Semantic Enrichment of Location-based Business Intelligence using Virtual Knowledge Graphs

Arka Ghosh[1 [0000-0003-3789-0900]], Benjamin Cogrel[2 [0000-0002-7566-4077]], Albulen Pano[3 [0000-0002-0905-9004]], and Diego Calvanese[2,3 [0000-0001-5174-9693]]

[1] Department of Computing Science, Umeå University, Sweden
arka.ghosh@umu.se
[2] Ontopic s.r.l., Italy
benjamin.cogrel@ontopic.ai
[3] Faculty of Engineering, Free University of Bozen-Bolzano, Italy
{albulen.pano,diego.calvanese}@unibz.it

Abstract. Practical business intelligence (BI) over heterogeneous data sources, including relational, vector (e.g., geometries), and raster data (e.g., satellite images), requires interconnecting the data in a semantically coherent manner so that they can be queried and analysed in a uniform way to extract business insights that aid informed decision-making. The *Virtual Knowledge Graph* (VKG) paradigm addresses the issue of data heterogeneity by relying on an ontology to expose domain knowledge and connecting it via declarative mappings to the underlying data sources. The VKG paradigm has thus far concentrated mainly on relational data. At the same time, only a few works address its combination with vector and raster data, which is especially important within Geospatial Business Intelligence (GeoBI), such as in the context of earth observation (EO), the integration of geographic information systems, and building information modelling (GIS/BIM). However, such a combination presents a considerable challenge for conventional DBs to manage or query efficiently, due to their multidimensional and complex characteristics. In this paper, we address this problem by extending the VKG paradigm to enable interface with specialised array database management systems. We then demonstrate how to utilise BI tools to derive location-based business insights, leveraging both standard semantic technologies and a novel technology that enables a knowledge graph to be accessed via a traditional SQL interface.

1 Introduction

Data-driven businesses seek assurance that their information is streamlined, production-ready, and trustworthy before using it to understand key business factors and make informed decisions at any given time. As corporate decision-making increasingly depends on data, *Business Intelligence* (BI), which traditionally concentrated on analysing attributed numerical data in tabular format, is now rapidly including other forms of data into its scope, such as graph and array data [11]. Nowadays, data are being generated relentlessly from various resources in different representations at a massive scale, leading to bottlenecks in data management and processing, which gives rise to the notorious issue of data heterogeneity [1]. Effective BI over heterogeneous data sources (both structured and unstructured) requires interconnecting the data in a semantically coherent manner so that the data can be queried and analysed in a uniform way to extract business insights that aid informed decision-making [11]. However, current data management mechanisms do not natively support every data format; for instance, raster data represented in multidimensional arrays poses a significant challenge in the BI domain.

Virtual Knowledge Graphs (VKGs) [37], also known as *Ontology-Based Data Access* (OBDA), provides a flexible and efficient data integration paradigm for large-scale, richly structured relational data. In VKGs, end-users interact with a high-level, conceptual representation of the relevant domain of interest, presented as an *ontology*, which is connected to the data source through declarative *mapping* assertions. The ontology is typically expressed in the lightweight OWL 2 QL profile of the Web Ontology Language (OWL 2) [30], which has its formal counterpart in a Description Logic (DL) of the *DL-Lite* family [9]. The actual data are maintained in a relational data management system (RDBMS) but are *not* materialised at the conceptual level (hence the term “virtual”). The primary emphasis of research on VKGs has been on query answering, which involves computing the appropriate responses to a user query formulated over the ontology. This is achieved by utilising the ontology axioms and the mapping layer to retrieve pertinent data from the underlying data source. Such traditional VKG systems are inefficient at query answering over multidimensional raster data [5], and at integrating them with classic relational data and its geometrical variant *vector* data (e.g., *points, lines, polygons*). Issues regarding the integration of raster and vector data are addressed in the author’s recent work ONTORASTER [17]. This extended VKG framework+ facilitates the integration and query answering of both raster and relational data in their native storage format. Still, it has limited support for complex geometrical feature data (or vector data), such as island features (multi-polygons), enclaves, as well as publicly available vast geometrical vector datasets, such as *GeoNames* data, *OpenStreetMap* (OSM) data, and *CityGML* data. All of these are crucial for businesses to generate actionable location-based insights in the context of location-based services, urban planning, land-use classification, and other related applications. To further enrich location-based insights, it would be necessary to obtain information from query answering across a wide range of high-resolution geo-raster data (e.g., satellite imagery); however, our prior work [17] has only addressed this issue in a limited manner.

Integrating and uniformly querying these different types of data requires practical geospatial data management skills and extensive domain knowledge, which may not be readily available to business managers, analysts, or policymakers. Hence, it becomes difficult to resolve contemporary issues in big geo data applications, including Earth Observation (EO), Geographic Information System/Building Information Modelling (GIS/BIM) integration, and 3D/4D urban planning [8]. Moreover, it involves the visualisation of query results in the form of vectors and rasters to facilitate perpetual accessibility and reproducibility of geo-visual analytics, which can lead to the generation of location-based intelligence. In the literature, this issue has been addressed so far only in a somewhat limited way in the VKG settings and their application towards BI, which could benefit both the GIS and Semantic Web (SW) communities. A recent work [27] discusses business insight generation and policymaking by utilising multidimensional satellite imagery (or raster data) and machine learning to detect unplanned urbanisation, which contributes to the financial crisis and impedes economic growth. A prevalent method for accessing and analysing these diverse data sources is to develop ad-hoc scripts utilising Python, R, or MATLAB; however, this necessitates extensive topic knowledge and skill across many tools at various levels. Furthermore, whenever the data sources change, the entire script must be adjusted at a minimum or, in a worst-case scenario, completely rewritten. Most of these works mentioned in the literature are related to GIS/BIM utilising basic vector data; however, almost none employ raster data in VKG settings.

We present three key contributions by using the *Virtual Knowledge Graphs* (VKGs) paradigm to mitigate data heterogeneity from diverse data forms.

(1) We extend the open source ONTORASTER framework to support query answering over integrated data of more complex forms, thus making the framework better suited for GeoBI applications. Specifically, with our extensions, ONTORASTER now supports: *(i)* complex geometrical features, e.g., multi-polygons, polygons with holes, and their combination, *(ii)* additional public vector data crucial for business, such as GeoNames, OpenStreetMap, and CityGML data, with respective ontologies capturing the relevant concepts therein, *(iii)* multiple raster data of arbitrary resolutions and dimensions. We observe that such raster data is already correctly conceptualised by the raster ontology of [17].

(2) We provide a comprehensive geo-map-based visualisation of query results that exhibits at-a-glance location-based information derived from integrated raster+relational data. Thus, ONTORASTER aids business professionals in making informed decisions, hence enhancing geo-visual analytics within GeoBI.

(3) We provide methods to query ONTORASTER from standard BI tools (e.g., Power BI[4]), so that the query results coming from ONTORASTER can be converted into actionable insights. For this, we rely on a novel technology that we have developed, called *Semantic SQL-Interface*, which enables a KG to be accessed via a traditional JDBC-based SQL interface.

The updated ONTORASTER framework is available as open source on Github[5].

2 Geospatial Business Intelligence (GeoBI)

Geospatial Business Intelligence (GeoBI) integrates Geographic Information Systems (GIS) with established BI technologies [2], and is becoming a prominent driving force that empowers geographical location-based decision-making and product design in industrial organisations [20]. The majority of GeoBI applications are based on relational data, including vector data. The inclusion of raster data in business analytics is often overlooked due to its complex nature and the need for domain expertise.

Raster data is often referred to as a *data cube* in the BI domain since 1990 to describe OLAP (online analytical processing) cubes [33]. These cubes organise statistical measures (e.g., mean, variance, median) across multidimensional, sometimes hierarchical, data [18]. The concept evolved into spatial OLAP (SOLAP) with the inclusion

[4] https://www.microsoft.com/en-us/power-platform/products/power-bi

[5] https://github.com/aghoshpro/OntoRaster.git

of spatial dimensions and vector features [19]. A variant called MOLAP (multidimensional OLAP) stores data in multifaceted arrays rather than relational tables [21].

In Earth Observation (EO), the term *datacube* refers to large, analysis-ready raster datasets generated by satellite sensors such as Landsat and Sentinel. These EO datacubes facilitate the access, analysis, visualisation, and distribution of geospatial data [16]. While structurally similar to the typically sparse BI data cubes, EO data cubes are densely populated, limiting the applicability of OLAP methods in EO contexts [4]. This complexity can pose challenges for business analysts lacking GIS expertise, impeding location-based BI insights.

3 Semantic Technologies for Accessing Geospatial Data

Knowledge graphs (KG) are by now a well-established paradigm grounded in the Semantic Web for representing, retrieving, and integrating data from highly heterogeneous sources [23]. KG-based solutions often rely on an ontology that conceptualises a user-specific domain of interest (e.g., medical, geospatial, life sciences) and thus imparts meaning to the data. The instances of an ontology are represented as Resource Description Framework (RDF) triples and queried using the SPARQL query language. Laborie et al., [28] described two possible approaches to combine the semantic web and BI: *(i)* the analysis-oriented, *(ii)* the modelling-oriented approach. In the former case, a standard ETL (extract-transform-load) process is executed. A SPARQL query retrieves the data, and the results are then loaded. The latter entails conducting the analysis directly on the Linked Data without prior ETL. This strategy appears more effective, but it necessitates an advanced conceptual representation of data, which is our primary focus for our methodology.

Virtual Knowledge Graphs (VKGs). We consider here a setting where the domain ontology conceptualises the information stored in a set of existing, typically heterogeneous (e.g., relational, tabular, or tree-structured) data sources and is linked to the sources through semantic mappings that expose the underlying data as a KG. Typically, the KG is not materialised but kept *virtual*, i.e., the relevant portions necessary to answer a query expressed over the ontology are generated at query time, hence the name of *VKG* given to this framework. Formally, a *VKG specification* $\mathcal{P} = (\mathcal{O}, \mathcal{M}, \mathcal{S})$ consists of *(i)* an *ontology* $\mathcal{O}$ expressed as a TBox in the lightweight ontology language OWL 2 QL, *(ii)* a relational *data source schema* $\mathcal{S}$, and *(iii)* a declarative mapping $\mathcal{M}$ that associates to each element (i.e., class or property) in $\mathcal{O}$ a (SQL) query over $\mathcal{S}$, specifying how to (virtually) populate that element through the data retrieved from the source. . In traditional VKGs, the mapping $\mathcal{M}$ consists of a set of R2RML *mapping assertions* of the form

$$Q_{sql}(\boldsymbol{x}) \rightsquigarrow E(\boldsymbol{f}(\boldsymbol{x})),$$

where $Q_{sql}(\boldsymbol{x})$ is a SQL query (called *source query*) over $\mathcal{S}$ with answer variables $\boldsymbol{x}$, and the *target* $E(\boldsymbol{f}(\boldsymbol{x}))$ consists of a class or property E of $\mathcal{O}$, and a set $\boldsymbol{f}(\boldsymbol{x})$ of so-called *IRI-templates* applied to the variables in $\boldsymbol{x}$. Each IRI-template is a function that constructs an ontology literal or an IRI identifying an ontology object, from the values in each answer to $Q_{sql}(\boldsymbol{x})$ instantiating $\boldsymbol{x}$. Then, a *VKG instance* is a pair

$(\mathcal{P}, \mathcal{D}^{rel})$, where $\mathcal{D}^{rel}$ is a relational database instance conforming to $\mathcal{S}$. By "applying" the mapping assertions in $\mathcal{M}$ to $\mathcal{D}^{rel}$, i.e., by evaluating each source query $Q_{sql}(\boldsymbol{x})$ over $\mathcal{D}^{rel}$ and using the returned answers to instantiate the target $E(\boldsymbol{f}(\boldsymbol{x}))$, one obtains a KG $\mathcal{M}(\mathcal{D}^{rel})$, which, however, is kept 'virtual'. Semantic queries formulated in SPARQL are posed over $\mathcal{O}$ and are answered by accessing the relational data $\mathcal{D}^{rel}$ through the mapping $\mathcal{M}$. Specifically, given a SPARQL query q over a VKG instance $\mathcal{J} = (\mathcal{P}, \mathcal{D}^{rel})$, we are interested in the *certain answers* to q over $\mathcal{J}$, denoted $\mathsf{cert}(q, \mathcal{J})$, which are the answers obtained by evaluating q over the knowledge base $(\mathcal{O}, \mathcal{M}(\mathcal{D}^{rel}))$ under the OWL 2 QL entailment regime. Actual VKG systems, such as ONTOP [38], avoid costly materialisation of the KG $\mathcal{M}(\mathcal{D}^{rel})$ and its storage in a triple store and rather translate the SPARQL query into a relational SQL query that is directly evaluated by the underlying RDBMS (e.g., PostgreSQL), thus ensuring also freshness of query answers concerning source updates.

Geospatial Knowledge Graphs (GeoKGs). In this research, we consider GeoKGs, which are KGs where geospatial information is modelled using geo-coordinates that capture, on the one hand, geometric regions delimited by polygons (i.e., vector data) and, on the other hand, values from a continuous domain (e.g., temperature, precipitation) associated to all points in a specified region (i.e., raster data) [14,17]. GeoKGs are often converted from geospatial vector data stored in spatial databases (PostGIS/PostgreSQL) or other popular formats like shapefile, CSV, and GeoJSON. VKGs provide a systematic method for such conversion by relying on the GeoSPARQL ontology [3], which has been designed by the *Open Geospatial Consortium* (OGC). The GeoSPARQL ontology utilises a representation of vector geometry literals compliant with *Geography Markup Language* (GML) and *Well-Known Text* (WKT), and relies on a vocabulary for topological relationships and ontologies for qualitative reasoning. The GeoSPARQL language itself extends SPARQL with geometrical functionalities to represent and query geo-enriched KGs. However, it still suffers from several limitations that restrict its effective use in practical settings, specifically when raster data plays a prominent role. The GeoSPARQL+ framework [24] provides an enhanced version of the GeoSPARQL vocabulary, query language, and ontology, extending RDF to support geospatial raster data. However, it has limited support for complex multidimensional raster and multi-polygonal vector data. Moreover, it relies on materialising geospatial data as RDF triples, resulting in very large KGs with potential efficiency issues. LinkedGeoData (LGD) [32] is one of the well-known GeoKG projects that relies on the VKG approach to expose OpenStreetMap (OSM)[6] data as RDF knowledge graphs. YAGO2geo [25] is another popular GeoKG that claims to be the largest currently, containing detailed geometries, i.e., 700,000+ polygons and 3.8 million lines, taken from the *Global Administrative Data and Maps* (GADM)[7] and OSM data.

The World Wide Web Consortium (W3C) has standardised the representation of statistical data cubes using the RDF *Data Cube Vocabulary* [34]; however, it provides minimal support for representing the multidimensional model. These limitations are addressed by QB4OLAP [35], a vocabulary for BI over Linked Data that facilitates the

[6] https://www.openstreetmap.org/

[7] https://gadm.org/index.html

representation of OLAP cubes in RDF and provides standard OLAP operations (including roll up, slice, dice, and drill-across) through SPARQL queries directly over RDF representations. Still, it does not support raster functionalities to query complex EO data cubes, which constitutes an important missing feature of location-based BI. Notably, a semantic data cube system, named *Plato* [7], has been proposed within the EU H2020 project DeepCube[8], which relies on a geospatial extension of the VKG system ONTOP [6] and uses PostgreSQL Foreign Data Wrappers.

The* ONTORASTER *framework. The recently proposed ONTORASTER [17] framework has been developed to overcome the limitations mentioned above by supporting on-the-fly query answering on multidimensional raster data integrated with relational data (including vector data). Specifically, it builds on the VKG framework and extends it as follows: *(i)* For the query language, it relies on RasSPARQL, which extends SPARQL with novel raster functionalities. *(ii)* It utilises an extended variant of the VKG engine ONTOP to translate RasSPARQL queries containing raster functions into suitable queries with corresponding database functions. *(iii)* Such queries are processed by a query transformation system implemented within PostgreSQL, which uses *stored procedures* in PL/Python to issue queries to an array DBMS and combine the results with relational data (including vector data).

Specifically, ONTORASTER utilises Rasdaman, a domain-agnostic array DBMS that exclusively accepts array indices or grid coordinates. However, it does not natively accommodate any domain-specific coordinates, such as geographic coordinates (i.e., longitude and latitude), which are the building blocks of vector geometries. Therefore, ONTORASTER provides the `geo2grid_coords()` function, implemented in PL/Python inside PostgreSQL, which translates geographic geometries into grid-based representations (grid geometries) by considering the respective coordinate reference system (CRS) of both vector and raster data. At its core, the `geo2grid_coords()` function manages the coordinate transformation process by generating an affine transformation matrix [36] derived from the metadata of the specified raster data in the user query, encompassing the minimum longitude, maximum latitude, and resolution values for both x and y axes. This matrix is then used to convert geographic coordinates to grid coordinates, with the results being rounded to integer values to ensure compatibility with a grid-based representation suitable for Rasdaman's query language *rasql*. The transformation process maintains the spatial relationships between points while adapting them to the grid system.

4 Enhancing the OntoRaster Framework

In this work, we have extended the ONTORASTER framework along different directions, which we discuss below.

4.1 Supporting Complex Geometrical Features

ONTORASTER still lacks support for complex geometries, such as polygons with holes, and multi-polygons, which are common in real-world GeoBI use cases. In the case

[8] https://deepcube-h2020.eu/

of simple polygons, OntoRaster performs a direct transformation of the polygon's exterior ring coordinates, converting them from geographic to grid space. When dealing with polygons containing holes, both the exterior ring and all interior rings (or holes) are processed separately, and then merged in grid space. For multi-polygons, this function employs a sophisticated approach that allows for processing each constituent polygon individually, transforming its coordinates into grid space. The function then attempts to merge these transformed polygons using Shapely's `unary_union` operation. Shapely[9] is a Python library for manipulating and analysing planar geometric objects using the widely distributed open-source geometry library GEOS (the engine of PostGIS), which conforms to OGC's simple feature access specification [22]. Suppose the merging operation fails for any reason, such as tiny constituent polygons (e.g., a tiny island). In that case, the function falls back to creating a new multi-polygon by combining the remaining transformed polygons. Error handling and edge cases are also implemented in `geo2grid_coords()`, including checks for invalid geometries (e.g., those with zero area) or invalid WKT representations, and specific handling for unsupported geometry types. It also manages potential failures in the transformation process and polygon merging operations, ensuring robust operation even for complex input geometries.

4.2 Extension to Additional Heterogeneous Data and Ontologies

The majority of GeoBI applications rely on relational data [10], and the inclusion of raster data in their native data structure in GeoBI analytics is often neglected due to its complex nature and the need for specialised knowledge. To mitigate this challenge, we are extending the integration and query processing capabilities of OntoRaster by incorporating multiple types of raster data having arbitrary resolutions and dimensions, along with large-scale public vector datasets essential for the business domain, such as GeoNames, OpenStreetMap, and CityGML 3D building data, along with the relevant ontologies encapsulating the pertinent concepts.

The area of interest (AOI) selected for this study is Munich, the capital of Bavaria, Germany, and its surrounding functional urban area (approx. 311.20 km^2), i.e., densely populated city centres integrated with the surrounding labour market (commuting zone through high travel-to-work flow, as defined by EU-OECD [12]). Fig. 1 displays different types of geospatial data about Munich (each small box in the upper right corner provides a zoomed-in view, to show details about the actual data). Table 1 depicts all the information about the data. We are considering 25 districts (Fig. 1a) and 105 subdistricts (Fig. 1b) of Munich as our primary vector data. Additionally, *OpenStreetMap* (OSM) data from Geofabrik Server[10] for Oberbayern and clipped based on the spatial extent of Munich. For the same AOI, Level of Detail 2 (LoD2) building data in *OGC CityGML* [31] format is obtained from the Bavarian Surveying Administration Open Data Portal[11] shown in Fig 1c. Finally, we consider five different types of raster data with various spatial and temporal resolutions, such as the SRTM Digital Elevation data

[9] https://shapely.readthedocs.io/en/stable/index.html
[10] https://download.geofabrik.de/europe/germany/bayern/oberbayern.html
[11] https://geodaten.bayern.de/opengeodata/

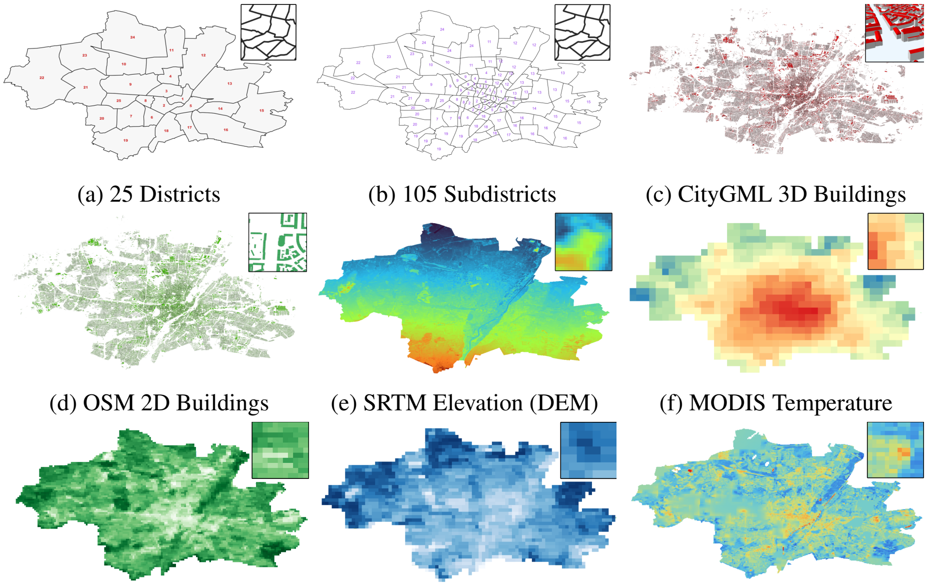

(a) 25 Districts (b) 105 Subdistricts (c) CityGML 3D Buildings (d) OSM 2D Buildings (e) SRTM Elevation (DEM) (f) MODIS Temperature (g) MODIS NDVI (h) NDSI Snow Cover (i) ECO Soil Moisture

Fig. 1: Heterogeneous geospatial data over the area of interest – Munich

(Fig. 1e), MODIS Temperature data (Fig. 1f), MODIS Vegetation (NDVI) (Fig. 1g), NDSI Snow Cover data (Fig. 1h) and ECOSTRESS Soil Moisture data (Fig. 1i).

OntoRaster relies on the GeoSPARQL 1.1 ontology describing the semantics of vector geometries and on the *Raster Ontology*, which represents n-dimensional generic raster data or coverage based on the OGC CIS v1.1 Standard. In this paper, OntoRaster adopted the GeoNames v3.3 ontology[12], which exposes the semantics of GeoNames[13] data, containing over 12 million unique geographical features with names including alternate and translated names, population, time-zone, geo coordinates, etc., gathered from various data sources.

The OpenStreetMap (OSM) dataset represents semantic concepts of map objects or physical ground features (e.g., roads or buildings) using key/value pairs *tags*. Tags are attached with additional properties, which are encoded as *nodes* (points of interest or centroids), *ways* (lines and polygons) and *relations* (groups of objects). The world data set currently contains approximately 9.9 billion nodes, 1.1 billion ways and 13 million relations to date[14]. For instance, "Hospital is a building" is implied as `tag: key="building" value="hospital"` in OSM terminology. But this may be not reflect building's current usage such as, a hospital that is abandoned or repurposed to be a something else is still a `building=hospital`, and to mark active hospitals

[12] https://www.geonames.org/ontology/documentation.html
[13] https://www.geonames.org/
[14] https://planet.openstreetmap.org/statistics/data_stats.html

Table 1: Information about the data in Fig. 1

	Data	**Data**	**Source**	**Feature**	**Feature**	**Spatial**	**Temporal**
	Type	**Field**	and **Sensors**	**Unit**	**Count**	**Resolution**	**Resolution**
(a)	Relational	25 Districts	-	point	4820	-	-
(b)	Relational	105 Sub-districts	-	point	6692	-	-
(c)	Relational	Building Footprints	OpenStreetMap	point	3285853	-	-
(d)	Relational	3D Buildings	CityGML 3D	point	5513088	-	-
(e)	Raster	Elevation	SRTM	pixel	873010	30m x 30m	-
(f)	Raster	Land Temperature	MODIS	pixel	1012	1km x 1km	daily
(g)	Raster	NDVI Vegetation	MODIS	pixel	15925	250m x 250m	16 days
(h)	Raster	Snow Cover	NDSI	pixel	4048	500m x 500m	daily
(i)	Raster	Soil Moisture	ECOSTRESS	pixel	3513088	70m x 70m	seconds

Table 2: List of prefixes used for RasSPARQL queries

Prefix	IRI Namespace
`:`	`https://github.com/aghoshpro/OntoRaster/`
`gn`	`https://www.geonames.org/ontology#`
`geo`	`http://www.opengis.net/ont/geosparql#`
`geof`	`http://www.opengis.net/def/function/geosparql/`
`rdfs`	`http://www.w3.org/2000/01/rdf-schema#`
`bldg`	`http://www.opengis.net/citygml/building/2.0/`
`lgdo`	`http://www.linkedgeodata.org/ontology/`
`rasdb`	`https://github.com/aghoshpro/RasterDataCube/`

`amenity=hospital` is used. Munich is identified by OSM ID 62428[15]. The semantics of this vast collection of OSM data are conceptualised using the LinkedGeoData (LGD) ontology under the LinkedGeoData project [32]. Employing this ontology and corresponding mappings, the OntoRaster framework integrates different types of raster data with various OSM data, including building data.

Let us start with some example RasSPARQL queries to demonstrate the workflow with newly added data. Queries are executed in the SPARQL endpoint provided by the VKG system Ontop under the OntoRaster framework. The namespace prefixes listed in Table 2 refer to the vocabularies utilised by OntoRaster.

Q1 is a simple query that demonstrates integration of vector data (district geometries) with raster data (elevation). By changing the `:Region_District_Munich` class name to `:Region_SubDistrict_Munich` one obtains results over sub-districts of Munich. ***Q2*** amalgamates OSM data with elevation raster data by utilising the OSM building class or tag for residential properties, i.e., `residential`. This query determines the number of residential buildings within those sub-districts of Munich where the spatial average elevation exceeds 520 meters, as calculated by OntoRaster's raster function `rasSpatialAverage`. The user can modify OSM tags in the query for various building categories, such as `School`, `Hospital`, including over 100+ classes. ***Q3*** depicts school buildings within the districts of Munich with high

[15] https://www.openstreetmap.org/relation/62428#map=11/48.1397/11.5380

vegetation (using raster data NDVI). The spatial query ***Q4*** utilises RSTN (means “railroad station“), one of 680+ GeoNames features[16] (e.g., S.UNIV, S.MTRO, etc.) as an additional input to integrate GeoNames with raster data w.r.t. respective ontologies.

Q1 Find all districts in Munich whose average terrain elevation is above 515 meters.

```
SELECT ?distName ?elevation {
  ?region a :Region_District_Munich ; rdfs:label ?distName ;
          geo:asWKT ?distWkt .
  ?gridCoverage a :Raster ; rasdb:rasterName ?rasterName .
  FILTER (CONTAINS(?rasterName, 'Elevation'))
  BIND ('2000-02-11T00:00:00+00:00'^^xsd:dateTime AS ?timeStamp)
  BIND (rasdb:rasSpatialAverage(?timeStamp, ?distWkt, ?rasterName)
        AS ?elevation)
  FILTER(?elevation > 515)
}
```

Q2 Find all residential buildings within those sub-districts in Munich whose average terrain elevation is above 520 meters.

```
SELECT ?bldgName ?subdistName ?elevation {
  ?region a :Region_SubDistrict_Munich ;
          rdfs:label ?subdistName ; geo:asWKT ?subdistWkt .
  ?bldg a lgdo:Residential ; rdfs:label ?bldgName ; geo:asWKT ?bldgWkt .
  ?gridCoverage a :Raster ; rasdb:rasterName ?rasterName .
  FILTER (geof:sfWithin(?bldgWkt, ?subdistWkt))
  FILTER (CONTAINS(?rasterName, 'Elevation'))
  BIND ('2000-02-11T00:00:00+00:00'^^xsd:dateTime AS ?timeStamp)
  BIND (rasdb:rasSpatialAverage(?timeStamp, ?subdistWkt, ?rasterName)
        AS ?elevation)
  FILTER(?elevation > 520)
}
```

Q3 Find all schools and respective districts in Munich where the average vegetation is high (ndvi > 0.35) on 1st January 2022.

```
SELECT ?bldgName ?distName ?ndvi {
  ?region a :Region_District_Munich ; rdfs:label ?distName ;
          geo:asWKT ?distWkt .
  ?building a lgdo:School ; rdfs:label ?bldgName ; geo:asWKT ?bldgWkt .
  ?gridCoverage a :Raster ; rasdb:rasterName ?rasterName .
  FILTER (geof:sfWithin(?bldgWkt, ?distWkt))
  FILTER (CONTAINS(?rasterName, 'NDVI'))
  BIND(''red''AS ?bldgWktColor)
  BIND ('2022-01-01T00:00:00+00:00'^^xsd:dateTime AS ?timeStamp)
  BIND (rasdb:rasSpatialAverage(?timeStamp, ?distWkt, ?rasterName) AS ?ndvi)
  FILTER(?ndvi > 0.35)
}
```

Q4 Find the spatial average temperature for a custom region in Munich and all railroad stations (gn:S.RSTN) lying within it.

```
SELECT ?featureName ?clipped {
  ?gname a gn:S.RSTN ; # or S.UNIV, H.LKS, S.MTRO etc., 680+ GeoNames classes
         rdfs:label ?featureName ; geo:asWKT ?featureWkt .
  ?gridCoverage a :Raster ; rasdb:rasterName ?rasterName .
  FILTER (geof:sfWithin(?featureWkt,?customRegionWkt))
  FILTER (CONTAINS(?rasterName, 'Munich_MODIS_Temperature_1km'))
  BIND ('POLYGON((11.548455354852285 48.14904359943597,
                  11.627292392870094 48.147177113486606,
                  11.58380485801181 48.11696471831948,
```

[16] https://www.geonames.org/export/codes.html

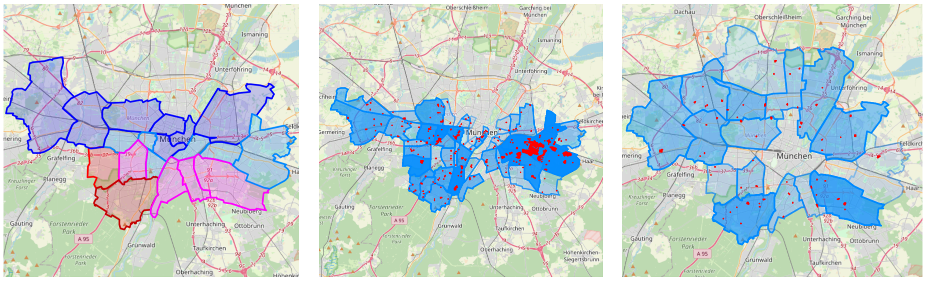
```
                  11.546675164347198 48.119171832113835,
                  11.548455354852285 48.14904359943597))' AS ?customRegionWkt)
  BIND ('2022-01-01T00:00:00+00:00'^^xsd:dateTime AS ?timeStamp)
  BIND (rasdb:rasSpatialAverage(?timeStamp, ?customRegionWkt, ?rasterName)
        AS ?clipped)
}
```

CityGML datasets comprise a collection of XML files, where each delineates a segment of the building information at a designated Level of Detail (LoD). But standard GML/XML encoding of CityGML is inadequate for intricate queries, especially those involving spatial-temporal analysis [13]. The primary approach for handling CityGML data is to store it as relational tables in the 3DCityDB[17] system, and thereafter accessing it using regular SQL. However, end users face challenges in constructing queries for their ad-hoc analytical tasks due to the incongruity between the conceptual semantics of CityGML and the relational schema utilised in 3DCityDB. ONTORASTER adopted the most well-known CityGML ontology[18], which is a direct translation of the CityGML XML Schema to OWL for describing the semantics of CityGML 3D building data, developed by the Knowledge Engineering @ CUI group at the University of Geneva. We use the 3DCityDB importer-exporter tool to import CityGML files into the relational database (PostgreSQL) for Munich. We created mappings to connect the data with the ontology mentioned above, thereby populating the RDF graph of CityGML. Here, we noticed that several columns in the CityGML building table are empty, and there is no column describing the buildings' locations, making it difficult to use this data to compose queries using raster data and additional spatial analysis.

OSM contains complementary spatial and semantic information related to CityGML data that can be utilised for spatial queries and applications [13]. But heterogeneity between the CityGML and OSM datasets makes it difficult to link them together. The building information in OSM data primarily comprises the building footprint layers (2D polygons) and the point of interest layer (points). In contrast, the CityGML dataset provides three-dimensional building models ranging from coarse models (LoD0) to very detailed ones (LoD4) with semantic information. The LoD0 building model in CityGML is essentially a 2D footprint represented as a closed polygon. LoD1 is a building model with height information, while the LoD2 building model features detailed roof structures and walls as extruded 3D objects derived from the 2D footprint. We observe that 2D footprints (polygons) are standard features shared between OSM and CityGML data, which can be geometrically matched to link the two datasets. We relied on a geometrical matching technique [15,29] to spatially align the CityGML data based on OSM building footprints. This provides geo-location information to CityGML data, which initially only contained building information. The majority of CityGML ground surfaces are successfully matched with OSM polygons. Since OSM building footprint data can already be integrated with raster data for spatial query processing, we can argue that CityGML building data (LoD0, LoD2) associated with OSM data can also be queried with raster data, provided that the ontologies are linked. More details on linking CityGML with OSM data are discussed in [13], while we concentrate here on combining CityGML with raster data.

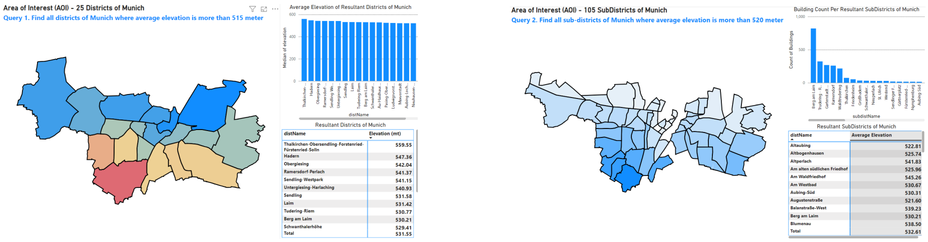
[17] https://www.3dcitydb.org/3dcitydb

[18] https://cui.unige.ch/isi/ke/ontologies



(a) $Q1$ (b) $Q2$ (c) $Q3$

Fig. 2: Visualisation of results of queries $Q1$–$Q3$ under ONTORASTER



Area of Interest (AOI) - 25 Districts of Munich

Query 1. Find all districts of Munich where average elevation is more than 515 meter

Resultant Districts of Munich

distName	Elevation (mt)
Thalkirchen-Obersendling-Forstenried-Fürstenried-Solln	559.55
Hadern	547.36
Obergiesing	542.04
Ramersdorf-Perlach	541.37
Sendling-Westpark	541.15
Untergiesing-Harlaching	540.93
Sendling	531.58
Laim	531.42
Tudering-Riem	530.77
Berg am Laim	530.21
Schwanthalerhöhe	529.41
Total	531.55

Area of Interest (AOI) - 105 SubDistricts of Munich

Query 2. Find all sub-districts of Munich where average elevation is more than 520 meter

Resultant SubDistricts of Munich

distName	Average Elevation
Altaubing	522.81
Altbogenhausen	525.74
Altperlach	541.83
Am alten südlichen Friedhof	525.96
Am Waldfriedhof	545.26
Am Westbad	530.67
Aubing-Süd	530.31
Augustenstraße	521.60
Balanstraße-West	539.23
Berg am Laim	530.21
Blumenau	538.50
Total	532.61

(a) $Q1$ (b) $Q2$

Fig. 3: Dashboard reports for $Q1$–$Q2$ in Ms Power BI

4.3 Visualisation of Query Result

Fig. 2 illustrates the visualisation of the previously mentioned queries at the SPARQL endpoint. Fig. 2a (for query $Q1$) displays the districts of Munich with an average elevation of more than 515 meters, using a colour map (where blue indicates low and red indicates high). Fig. 2b (for query $Q2$) depicts all residential buildings, i.e., red dots, over those sub-districts of Munich where the average elevation is more than 520 meters. Similarly, Fig. 2c (for query $Q3$) shows all schools in those selected districts of Munich where the average vegetation (raster) is high. Note that, in the map, some regions (e.g., districts or sub-districts) exhibit a deeper blue hue than others due to a higher concentration of OSM objects, such as schools, residential buildings, or hospitals. By analysing these at-a-glance visuals, users can obtain location-based information to support urban planning efforts or other use cases, depending on business demands.

5 Extending VKGs to Geospatial Business Intelligence (GeoBI)

Formulating SPARQL queries over a KG can be challenging, especially for business professionals who are not familiar with its syntax and semantics, and semantic web notions in general [26]. We have therefore experimented with different methods to support the use of BI tools in the VKG setting.

Method 1: OntoRaster + Power BI Tools. The ONTORASTER framework yields two different outputs, namely spatial-temporal aggregated values and filtered raster arrays. The first method is simple and essentially restricted to aggregated values, since handling filtered raster arrays (which are retrieved as RDF strings) requires further processing, depending on the use case, since RDF does not yet support the array data type. Hence, processing large arrays simply as strings would negatively impact performance. ONTORASTER provides users with the option to export query results in two file formats: `.csv` and `.json`, which are widely used in the business domain. The exported `.csv` file containing the query results can be loaded into Power BI using the native built-in CSV connector. Similarly, if the output is in `.json` format, it can be loaded into Power BI using a JSON connector. Fig. 3 illustrates the interactive dashboards based on the results of queries $Q1$–$Q2$, created with Power BI's supported map visualisations: *shape map*, *filled map*, and *world map*. Business professionals can interact with these dashboards to evaluate the RasSPARQL query results based on the VKGs generated from combined relational+raster data via mappings.

Method 2: Ontopic Suite's Semantic SQL Interface and BI Tools. Ontopic Suite[19] is an intuitive, no-code environment that connects to cloud or on-premises data and provides a user-friendly interface for designing the declarative mappings of a VKG, through which the underlying legacy data is exposed. In general, (V)KGs cannot be queried directly from BI tools, since such tools expect a traditional relational interface and issue SQL queries directly (e.g., via JDBC). To overcome this restriction, while still leveraging the semantic abstraction layer provided by the KG, Ontopic Suite can expose the KG via its *Semantic SQL Interface* component as a set of relational tables that can be queried using traditional SQL code, as written within BI tools. Such tables are organised to support efficient query evaluation, with a reduced number of relational joins. E.g., when data properties are relevant for the instances of a class C, they are grouped as attributes in a single table, whose primary key is given by an attribute containing the IRIs of C. In this way, one can avoid numerous costly joins whenever the query needs to return instances of the class, along with (some or all of) their data properties, which is a common form of request in a BI scenario. A SQL query issued via Ontopic Suite's Semantic SQL Interface is translated into ONTOP's internal *intermediate query* (IQ) representation, which is a uniform format adopted for processing queries issued over the KG. From that moment onwards, queries are processed by ONTOP in the same manner as SPARQL queries issued directly over the VKG. We are currently extending the SQL Interface component to support raster query functionalities, allowing the corresponding requests to be issued directly from BI tools. Such extended function calls are recognised internally by ONTORASTER and processed analogously to those appearing in RasSPARQL queries. We observe that the Semantic SQL Interface is a proprietary software component commercialised by Ontopic, and therefore, its extension towards raster functionalities cannot be released as open source.

[19] https://docs.ontopic.ai/suite/

6 Conclusions

In our work, we have developed an enhanced version of the OntoRaster framework capable of handling more complex vector geometries such as polygons with holes, multi-polygons and their combinations, and multiple raster data with different dimensions (spatial, temporal, and spectral), which are common in real-world GeoBI use cases. We have also outlined approaches to utilise semantic query results from OntoRaster within BI tools to produce semantically enriched business information through map-based visualisations, contributing to Geospatial Business Intelligence (GeoBI). Furthermore, we have presented Ontopic Suite's Semantic SQL interface to enable direct querying of KGs from BI tools, and we have proposed enhancing the SQL interface with raster functions to query integrated raster and relational data under OntoRaster. We are extending OntoRaster's ability to export clipped raster data in OGC formats like GeoTIFF[20] and CovJSON[21], for better visualisation of raster data over OSM. We also intend to provide a more intuitive dashboard to facilitate direct and robust query formulation and result visualisation. As an additional feature, we are developing an LLM-based RAG approach to generate RasSPARQL queries from natural language queries.

Acknowledgments. This research has been partially supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP), funded by the Knut and Alice Wallenberg Foundation, by the HEU project CyclOps (GA n. 101135513), by the Province of Bolzano and FWF through project OnTeGra (DOI 10.55776/PIN8884924), by the Province of Bolzano and EU through projects ERDF-FESR 1078 CRIMA, and ERDF-FESR 1047 AI-Lab, by MUR through the PRIN project 2022XERWK9 S-PIC4CHU, and by the EU and MUR through the PNRR project PE0000013-FAIR.

References

1. Amagata, D., Hara, T., Nishio, S.: Distributed Top-k Query Processing on Multi-Dimensional Data with Keywords. In: Proc. of SSDBM. pp. 1–12. ACM (2015). https://doi.org/10.1145/2791347.2791355
2. Angelaccio, M., Buttarazzi, B., Basili, A., Liguori, W.: Using Geo-Business Intelligence to Improve Quality of Life. In: Proc. of the IEEE 1st AESS European Conf. on Satellite Telecommunications (ESTEL) (2012). https://doi.org/10.1109/ESTEL.2012.6400196
3. Battle, R., Kolas, D.: Enabling the Geospatial Semantic Web with Parliament and GeoSPARQL. Semantic Web J. (2012). https://doi.org/10.3233/SW-2012-0065
4. Baumann, P.: The Datacube Manifesto. Tech. rep., EU EarthServer (2017), https://earthserver.eu/tech/datacube-manifesto/The-Datacube-Manifesto.pdf
5. Baumann, P., Misev, D., Merticariu, V., Huu, B.P.: Array Databases: Concepts, Standards, Implementations. J. of Big Data **8**(28) (2021). https://doi.org/10.1186/s40537-020-00399-2
6. Bereta, K., Xiao, G., Koubarakis, M.: Ontop-spatial: Ontop of Geospatial Databases. J. of Web Semantics **58** (2019). https://doi.org/10.1016/j.websem.2019.100514

[20] https://www.ogc.org/publications/standard/geotiff/
[21] https://www.ogc.org/publications/standard/coveragejson/

7. Bilidas, D., Mantas, A., Yfantis, F., Stamoulis, G., Koubarakis, M., Habas, J.M.T., Marco, E.S., Castel, F., Laine, C.: The Semantic Data Cube System Plato and Its Applications. In: Proc. of the IEEE Int. Geoscience and Remote Sensing Symposium (IGARSS). pp. 2514–2518 (2024). https://doi.org/10.1109/IGARSS53475.2024.10640737
8. Breunig, M., Bradley, P.E., Jahn, M., Kuper, P., Mazroob, N., Rösch, N., Al-Doori, M., Stefanakis, E., Jadidi, M.: Geospatial Data Management Research: Progress and Future Directions. ISPRS Int. J. of Geo-Information **9**(2) (2020). https://doi.org/10.3390/ijgi9020095
9. Calvanese, D., De Giacomo, G., Lembo, D., Lenzerini, M., Rosati, R.: Tractable Reasoning and Efficient Query Answering in Description Logics: The *DL-Lite* Family. J. of Automated Reasoning **39**(3) (2007). https://doi.org/10.1007/s10817-007-9078-x
10. Chaiprasert, A., Chongwatpol, J.: Business Intelligence and Geographic Information Systems in the Banking Industry: A Case Study of Home Loan Valuation. J. Information Technology Teaching Cases **14**(1), 90–107 (2024). https://doi.org/10.1177/20438869231155935
11. Dey, L.: Knowledge Graph-Driven Data Processing For Business Intelligence. WIREs Data Mining and Knowledge Discovery **14**(3) (2024). https://doi.org/10.1002/widm.1529
12. Dijkstra, L., Poelman, H., Veneri, P.: The EU-OECD Definition of a Functional Urban Area (FUA) (2019). https://doi.org/10.1787/d58cb34d-en
13. Ding, L., Xiao, G., Pano, A., Fumagalli, M., Chen, D., Feng, Y., Calvanese, D., Fan, H., and, L.M.: Integrating 3D City Data through Knowledge Graphs. Geo-spatial Information Science pp. 1–20 (2024). https://doi.org/10.1080/10095020.2024.2337360
14. Ding, L., Xiao, G., Pano, A., Stadler, C., Calvanese, D.: Towards the Next Generation of the LinkedGeoData Project Using Virtual Knowledge Graphs. J. of Web Semantics **71** (2021). https://doi.org/10.1016/j.websem.2021.100662
15. Fan, H., Zipf, A., Fu, Q., and, P.N.: Quality assessment for Building Footprints Data on OpenStreetMap. Int. J. of Geographical Information Science **28**(4), 700–719 (2014). https://doi.org/10.1080/13658816.2013.867495
16. Gao, F., Yue, P., Cao, Z., Zhao, S., Shangguan, B., Jiang, L., Hu, L., Fang, Z., Liang, Z.: A Multi-source Spatio-temporal Data Cube For Large-scale Geospatial Analysis. Int. J. of Geographical Information Science **36**(9) (2022). https://doi.org/10.1080/13658816.2022.2087222
17. Ghosh, A., Pano, A., Xiao, G., Calvanese, D.: OntoRaster: Extending VKGs with Raster Data. In: Proc. of the 8th Int. Joint Conf. on Rules and Reasoning (RuleML+RR). pp. 108–123. Springer (2024). https://doi.org/10.1007/978-3-031-72407-7_9
18. Gray, J., Chaudhuri, S., Bosworth, A., Layman, A., Reichart, D., Venkatrao, M., Pellow, F., Pirahesh, H.: Data cube: A relational aggregation operator generalizing group-by, cross-tab, and sub-totals. Data Mining and Knowledge Discovery **1** (1997). https://doi.org/10.1023/A:1009726021843
19. Han, J., Stefanovic, N., Koperski, K.: Selective materialization: An efficient method for spatial data cube construction. In: Research and Development in Knowledge Discovery and Data Mining. Lecture Notes in Computer Science, vol. 1394, pp. 144–158. Springer (1998). https://doi.org/10.1007/3-540-64383-4_13
20. Hanine, M., Boutkhoum, O., Agouti, T., Tikniouine, A.: A New Integrated Methodology Using Modified Delphi-fuzzy AHP-PROMETHEE for Geospatial Business Intelligence Selection. Information Systems and e-Business Management **15**, 897–925 (2017). https://doi.org/10.1007/s10257-016-0334-7
21. Hasan, K.M.A., Tsuji, T., Higuchi, K.: An Efficient Implementation for MOLAP Basic Data Structure and Its Evaluation. In: Advances in Databases: Concepts, Systems and Applications. pp. 288–299. Springer (2007). https://doi.org/10.1007/978-3-540-71703-4_26
22. Herring, J., et al.: OpenGIS® Implementation Standard for Geographic information - Simple Feature Access - Part 1: Common architecture [Corrigendum]. Tech. rep., Open Geospatial Consortium (2011). https://doi.org/10.25607/OBP-630

23. Hogan, A., et al.: Knowledge Graphs. ACM Computing Surveys (2021). https://doi.org/10.1145/3447772
24. Homburg, T., Staab, S., Janke, D.: GeoSPARQL+: Syntax, Semantics and System for Integrated Querying of Graph, Raster and Vector Data. In: Proc. of the 19th Int. Semantic Web Conf. (ISWC). Lecture Notes in Computer Science, vol. 12506. Springer (2020). https://doi.org/10.1007/978-3-030-62419-4_15
25. Karalis, N., Mandilaras, G., Koubarakis, M.: Extending the YAGO2 Knowledge Graph with Precise Geospatial Knowledge. In: Proc. of the 18th Int. Semantic Web Conf. (ISWC). pp. 181–197. Springer (2019). https://doi.org/10.1007/978-3-030-30796-7_12
26. Klímek, J., Škoda, P., Nečaskỳ, M.: Survey of Tools For Linked Data Consumption. Semantic Web J. **10**(4), 665–720 (2019). https://doi.org/10.3233/SW-180316
27. Kyriakos, C., Vavalis, M.: Business Intelligence through Machine Learning from Satellite Remote Sensing Data. Future Internet (2023). https://doi.org/10.3390/fi15110355
28. Laborie, S., Ravat, F., Song, J., Teste, O.: Combining Business Intelligence with Semantic Web: Overview and Challenges. In: Proc. of INFORSID. pp. 99–114 (2015), https://publications.ut-capitole.fr/id/eprint/29563
29. Liu, L., Fu, Z., Xia, Y., Lin, H., Ding, X., Liao, K.: A Building Polygonal Object Matching Method Based on Minimum Bounding Rectangle Combinatorial Optimisation and Relaxation Labelling. Transactions in GIS **27**(2), 541–563 (2023). https://doi.org/10.1111/tgis.13039
30. Motik, B., Cuenca Grau, B., Horrocks, I., Wu, Z., Fokoue, A., Lutz, C.: OWL 2 Web Ontology Language Profiles (Second Edition). W3C Recommendation, World Wide Web Consortium (Dec 2012), http://www.w3.org/TR/owl2-profiles/
31. Open Geospatial Consortium: OGC® City Geography Markup Language (CityGML). https://www.ogc.org/publications/standard/citygml/ (2021)
32. Stadler, C., Lehmann, J., Höffner, K., Auer, S.: LinkedGeoData: A Core for a Web of Spatial Open Data. Semantic Web J. pp. 333–354 (2012). https://doi.org/10.3233/SW-2011-0052
33. Stefano Nativi, P.M., Craglia, M.: A View-Based Model of Data-Cube To Support Big Earth Data Systems Interoperability. Big Earth Data (2017). https://doi.org/10.1080/20964471.2017.1404232
34. Tennison, J.: The RDF Data Cube Vocabulary. W3C Recommendation, World Wide Web Consortium (2014), https://www.w3.org/TR/vocab-data-cube/
35. Varga, J., Etcheverry, L., Vaisman, A.A., Romero, O., Pedersen, T.B., Thomsen, C.: QB2OLAP: Enabling OLAP on Statistical Linked Open Data. In: Proc. of the 32th IEEE Int. Conf. on Data Engineering (ICDE). pp. 1346–1349 (2016). https://doi.org/10.1109/ICDE.2016.7498341
36. Warmerdam, F.: The Geospatial Data Abstraction Library, Advances in Geographic Information Science, vol. 2, pp. 87–104. Springer (2008). https://doi.org/10.1007/978-3-540-74831-1_5
37. Xiao, G., Calvanese, D., Kontchakov, R., Lembo, D., Poggi, A., Rosati, R., Zakharyaschev, M.: Ontology-Based Data Access: A Survey. In: Proc. of the 27th Int. Joint Conf. on Artificial Intelligence (IJCAI) (2018). https://doi.org/10.24963/ijcai.2018/777
38. Xiao, G., Lanti, D., Kontchakov, R., Komla-Ebri, S., Güzel-Kalaycı, E., Ding, L., Corman, J., Cogrel, B., Calvanese, D., Botoeva, E.: The Virtual Knowledge Graph System Ontop. In: Proc. of the 19th Int. Semantic Web Conf. (ISWC) (2020). https://doi.org/10.1007/978-3-030-62466-8_17