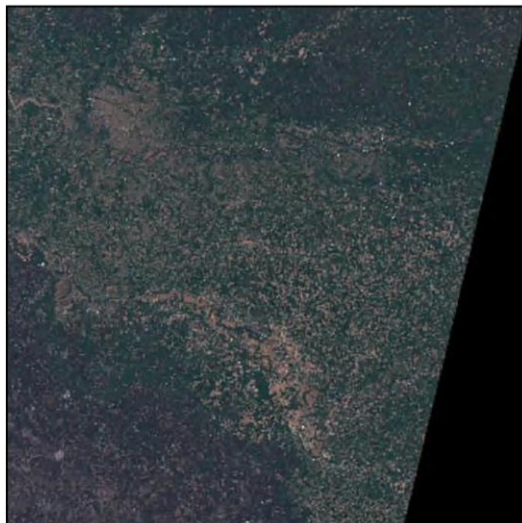
Provide the ids of the Sentinel data products used to analyze the observation of interest

```
In [13]: """ Search Sentinel data product with the desired observation
:param DataFrame that contains a column with the id of an observation
:param line_num: Row that corresponds to the desired observation (default value: 0)
"""
products=semanticlib.query_observation_source(observations,
                                             line_num=0)
```

	image
0	Product/S2A_MSIL2A_20170419T105621_N0204_R094_...
1	Product/S2A_MSIL2A_20170429T105651_N0205_R094_...

```
In [14]: semanticlib.show_image(products)

S2A_MSIL2A_20170419T105621_N0204_R094_T30TYQ_20170419T110601
Quick look from L1C
Search for results...
1285 products found
```



Document name:	D2.6 Semantic Search v2			Page:	46 of 52
Reference:	D2.6	Dissemination:	PU	Version:	2.2
				Status:	Final

Search weather data according to an observation

Provide a DataFrame with the available weather data (Only data related to the vineyard and forest use case, for other area please ask)

```
In [16]: """ Identify the wheater data
        """
        measures=semanticlib.query_measure()
```

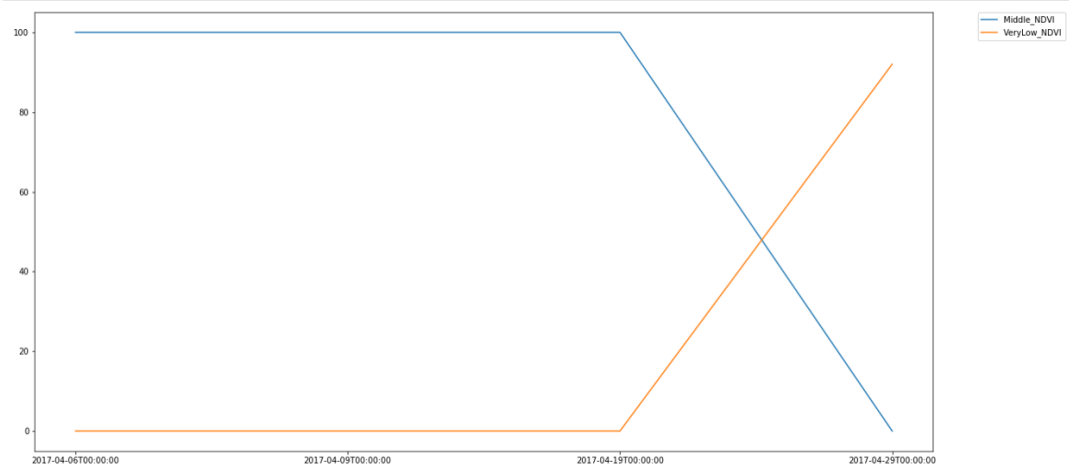
	name
0	Humidity
1	Max_Temperature
2	Min_Temperature
3	Mean_Temperature
4	Precipitation
5	WindSpeed

Provide a DataFrame with the desired weather data

```
In [ ]: """ Search a specific weather data from an observation of interest
        :param measure: Name of the desired weather data (default value: all --> for all measure)
        :param observations: Dataframe that contains the observation of interest, the dataframe must contains features geometry
        :param line_num: Row that corresponds to the observation of interest
        """
        weather=semanticlib.query_weather_by_observation(measure='Min_Temperature',
        observations=observations,
        line_num=0)
```

Show graph from dataframe (ndvi, change..)

```
In [17]: semanticlib.show_graph(observations, 'line')
```



Document name:	D2.6 Semantic Search v2			Page:	47 of 52	
Reference:	D2.6	Dissemination:	PU	Version:	2.2	Status: Final

Annex B: Jupyter notebook - Hackathon

The figure below is a screenshot of the Jupyter notebook document allowing to triplify cadastral data and change detection results and upload triplified data to the semantic base. The notebook was used to demonstrate our tools during the Hackathon in Frascati.

This document is accessible on CANDELA platform in the folder *public/Hackaton/Exercise1*.

```
[1]: # From the above list of villages, choose a village and download its cadastral parcels data through its insee number.
# The cadastral data will be used for ndvi and change detection computation
from semsearch import viz, download, change, parcel, upload_webdav
fn = download.download('cadastre', './download', '33227')
print(fn)

./download/cadastre-33227-parcelles.json

[ ]: #Compute change detection levels on cadastral parcels of the selected village
#First of all, show the template used for the triplification
viz.showFile('./semsearch/templates/template_Change_dept.ttl', 20)

[2]: #Execute cadastral parcels triplification with Land-cover data from Cesbio
files = parcel.triplify_dataset('./download/OCS_2018_CESBIO.tif', fn, '33227')
print(files)

6749/6750
total nr of triples 40012
Writing triples to file: ./rdf/Parcel_33227_4.ttl
Elapsed time : 43.35386347770691 seconds
['./rdf/Parcel_33227_0.ttl', './rdf/Parcel_33227_1.ttl', './rdf/Parcel_33227_2.ttl', './rdf/Parcel_33227_3.ttl', './rdf/Parcel_33227_4.ttl']

[ ]: viz.showFile(files[0],30)

[3]: upload_webdav.upload_dataset(files, '/opt/tomcat/webapps/rdf/', 'melodi', 'tomcat', 'http://melodi.irit.fr/ep/Store', 'http://melodi.irit.fr/rdf/')
Enter password for endpoint http://melodi.irit.fr/ep/Store (user melodi )
.....
Enter password for webdav http://melodi.irit.fr/rdf/ (user tomcat )
.....
Validating file: /home/jovyan/work/rdf/Parcel_33227_0.ttl ( 1332 KB)
Uploading file /home/jovyan/work/rdf/Parcel_33227_0.ttl to webdav: http://melodi.irit.fr/rdf/Parcel_33227_0.ttl
Upload file to Strabon: /opt/tomcat/webapps/rdf/Parcel_33227_0.ttl
Connection: 200
Result: Data stored successfully!
Validating file: /home/jovyan/work/rdf/Parcel_33227_1.ttl ( 1327 KB)
Uploading file /home/jovyan/work/rdf/Parcel_33227_1.ttl to webdav: http://melodi.irit.fr/rdf/Parcel_33227_1.ttl
Upload file to Strabon: /opt/tomcat/webapps/rdf/Parcel_33227_1.ttl
Connection: 200
Result: Data stored successfully!
Validating file: /home/jovyan/work/rdf/Parcel_33227_2.ttl ( 1354 KB)
Uploading file /home/jovyan/work/rdf/Parcel_33227_2.ttl to webdav: http://melodi.irit.fr/rdf/Parcel_33227_2.ttl
Upload file to Strabon: /opt/tomcat/webapps/rdf/Parcel_33227_2.ttl
Connection: 200
Result: Data stored successfully!
Validating file: /home/jovyan/work/rdf/Parcel_33227_3.ttl ( 1356 KB)
Uploading file /home/jovyan/work/rdf/Parcel_33227_3.ttl to webdav: http://melodi.irit.fr/rdf/Parcel_33227_3.ttl
Upload file to Strabon: /opt/tomcat/webapps/rdf/Parcel_33227_3.ttl
Connection: 200
Result: Data stored successfully!
Validating file: /home/jovyan/work/rdf/Parcel_33227_4.ttl ( 213 KB)
Uploading file /home/jovyan/work/rdf/Parcel_33227_4.ttl to webdav: http://melodi.irit.fr/rdf/Parcel_33227_4.ttl
Upload file to Strabon: /opt/tomcat/webapps/rdf/Parcel_33227_4.ttl
Connection: 200
Result: Data stored successfully!
Elapsed time : 1 minutes and 16.087368488311768 seconds

[4]: #Execute the change detection on T30TYQ S2 Tile. To save time, let's suppose that we already have the result
#Execute the computation
files = change.triplify_dataset("./download/result_L2A_T30TYQ_20170419T105621_dense.tif", fn, '33227')
print(files)

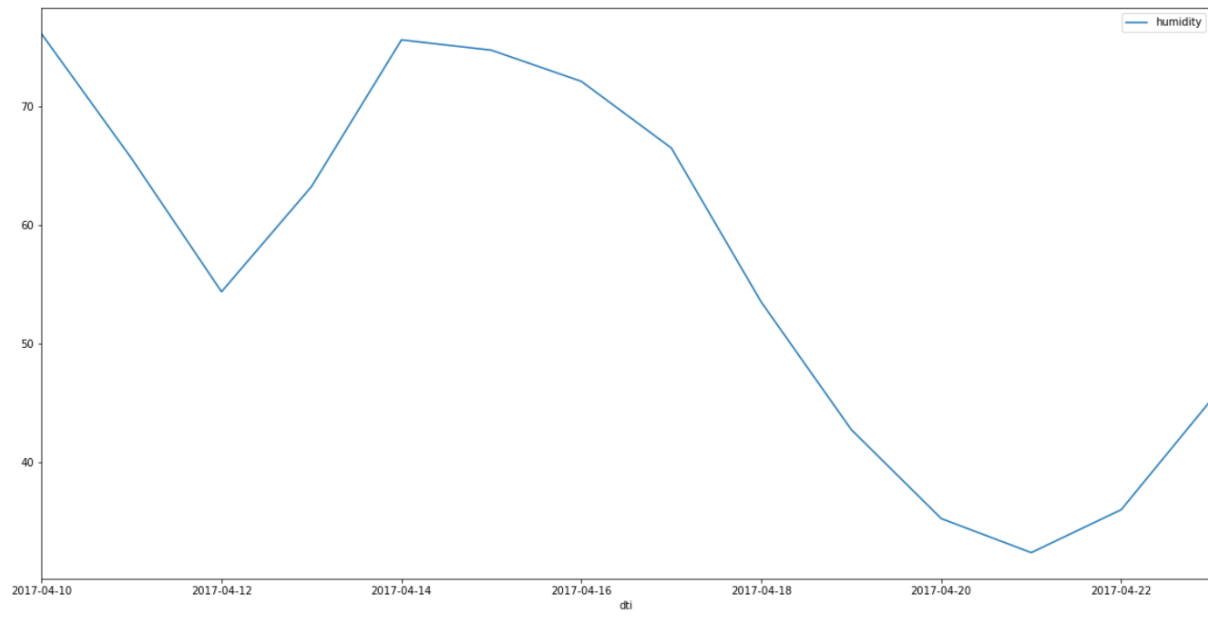
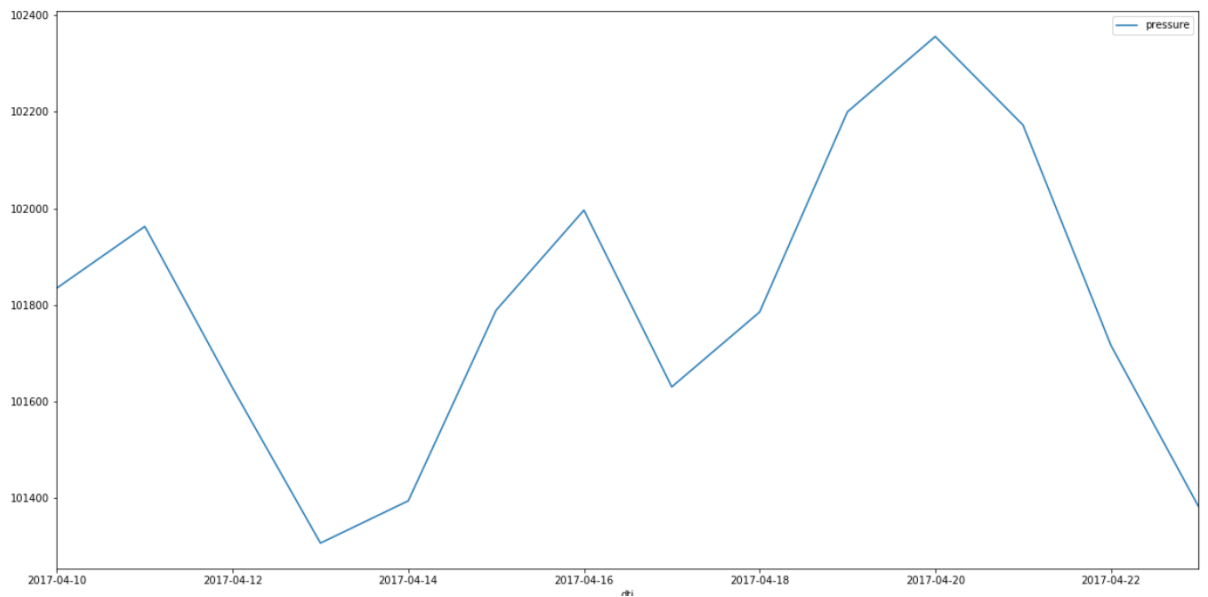
Number of triples 90048
Writing triples to file: 33227Change6.ttl
Elapsed time : 42.8044490814209 seconds
Elapsed time : 42.805338621139526 seconds
['33227Change0.ttl', '33227Change1.ttl', '33227Change2.ttl', '33227Change3.ttl', '33227Change4.ttl', '33227Change5.ttl', '33227Change6.ttl']

[ ]: #Show the result
viz.showFile(files[0],30)

[5]: #Upload semantic data to the triplestore
upload_webdav.upload_dataset(files, '/opt/tomcat/webapps/rdf/', 'melodi', 'tomcat', 'http://melodi.irit.fr/ep/Store', 'http://melodi.irit.fr/rdf/')
Enter password for endpoint http://melodi.irit.fr/ep/Store (user melodi )
.....
Enter password for webdav http://melodi.irit.fr/rdf/ (user tomcat )
.....
Validating file: /home/jovyan/work/33227Change0.ttl ( 2112 KB)
Uploading file /home/jovyan/work/33227Change0.ttl to webdav: http://melodi.irit.fr/rdf/33227Change0.ttl
Upload file to Strabon: /opt/tomcat/webapps/rdf/33227Change0.ttl
Connection: 200
```

Document name:	D2.6 Semantic Search v2			Page:	48 of 52
Reference:	D2.6	Dissemination:	PU	Version:	2.2
				Status:	Final


```
[3]: df=query.query_meteo('33227', '2017-04-10', '2017-04-25', 1000)
df['value'] = df['value'].astype(float)
plt.rcParams["figure.figsize"] = (20,10)
df1=df.loc[df['procedure'] == 'Procedure/pres']
df3=df.loc[df['procedure'] == 'Procedure/u']
df1.plot(kind='line',x='dti',y='value', label='pressure')
df3.plot(kind='line',x='dti',y='value', label='humidity')
plt.show()
```



Document name:	D2.6 Semantic Search v2			Page:	50 of 52
Reference:	D2.6	Dissemination:	PU	Version:	2.2
				Status:	Final

Annex C: Collaboration with the other projects

IRIT has begun to examine the possibility to collaborate with other European projects which also work on EO data. Concretely, during a Hackaton in Frascati, on November 2019, we had many discussions with the CERTH team of the EOPEN project, who works on the information available in tweets thanks to a semantic approach. Below is the comparison of the two systems proposed by CERTH and IRIT.

Semantics	EOPEN - GraphDB	EOPEN - Strabon	CANDELA
Input data format	JSON	GeoTIFF	CSV, JSON/GeoJSON, GeoTIFF, Shape files
Metadata type	EO data (satellite images) and non-EO (tweets)	EO derived products	EO data (satellite images) Open Data available as raster files (land cover, land use) or vectorial geometry (territorial units) Data resulting from EO image analysis (change, land cover annotation) Contextual data
Derived data	Tweet locations, topics		
Semantic language	RDF	RDF	OWL
Semantic format	Turtle	N3, N-Triples, Turtle	N-triples, Turtle
Vocabularies reused in ontology	Web Annotation, Basic Geo	EOP, DCAT, GMD, GML, Atom, ...	GeoSPARQL, OWL-Time, SOSA, DCAT, PROV-O
Procedure	<ul style="list-style-type: none"> § Data collection § Ontology development § Data integration § Semantic reasoning 	<ul style="list-style-type: none"> § Dataset/metadata selection § Mapping rules development § Data conversion and integration 	<ul style="list-style-type: none"> § Dataset selection § Ontology development § Data integration § Semantic search
Triple store	GraphDB	Strabon	Strabon
Service language	Java	Python	Python
Semantic search/querying	SPARQL	SPARQL, GeoSPARQL	SPARQL, GeoSPARQL
Use of Docker	Yes	Yes	Yes

Document name:	D2.6 Semantic Search v2			Page:	51 of 52
Reference:	D2.6	Dissemination:	PU	Version:	2.2
				Status:	Final

Currently, CERTH has sent IRIT a sample of collected tweets. We are examining the possibility of exploiting collected tweets either through data integration or as linked data. Since tweet datasets have both spatial and temporal dimensions, we can demonstrate one more time the feasibility of our approach which proposed to link entities by their spatio-temporal relations.

CERTH will start the data crawling process for tweets related to the Nouvelle Aquitaine region to make the dataset exploitable and demonstrate the collaboration on the vineyard use case.

Document name:	D2.6 Semantic Search v2				Page:	52 of 52	
Reference:	D2.6	Dissemination:	PU	Version:	2.2	Status:	Final