Linking Earth Observation data with ground truth from the Open Web

Geert Koster[a], A. Cornelis Valk*[a], Ioannis N. Athanasiadis[b]
[a]NEO BV, Amersfoort, The Netherlands; [b]Wageningen University and Research, Wageningen, The Netherlands

ABSTRACT

This paper presents a system for linking Earth Observation with open Web data, into a Linked Open Data architecture. The architecture has two components, one for extracting signals from Earth Observation data, and another for harvesting web sources. Both are linked with spatial objects. Web scraped data, either from APIs or crowd-sourced websites are geo-referenced and thematically annotated with standard vocabularies. The architecture has been demonstrated in two case studies, one for building permits and another for crowd-sourced observations of invasive aquatic plants.

Keywords: Earth Observation, Linked Open Data, Semantic Web, GeoSPARQL, Building Permits, Invase Aquatic Plants

1. INTRODUCTION

Earth Observation (EO) data is inherently geospatial. Commonly data is structured as multidimensional arrays, including at least two spatial dimensions. With metadata accurately describing geolocation, this data can be accurately overlayed, compared and linked with other geospatial data such as digital topographic maps. EO data is often used to derive information about the state, nature, or composition of objects on the ground that are already known and mapped. New or disappeared objects (e.g., buildings, roads, trees, …) can be identified with limited effort using this approach.

However, information derived from EO data alone is not able to provide the required detail in all cases and may be more or less accurate depending on e.g., the EO data spatial resolution, geolocation accuracy, atmospheric conditions, season, or sun angles. Also, some information simply cannot be derived directly from EO data, such as number of rooms in a house with a roof. Other geospatial data is often used to augment the information derived from EO data, but documented trials to use non-geospatial data for this purpose are rare[1]. Depending on the use case, there are many non-geospatial datasets that could be valuable for either cross-checking information derived from EO data or enriching the information with further attributes. Nowadays, there are several such sources available on the Web as web services, crowd-sourced datasets, social media, or webpages. How to link these sources with EO data still is a challenge.

In this paper, we examine how to augment EO data with open Web data sources that can be used as ground truth and align EO with Web data using Linked Open Data technologies. We use two case studies to demonstrate our approach: The first one focusses on linking EO with structured web data, in this case: building permits issued by the Dutch authorities. The second case study focusses on linking unstructured crowd-sourced data, specifically observations of invasive water vegetation species.

The rest of the paper is structured as follows. Section 2 offers some background on geo-spatial Linked Data and details the two case studies. Section 3 presents our approach and architecture. Section 4 reports on our implementation, demonstrating the two case studies, and Section 5 discusses our findings and conclusions.

* cornelis.valk@neo.nl; phone +31 33 2100 700; neo.nl

Image and Signal Processing for Remote Sensing XXIX, edited by Lorenzo Bruzzone,
Francesca Bovolo, Proc. of SPIE Vol. 12733, 127330N · © 2023 SPIE
0277-786X · doi: 10.1117/12.2680216

2. BACKGROUND

2.1 Geo-spatial Linked Data

Linked Data and Semantic Web technologies may assist in aligning and linking EO data with each other, but also other kind types of geo-information, using the Resource Description Framework (RDF) for interconnecting data. W3C has proposed standards for representing and querying geospatial data in RDF allowing for topological querying in RDF with GeoSPARQL[1,2]. Such vocabularies and querying languages have been adopted by OpenStreetMaps, DBpedia, and Ordinance Services across the world. At the same time there have not been that many applications reported in the literature that demonstrated the use of Semantic Web and Linked Data technologies in linking EO datasets with other geo-information sources. Inspiring examples of such use are related to EO for crop type classification[3], wildfire monitoring[4] and wildlife modelling[5].

Combining and linking EO with open web data comes with many challenges. The first challenge is already in the storage of the data; both datatypes are stored differently. Geospatial data is usually stored as multidimensional arrays (for raster data) or in spatial relational databases (for vector data). Linked data stores are traditionally organized in triple stores because of its ability to store relations and flexibility in schema definition. Open web data is typically stored as HTML webpages, or XML webservices. Actual linking is difficult because big web data is very (semantically) heterogenous. People use many different words to describe the same thing: It is hard to know to which mapped geospatial object any bit of internet information corresponds. Therefore, geo-referencing open Web resources is a major challenge when harvesting data from the Web. Much of the existing literature tackles the challenge of linking heterogeneous datatypes by translating everything into common ontologies/vocabularies, which can then be linked purely through semantics[6, 7, 8].

2.2 Introduction to Case Studies

Case 1: Structured data (building permits)
Earth observation data is highly suitable and therefore oftentimes used for detecting changes on objects on the earth surface. Buildings are one of the most important object types monitored for changes by NEO. Information about building changes can be used to update maps, monitor construction activities, locate illegal constructions, etc.

Within this case study we attempt to link building change signals detected from EO data, with information about building permits. This is valuable for municipality workers as it enables them to detect illegal building activities, but also to know when exactly building activities start after a permit issue to plan inspection visits.

Information about building permits (messages about building permit requests, withdrawals, refusals, extensions, and issues) are made available by the Dutch authorities to the public through a restful API. These messages are semi-structured following predefined XML schemas. However, geo-information is not explicit in this schema.

Case 2: Unstructured data (crowd-sourced observations of invasive water vegetation species)
Whereas it is easy to detect vegetation growth using Earth Observation, it is not always possible to classify the exact type of vegetation, especially when the combination of the spatial, temporal, and spectral resolution is insufficient for the task. Supplementary information, such as human field observations can be an asset. As part of this case study, we link crowd-sourced field observations of invasive water plants to EO-based water vegetation detections system. Such a system allows water boards to perform targeted maintenance and to have insights into the spatial and temporal distribution of invasive water vegetation species in their waterways.

The crowd sourced observations of invasive water species are scraped from a website on which nature-enthusiasts can register natural field observations which may be to anyone's interest. This data source is by its nature less structured and reliable. Not all observations are geo-referenced, and anyone can register an observation and not all registered observations provide the same level of detail and accuracy.

3. AN ARCHITECTURE FOR LINKING EO WITH OPEN WEB DATA

In this work we have designed a hybrid approach for linking EO data with open Web data sources. Instead of storing both EO and geo-data scrapped from the web in a common infrastructure, we kept the two sources separated: Geospatial EO data are handled separately from the scraped web (big) data and are linked semantically through an intermediate object database. Our system architecture is shown in Figure 1.

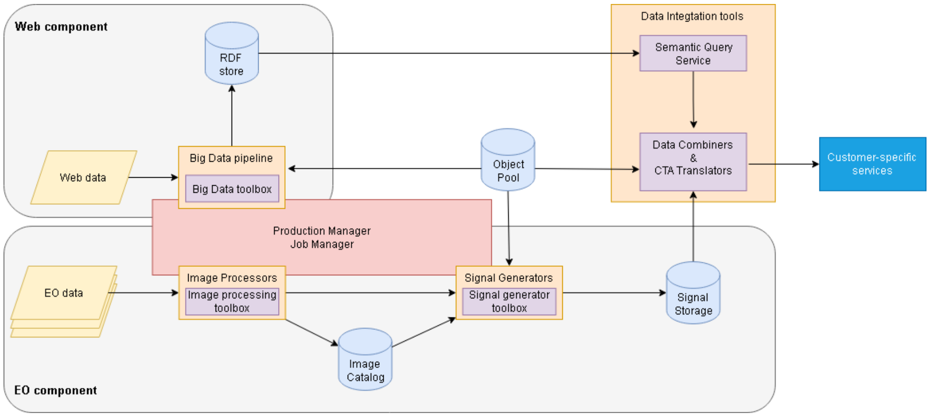


Figure 1. The proposed architecture, showing the EO component (below) and the open Web scraping and annotating component (above).

The EO component of our infrastructure is a traditional EO data processing system. Inside, EO data is pre-processed to a level on which analysis and comparison to other geospatial data is possible and stored in the 'Image Catalog.' For many known geospatial objects, stored in the 'Object Pool,' information is derived from EO data using 'Signal Generators' and stored in the 'Signal Storage.' These Signals are further refined to use case specific and customer-specific products. The geospatial objects that are monitored are derived from open national registrations, created from EO data, or supplied by individual customers. EO data, objects and signals are stored in relational spatial databases, with rigid schemas.

Web data is scraped from the web by source-specific 'harvesters'. A 'big data toolbox' has been developed with generic tools for filtering and translating data. These tools geo-reference the open data and define associations with existing Objects from the Object pool. Also, the content is annotated using domain-specific vocabularies, and provenance information. The data is stored into a geo-referenced RDF triple store, using predefined formalized ontologies. In this way, web-scaped data can easily be queried using GeoSPARQL, and linked to existing Objects.

Linkage of the web data to EO geospatial data is done by using the object pool as a midway point of reference. Whilst our geospatial EO-derived signals are linked to objects purely spatially, the web data in the RDF store can be linked to these objects semantically as well. For example, using building identifiers instead of location.

By handling information derived from each data source (EO-derived data and web data) separately, for each data source we can use tools which are best suited for their purpose. RDF triple stores are highly suited for storing web data. It allows for storing relations and does not need rigid schemas. Furthermore, RDF triple stores can use inference for inferring new information out of the existing relations and handle powerful semantic queries. Multidimensional arrays and spatial relational databases are very mature technologies highly suitable for geospatial data. Efficient spatial indexing methods are available for quick spatial querying of data in spatial databases and effective compression algorithms allow for cost-efficient storage of raster data.

Regarding the vocabulary used for annotating harvested web data, we selected four existing general ontologies to capture properties that we commonly like to store (Table 1). Dublin Core is used for general annotation of metadata, such as creator, description, and source. PROV is used for documenting where, when, and how data came to be. GeoSPARQL is used for geospatial data and ISA Programme Location Core is used for address information.

Table 1: Reused vocabulary reused for annotating harvested web data.

Name	Prefix	URL
Dublin Core	dc	http://purl.org/dc/elements/1.1/
PROV	prov	https://www.w3.org/ns/prov#
GeoSPARQL	geo	http://www.opengis.net/ont/geosparql#
ISA Location Core	locn	http://www.w3.org/ns/locn#

To connect to the spatial objects that we have available in the object pool, we introduce `neo:NEOObject` classes as subclass of `prov:Entity`. A `NEOObject` has exactly one `geo:Geometry` and exactly one `neo:NEOID`. `NEOID` is a string that corresponds with the unique identifier of the same object in the object pool.

Most geospatial information that does not have a formal geometric description is linked to addresses. We introduce data properties that together with `locn:PostCode` provide a unique identification of any address in the Netherlands: plaats, straat, huisnummer and huisnummertoevoeging.

The distinction between originally reported and inferred or predicted information, especially concerning spatial information is important. While reported information may be regarded as factual, inferred, or predicted information will often be less complete and correct. To support this distinction, we introduce different object properties `reportedLocation`, `reportedAddress`, `scrapedLocation` and `scrapedAddress`. We use a NEO `prov:Organisation` individual and `prov:SoftwareAgent` individuals to keep metadata on the software used for harvesting.

The actual combination of EO-derived signals and open Web data is possible when data is extracted from the system. When a user requests signals for his/her objects for a certain time range, EO-signals and big web data linked to his/her signals are returned. During this extraction phase, additional information can be inferred to create call-to-actions. For example, the EO-change signal observed can be matched with the construction activity for which a building permit has been requested.

4. DEMONSTRATION

4.1 Structured Data (building permits)

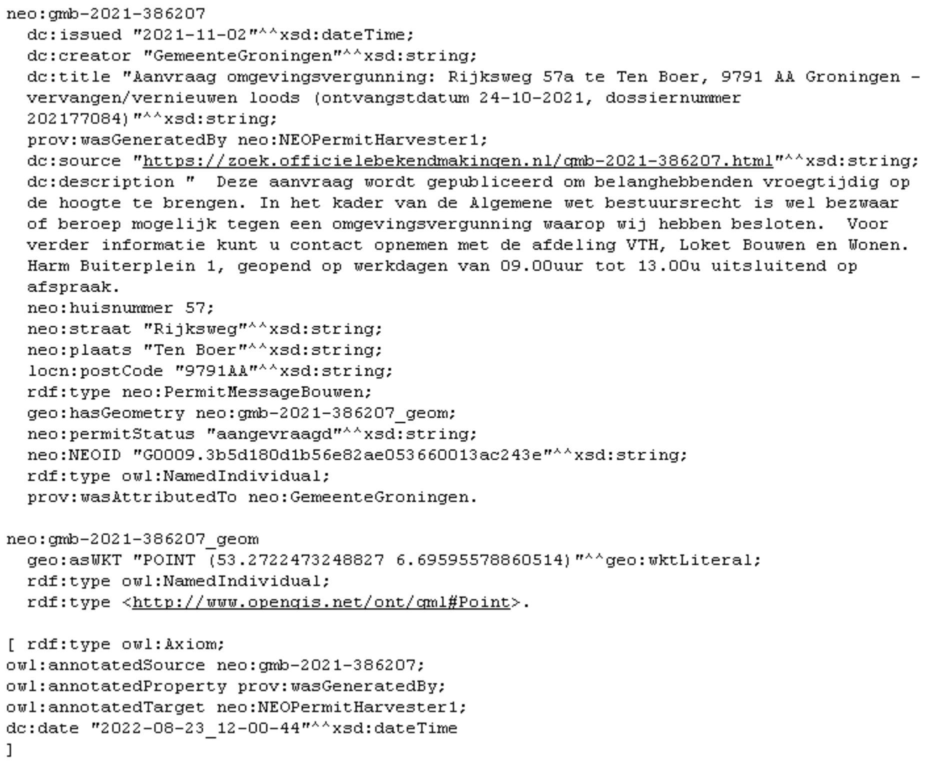
The vocabulary for the building permits case study builds on top of the PROV ontology, to store building permit data into the RDF store. We introduced a `NEOBuildingObject` class (subclass of `neo:NEOObject`), several `PermitActivity` classes forming a typology of different permit activities (subclass of `prov:Activity`), a typology of `PermitMessage` classes (subclass of `prov:Entity`) and with the Dutch national, provincial, and municipal entities that issue permits as individuals of the class `prov:Organisation`. The object property `requestedForActivity` (subclass of `prov:wasUsedBy`) is added to connect a `PermitMessage` to a `PermitActivity`.

`PermitMessages` are the harvested entities. A `PermitMessage` has the following properties:

- `geo:hasGeometry` exactly one `geo:Geometry`
- may have one or more `neo:scrapedAddress`
- may have one or more `neo:reportedAddress`
- `prov:wasAttributedTo` at least one `prov:Organisation`
- `prov:wasAttributedTo` at least one `prov:SoftwareAgent`
- `neo:requestedForActivity` at least one `neo:PermitActivity`

`PermitMessages` are further annotated using `permitStatus`, `dc:title`, `dc:description`, `dc:creator`, `dc:identifier`, `dc:issued`, `dc:source`, and `locn:fullAddress`.

The building permit messages are harvested, filtered, and translated by the developed *building permit message harvester*, which is part of the big data pipe-line. The messages are semantically linked to buildings in the object pool and stored in the RDF triple store. Linkage is done based on approximate location and address, but can easily be expanded to other building properties such as height and use in cases such information is included in the permit message. Figure 2 shows an example of a translated building permit message in Turtle format.



```
neo:gmb-2021-386207
  dc:issued "2021-11-02"^^xsd:dateTime;
  dc:creator "GemeenteGroningen"^^xsd:string;
  dc:title "Aanvraag omgevingsvergunning: Rijksweg 57a te Ten Boer, 9791 AA Groningen -
  vervangen/vernieuwen loods (ontvangstdatum 24-10-2021, dossiernummer
  202177084)"^^xsd:string;
  prov:wasGeneratedBy neo:NEOPermitHarvester1;
  dc:source "https://zoek.officielebekendmakingen.nl/gmb-2021-386207.html"^^xsd:string;
  dc:description "  Deze aanvraag wordt gepubliceerd om belanghebbenden vroegtijdig op
  de hoogte te brengen. In het kader van de Algemene wet bestuursrecht is wel bezwaar
  of beroep mogelijk tegen een omgevingsvergunning waarop wij hebben besloten.  Voor
  verder informatie kunt u contact opnemen met de afdeling VTH, Loket Bouwen en Wonen.
  Harm Buiterplein 1, geopend op werkdagen van 09.00uur tot 13.00u uitsluitend op
  afspraak.
  neo:huisnummer 57;
  neo:straat "Rijksweg"^^xsd:string;
  neo:plaats "Ten Boer"^^xsd:string;
  locn:postCode "9791AA"^^xsd:string;
  rdf:type neo:PermitMessageBouwen;
  geo:hasGeometry neo:gmb-2021-386207_geom;
  neo:permitStatus "aangevraagd"^^xsd:string;
  neo:NEOID "G0009.3b5d180d1b56e82ae053660013ac243e"^^xsd:string;
  rdf:type owl:NamedIndividual;
  prov:wasAttributedTo neo:GemeenteGroningen.

neo:gmb-2021-386207_geom
  geo:asWKT "POINT (53.2722473248827 6.69595578860514)"^^geo:wktLiteral;
  rdf:type owl:NamedIndividual;
  rdf:type <http://www.opengis.net/ont/gml#Point>.

[ rdf:type owl:Axiom;
owl:annotatedSource neo:gmb-2021-386207;
owl:annotatedProperty prov:wasGeneratedBy;
owl:annotatedTarget neo:NEOPermitHarvester1;
dc:date "2022-08-23_12-00-44"^^xsd:dateTime
]
```

Figure 2. A sample building permit message in Turtle format, as extracted by the *building permit message harvester*. The original message has been annotated with prove-nance, location, and permit message type metadata.

The building change signals are created by *building change signal generators* which make use of aerial/satellite images to detect changes on buildings between two time periods. These change signals are stored as vectors (specifying the size and location of the change) in the signal storage database. Each change signal has been linked (spatially) to building objects in the object pool.

When a client requests a delivery of building changes for a specific area between a certain start and end-date, the EO based building change signals linked to build-ing objects in that area are extracted from the signal storage. These signals are complemented with building permit information from the RDF triple store extract-ed using the query service. This is valuable for municipality workers as it enables them to detect illegal building activities but also to know when exactly building activities start after a permit issue to plan inspection visits.

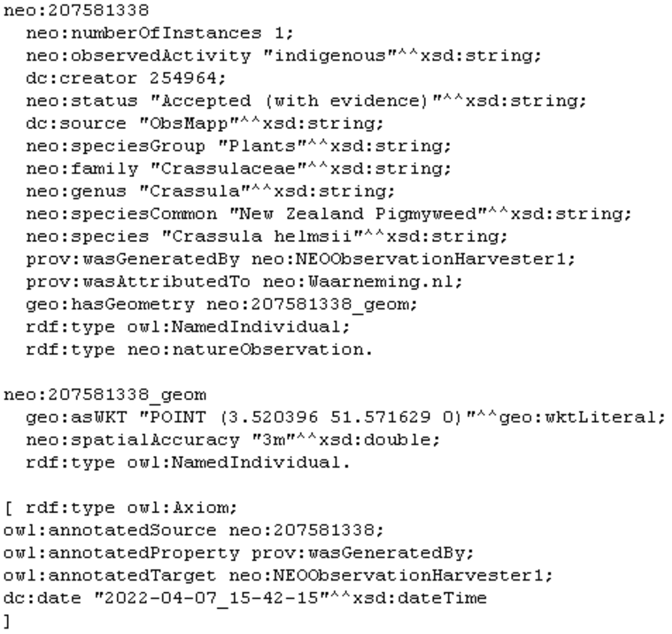
4.2 Unstructured Data (crowd sourced observations of invasive water vegetation species)

To support this use case, we introduce a single `natureObservation` class as `prov:Entity` subclass with some object properties, like the `PermitMessage`:

- `geo:hasGeometry` exactly one `geo:Geometry`
- `prov:wasAttributedTo` at least one `prov:Organisation`
- `prov:wasAttributedTo` at least one `prov:SoftwareAgent`

Most of the information available for these field observations are stored as annotations and properties: `dc:creator`, `dc:identifier`, `dc:source`, `dc:date`, `status`, `family`, `genus`, `species`, `speciesGroup`, `speciesCommon`, `observedActivity`, `NumberOfInstances`.

We have developed the *invasive water vegetation species harvester* as part of the big data pipeline. It harvests invasive water species observations from a Dutch website for crowd source nature observations. The harvester translates these scraped observations into RDF, finds the linked water objects from the object pool (retrieving their object-IDs) and saves them into the RDF triple store. Linkage is in this case done based on the location of the observation. Figure 3 shows an example of a translated observation in Turtle format.



```
neo:207581338
  neo:numberOfInstances 1;
  neo:observedActivity "indigenous"^^xsd:string;
  dc:creator 254964;
  neo:status "Accepted (with evidence)"^^xsd:string;
  dc:source "ObsMapp"^^xsd:string;
  neo:speciesGroup "Plants"^^xsd:string;
  neo:family "Crassulaceae"^^xsd:string;
  neo:genus "Crassula"^^xsd:string;
  neo:speciesCommon "New Zealand Pigmyweed"^^xsd:string;
  neo:species "Crassula helmsii"^^xsd:string;
  prov:wasGeneratedBy neo:NEOObservationHarvester1;
  prov:wasAttributedTo neo:Waarneming.nl;
  geo:hasGeometry neo:207581338_geom;
  rdf:type owl:NamedIndividual;
  rdf:type neo:natureObservation.

neo:207581338_geom
  geo:asWKT "POINT (3.520396 51.571629 0)"^^geo:wktLiteral;
  neo:spatialAccuracy "3m"^^xsd:double;
  rdf:type owl:NamedIndividual.

[ rdf:type owl:Axiom;
owl:annotatedSource neo:207581338;
owl:annotatedProperty prov:wasGeneratedBy;
owl:annotatedTarget neo:NEOObservationHarvester1;
dc:date "2022-04-07_15-42-15"^^xsd:dateTime
]
```

Figure 3. A sample observation in Turtle format, as extracted by the invasive water vegeta-tion species harvester. The original message has been annotated with prove-nance, location, and species type metadata.

Water vegetation signal generators make use of aerial/satellite images to detect water vegetation within a specific area between a start and end-date. These sig-nals are stored as vectors in the signal storage database (specifying the location and extent of vegetation growth) linked spatially to water objects in object pool.

When a client requests a delivery of locations of invasive water vegetation species for a specific area between a certain start and end-date, water vegetation signals are extracted from the signal storage database whose water objects also have multiple invasive water vegetation species observations in in the RDF triple store. This information allows water boards to perform targeted maintenance and to have insights into the spatial and temporal distribution of invasive water vegetation species in their waterways.

5. CONCLUSION AND DISCUSSION

The implementation of our approach for linking Web with EO data has been suc-cessful for both case studies. Storing EO-derived geospatial data separately from (big) data harvested from the web allows each data type to be stored using the technology that fits it best. Using traditional geospatial storage methods for EO-derived data such as multidimensional arrays (for raster data) and spatial rela-tional databases (for vector data) allows us to benefit from efficient and mature spatial indexing algorithms (resulting to fast spatial querying) and effective compression. Using an RDF triple store and formalizing ontologies for data scraped from the web allows for standardized storage of this data as well as efficient se-mantic linkages to other scraped web data and the possibility to perform semantic reasoning and inference. The 'object pool' is the intermediate layer which allows both data types to be linked to one another. Objects can be linked spatially to EO-derived geospatial data and semantically to harvested web data. The linked open data nature of RDF allows to further extend web-scrapped data with new Objects in the future. It even enables the potential of using OWL axioms to infer classes or properties that may are specified at a later stage.

This approach differs from that chosen in other literature (e.g. [6,7,8]) where the challenge of linking heterogeneous datatypes is solved by translating everything into RDF which can be linked semantically. Even though this might allow for better integration of data, we have found that storing EO-derived geospatial within a triple store came with several implementation drawbacks. The main drawback being the inferior performance of GeoSPARQL in spatial querying compared to e.g. POSTGIS.

A point of discussion is whether we have fully used the potential of semantics within our chosen case studies. The web data from our case studies could be linked to EO data easily, and it can be argued that this could also be done without using semantics. Furthermore, no use has been made of semantic reasoning or inference, which is one of the additional benefits of using semantics. However, as soon as the linkages become more complicated, semantic integration does seem to be the most flexible option. As an example, twitter messages about construction activities could also be linked to building objects in the future but might have to consider the height, size, or type of a building for linking because the spatial information is insufficient. Linking tree felling permits tree objects might also have to consider the type of tree, again because the spatial information might not always be sufficient. In such more complex cases, Linked Open Data can demonstrate their full potential.

REFERENCES

[1] Battle, R. & Kolas, D. GeoSPARQL: Enabling a Geospatial Semantic Web. *Semantic Web Journal* **3**, 355–370 (2011).

[2] Car, N. J. & Homburg, T. GeoSPARQL 1.1: Motivations, Details and Applications of the Decadal Update to the Most Important Geospatial LOD Standard. ISPRS International Journal of Geo-Information (2022).

[3] Rousi, M. et al. Semantically Enriched Crop Type Classification and Linked Earth Observation Data to Support the Common Agricultural Policy Monitoring. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing 14, 529–552 (2021).

[4] Kyzirakos, K. et al. Wildfire monitoring using satellite images, ontologies and linked geospatial data. Journal of Web Semantics 24, 18–26 (2014).

[5] Athanasiadis, I. N., Villa, F., Examiliotou, G., Iliopoulos, Y. & Mertzanis, Y. Towards a semantic framework for wildlife modeling. Proc. 28th Int'l Conf. Informatics for Environmental Protection (Enviroinfo 2014), pg. 287-292, 2014, Univ of Oldenburg BIS Verlag (2014).

[6] Yunqiang Zhu, A-Xing Zhu, Jia Song, Jie Yang, Min Feng, Kai Sun, Jingqu Zhang, Zhiwei Hou & Hongwei Zhao (2017) Multidimensional and quantitative interlinking approach for Linked Geospatial Data, International Journal of Digital Earth, 10:9, 923-943, DOI: 10.1080/17538947.2016.126604

[7] Kamran Munir, Mohammed Odeh, and Richard McClatchey. 2009. Managing the mappings between domain ontologies and database schemas when formulating relational queries. In Proceedings of the 2009 International Database Engineering & Applications Symposium (IDEAS '09). Association for Computing Machinery, New York, NY, USA, 131–141. https://doi.org/10.1145/1620432.1620446

[8] Ulutaş Karakol, Deniztan & Kara, Gülten & Yilmaz, Cemre & Cömert, Çetin. (2018). Semantic Linking Spatial RDF Data to Web Data Sources. XLII-4. 639-645. 10.5194/isprs-archives-XLII-4-639-2018.