Providing Satellite Data to Mobile Developers Using Semantic Technologies and Linked Data

Konstantina Bereta*, Hervé Caumont†, Ulrike Daniels‡, Daems Dirk§, Manolis Koubarakis*,
Despina-Athanasia Pantazi*, George Stamoulis*, Sam Ubels¶, Valentijn Venus¶ and Firman Wahyudi¶
*National and Kapodistrian University of Athens, Greece
†Terradue Srl, Italy
‡AZO Anwendungszentrum GmbH, Germany
§VITO, Belgium
¶RAMANI B.V., The Netherlands

***Abstract*—Copernicus is the European program for monitoring the Earth. It consists of a set of complex systems that collect data from satellites and in-situ sensors, process it, and provide users with reliable and up-to-date information on a range of environmental and security issues. Information extracted from Copernicus data is made available to users through Copernicus services addressing six thematic areas: land, marine, atmosphere, climate, emergency and security. The data processed and disseminated puts Copernicus at the forefront of the big data paradigm and gives rise to all relevant challenges: volume, velocity, variety, veracity and value. In this paper we discuss the challenges of big Copernicus data and how the Copernicus program handled them. We also present lessons learned from our project Copernicus App Lab, which takes Copernicus services information and makes it available on the Web using semantic technologies to aid its take up by mobile developers.**

I. Introduction

Earth observation (EO) is the gathering of data about our planet's physical, chemical and biological systems via satellite remote sensing technologies supplemented by Earth surveying techniques. *Copernicus* is currently the world's biggest EO program. It consists of a set of complex systems that collect data from satellites and in-situ sensors, process this data and provide users with reliable and up-to-date information on a range of environmental and security issues. The EO satellites that provide the data of the Copernicus programme are the *Sentinels*, and the *contributing missions*, which are operated by national, European or international organizations. Copernicus data is a paradigmatic case of big data which is made freely available to users through the *Copernicus services*, which address six thematic areas: land, marine, atmosphere, climate, emergency and security.

Copernicus App Lab (http://www.app-lab.eu/) is a two year project funded by the European Commission under the H2020 program. It focuses on the *volume* and *variety* challenges of Copernicus data and it is based on the previous research projects TELEIOS, LEO, and MELODIES, funded by FP7 ICT. It goes beyond these projects in the following important ways. Firstly, it creates a software architecture that provides on demand access to big Copernicus data. To achieve this it uses the OPeNDAP framework and the geospatial ontology-based data access system Ontop-spatial [2]. Now, mobile developers just need to develop an ontology describing the data they use and R2RML mappings that describe the correspondence between the data sources containing the data and the ontology. Using traditional approaches, developers have to implement different clients/adapters in their applications corresponding to the different file formats their data is in to process it. Now, they can use the functionalities of the Ontop-spatial mapping language for all data sources without considering their formats. Secondly, it makes the Copernicus App Lab tools available as Docker images that are deployed in the Terradue cloud platform as cloud services. In this way, it brings computing resources close to the data. Moreover, the platform allows mobile developers to carry out massively parallel processing without downloading the data and carry out the processing locally. Lastly, it allows search engines like Google to store knowledge about datasets produced by Copernicus in their internal knowledge graph. In this way, search engines will be able to answer sophisticated users' questions which are beyond the reach of modern search engines today. A more detailed description of the Copernicus App Lab project is given in [3].

II. The Copernicus App Lab Architecture

Figure 1 presents the Copernicus App Lab software architecture and the conceptual architecture of the *Copernicus integrated ground segment*. A *ground segment* is the hardware and software infrastructure where raw data, often from multiple satellite missions, is ingested, processed, cataloged, and archived. At the bottom of the figure, the Copernicus *data sources* are shown. These are Sentinel data from ESA, Sentinel data from the European Organisation for the Exploitation of Meteorological Satellites (EUMETSAT), satellite data from contributing missions and in-situ data. The next layer makes Copernicus data and information available to interested parties in three ways: via the Copernicus Open Access Hub, via the Copernicus Core Services and via the Data and Information Access Service (DIAS).

All the software components of the project run in the Terradue cloud platform (https://www.terradue.com/portal/). The platform allows cloud orchestration, storage virtualisation, and virtual machine provisioning, as well as application burst-loading and scaling on third-party cloud infrastructures. Within

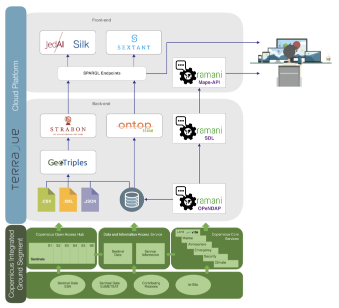

Fig. 1. The Copernicus integrated ground segment and the Copernicus App Lab software architecture

the Terradue cloud platform, the developer cloud sandbox service provides a platform-as-a-service (PaaS) environment to prepare data and processors. It has been designed in order to automate the deployment of the resulting EO applications to any cloud computing facility that can offer storage and computing resources (e.g., AWS).

In Copernicus App Lab, the mobile developers can access the Copernicus data in two ways: they can either download the data from the Copernicus Open Access Hub or the Web sites of individual Copernicus services, or from the popular OPeNDAP framework (https://www.opendap.org/) for accessing scientific data. By following the first approach (workflow on the left part of the two top layers of Figure 1), after downloading the data, they should transform it into RDF using the tool GeoTriples [7] or scripts written especially for this task. GeoTriples allows the transformation of geospatial data stored in raw files (shapefiles, CSV, KML, XML, GML and GeoJSON) and spatially-enabled RDBMS (PostGIS and MonetDB) into RDF graphs using well-known geospatial vocabularies such as the Open Geospatial Consortium (OGC) standard GeoSPARQL [10]. The performance of GeoTriples has been studied experimentally [7] using large publicly available geospatial datasets. It has been shown that it is very efficient, especially if the mapping processor we implement uses Apache Hadoop.

After tranforming the Copernicus data into RDF, it can be stored in the spatiotemporal RDF store Strabon [4], [6]. Strabon can store and query linked geospatial data that changes over time. The benchmark Geographica [4], [5] showed that Strabon is the most efficient spatiotemporal RDF store available today. After storing the Copernicus data in Strabon the application developers may also interlink it with other relevant data using the interlinking tools JedAI and Silk. JedAI is a toolkit for entity resolution and its multi-core version has been shown to be scalable to large datasets [9]. Silk is a well-known framework for interlinking RDF datasets which we extended to deal with geospatial and temporal relations [11].

The *novel way of accessing Copernicus data and information* in Copernicus App Lab is based on the popular *OPeNDAP framework* and it is showed in the workflow on the right part of the two top layers of Figure 1. OPeNDAP provides a powerful data translation facility so that users do not need to know the detailed formats of data stored in servers, and can simply use a client that supports a model they are comfortable with. The *streaming data library (SDL)* implemented by RAMANI communicates with the OPeNDAP server to receive Copernicus services data as *streams*. In this way, SDL enables on-the-fly computation of spatial and temporal aggregations (e.g., a longterm moving average). The users can access SDL through a list of APIs that are enhanced with an API ontology directly linked to a function ontology that describes the offered analytics and functionality. This ontology describes calls and responses of the API and assists users in determining valid functions over different data types. The API responses are provided as JSON-LD with direct references to the semantics of the returned variables, allowing easier interpretation. OPeNDAP and SDL are installed and configured by VITO on a virtual machine running on the VITO hosted PROBA-V mission exploitation platform (https://proba-v-mep.esa.int), which has direct access to the data archives of the Copernicus global land service. The installation of OPeNDAP was done using Docker and access to the Copernicus global land and PROBA-V datasets via OPeNDAP is realised by mounting the necessary disks on the virtual machine.

One major contribution of Copernicus App Lab is the extension of the ontology-based data access system Ontop-spatial [1] with OPeNDAP support. Ontop-spatial is a system that connects to existing geospatial databases and creates virtual semantic graphs on top of them using ontologies and mappings, without downloading files and transforming them into RDF. Mappings encode how we map relational data to RDF terms. As we describe in [2], the new version of Ontop-spatial is able to connect to non-relational external data sources (e.g., APIs like OPeNDAP) and enable users to pose GeoSPARQL queries on top of them without having to import the data in relational databases.

Finally, developers can visualize the data using the tools Sextant [8] or Maps-API (https://ramani.ujuizi.com/maps/index.html). Sextant is essentially a GIS for linked geospatial data. It allows users to build layered maps consisting of geospatial data which is made available in various formats (e.g., KML, GML etc.) and SPARQL or GeoSPARQL endpoints. The Maps-API is similar to Sextant in terms of visualization functionality, but it takes its data from SDL and it cannot deal with linked geospatial data sources accessed by SPARQL or GeoSPARQL.

All tools are open source and they are available on the following Web page: http://kr.di.uoa.gr/#systems

III. A Copernicus App Lab Case Study

We now present a simple case study, which shows the functionality of the Copernicus App Lab software, by studying the "greenness" of Paris. We achieve this goal by relating "greenness" features of Paris using geospatial data sources such as OpenStreetMap and relevant Copernicus datasets from the land monitoring service of Copernicus, which are the leaf-area index dataset (global), the CORINE land cover dataset (pan-European) and the Urban Atlas dataset (local).

Leaf area index (LAI) is a dimensionless quantity that characterizes plant canopies, defined as the one-sided green leaf area per unit ground surface area in broadleaf canopies (https://en.wikipedia.org/wiki/Leaf_area_index). The *CORINE land cover dataset* covers 39 EU countries (https://land.copernicus.eu/pan-european/corine-land-cover). Land cover is characterized using a 3-level hierarchy of classes with 44 classes in total at the 3rd level. The *Urban Atlas dataset* (https://land.copernicus.eu/local/urban-atlas/view) provides land use and land cover data for European urban areas and it covers 800 urban areas in 28 EU countries. Our case study also utilizes data from *OpenStreetMap* and the global administrative divisions dataset *GADM*. OpenStreetMap is an open and free map of the whole world constructed by volunteers. GADM (https://gadm.org/) is an open and free dataset giving us the geometries of administrative divisions of various countries.

The first task of creating a case study using the Copernicus App Lab software is to develop INSPIRE-compliant ontologies for the Copernicus data. The *INSPIRE directive* (https://inspire.ec.europa.eu/) creates an interoperable spatial data infrastructure for the EU, to enable the sharing of spatial information among public sector organizations and better facilitate public access to spatial information across Europe.

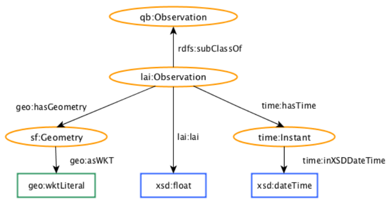

Fig. 2. The LAI ontology

An ontology for the LAI dataset is shown in Figure 2. We re-used classes and properties from the Data Cube ontology (https://www.w3.org/TR/vocab-data-cube/, namespace `qb`) specializing them when appropriate, classes and properties from the GeoSPARQL ontology [10] (namespaces `sf`, `geo`), from the Time Ontology (https://www.w3.org/TR/owl-time/, namespace `time`), and datatypes from XML-Schema. The class and properties introduced by us use the prefix `lai`.

After defining all the ontologies, we can easily translate them into RDF using a custom script, store them in Strabon and pose interesting queries. For example, assuming appropriate PREFIX definitions, the following GeoSPARQL query asks for the LAI values of the area occupied by the Bois de Vincennes park in Paris.

```
SELECT DISTINCT ?geoA ?geoB ?lai WHERE {
  ?areaA osm:poiType osm:park.
  ?areaA geo:hasGeometry ?geomA . ?geomA geo:asWKT ?geoA .
  ?areaA osm:hasName  "Bois de Vincennes"^^xsd:string .
  ?areaB lai:lai ?lai .
  ?areaB geo:hasGeometry ?geomB . ?geomB geo:asWKT ?geoB .
  FILTER(geof:sfIntersects(?geoA, ?geoB))}
```

Similarly, in Figure 3, we used Sextant to build a temporal map that shows the "greenness" of Paris, using the datasets LAI, GADM, CORINE land cover, Urban Atlas and OpenStreetMap. We show how the LAI values (small circles) change over time in each administrative area (magenta lines) of Paris and correlate these readings with the land cover of each area (taken from the CORINE land cover or Urban Atlas).

All RDF datasets and ontologies that have been discussed above are freely available at: http://kr.di.uoa.gr/#datasets.

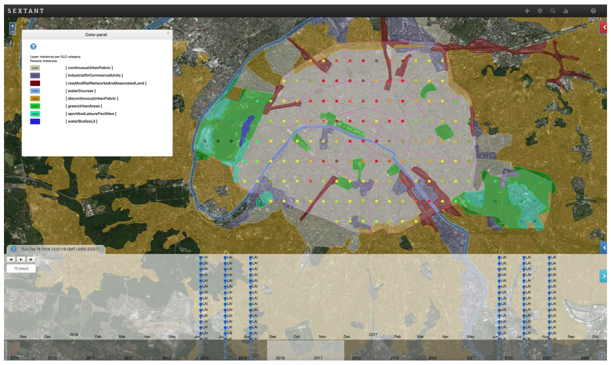

Fig. 3. The "greenness" of Paris

The described case study can also be developed using the workflow on the right in the Copernicus App Lab software architecture of Figure 1. In this case, the datasets are queried using Ontop-spatial and visualized in Sextant without transforming any datasets into RDF, as developers write R2RML mappings expressing the correspondence between a data source and classes/properties in the corresponding ontology. A mapping expressed in the native mapping language of Ontop-spatial which is less verbose than R2RML, is provided below.

```
mappingId opendap_mapping
target lai:{id} rdf:type lai:Observation .
      lai:{id} lai:lai {LAI}^^xsd:float;
                time:hasTime {ts}^^xsd:dateTime .
      lai:{id} geo:hasGeometry _:g .
      _:g geo:asWKT {loc}^^geo:wktLiteral .
source SELECT id, LAI, ts, loc FROM (ordered opendap
      url:https://analytics.ramani.ujuizi.com/
      thredds/dodsC/Copernicus-Land-timeseries-global
          -LAI%29/readdods/LAI/)  WHERE LAI > 0
```

In this mapping, the `source` is the LAI dataset, which contains observations that are LAI values, the time and the location for each observation. The dataset is provided through the RAMANI OPeNDAP server of the Copernicus App Lab software stack. The Operator `Opendap` retrieves this data and populates a virtual SQL table with schema `(id,LAI,ts,loc)`. Because of the fact that the `Opendap` operator is implemented as an SQL user-defined operator, it can be embedded into any SQL query. In the above mapping, we also add a filter to the query to eliminate negative or zero LAI values, in order to refine the data we want to translate into virtual RDF terms. The way the relational data is mapped into RDF terms is encoded by the `target` part of the mapping.

IV. Lessons Learned and Future Challenges

OPeNDAP and SDL provide streaming data to the user and have some significant advantages over the OGC Web Coverage

Service standard which is already offered by VITO. Firstly, from a data provider perspective, OPeNDAP is easier to use, especially for mobile developers that are not experts in EO. It is able to deal with a wider variety of grid types and it can be easily extended with different conventions, allowing for easier integration of different datasets, without causing overhead like file conversion. Moreover, OPeNDAP provides easy access to the user through a single access point, as it enables the loose coupling of different Copernicus data sources into one data model. OPeNDAP also allows for the caching of datasets by serialisation based on internal array indices, while it ensures that metadata are intrinsically embedded in the TCP/IP response, regardless of container type (GeoTIFF, NetCDF, etc.), which is beneficial for the semantic enrichment process that may happen at higher layers of the architecture.

The most popular linked geospatial data tools have been Sextant and Ontop-spatial. Sextant has been irreplaceable as there is currently no other tool for visualizing linked geospatial data. The most innovative aspect of using Ontop-spatial in Copernicus App Lab is its ability to give access to Copernicus data through the OPeNDAP framework. When data is stored in a database connected with Ontop-spatial, DBMS optimisations and database constraints are applied and query plans are optimized. This does not happen in the case where Ontop-spatial retrieves data on-the-fly from OPeNDAP, since data is preprocessed before it gets translated into virtual triples using Ontop-spatial. However, if we want to access Copernicus data that gets frequently updated, the virtual RDF graphs approach is useful as it avoids the repeated translation steps that have to be done by the data provider. To improve performance, we implemented a caching mechanism so that queries that result in the same API calls for a time window w, whose length is a configurable parameter, can get cached data. We also extended our system with the ability to integrate other kinds of data e.g., HTML tables and social media data (e.g., twitter, foursquare). In our current work, we try to improve performance by developing further optimisation techniques.

In September 2017 and 2018, participants of the ESA Space App Camp (www.app-camp.eu/) had the opportunity to use the Copernicus App Lab technologies for the implementation of their demo applications. The goal was to make EO data, particularly from Copernicus, accessible to a wide range of citizens and businesses. The winning teams of these two App Camps, AiR and URBANSAT, used Copernicus App Lab tools to access and integrate data from different sources.

It is important to discuss the approach CREODIAS platform takes, which is very similar to our projects TELEIOS, LEO, Melodies and Copernicus App Lab. CREODIAS platform is a cloud-based one-stop shop for all Copernicus satellite data and imagery, as well as the Copernicus services information (https://creodias.eu/). The CREODIAS approach is limited though since only *metadata* of Copernicus datasets are available as linked data and can be queried by relevant discovery tools. We, on the other hand, allow users to also make information and knowledge extracted from Copernicus data available as linked data. In this way, we contribute to the *value* dimension of big Copernicus data as we allow users to combined their produced linked data with other public or private linked datasets, while enabling the development of applications easily.

Google has recently activated the beta version of its dataset search (https://toolbox.google.com/datasetsearch), where the datasets that are annotated using *schema.org* (https://schema.org/), as proposed by Google, show up. We have followed these guidelines and annotated all the datasets used in the use case of Section III, and made them available at the following link: http://kr.di.uoa.gr/#datasets. We have also recommended that the same practice is followed by the Copernicus services we have worked with (land monitoring, global land and atmosphere services). In our current work we design an extension of the community vocabulary *schema.org* which allows the annotation of EO data in general and Copernicus data in particular, by extending the class *Dataset* with subclasses and properties according to the EO dataset metadata defined in relevant OGC standards.

V. Conclusion

In this paper we argued that Copernicus data is a paradigmatic source of big data. Our project Copernicus App Lab has developed a novel software stack that can be used to develop applications using Copernicus data even by developers that are not experts in EO. We presented a case study developed using the Copernicus App Lab software stack and discussed lessons learned and future plans for information retrieval, database and knowledge management research in the context of Copernicus.

Acknowledgment

This work has been funded by the EU project Copernicus App Lab (730124).

References

[1] K. Bereta and M. Koubarakis, *Ontop of Geospatial Databases*, ISWC, 2016

[2] K. Bereta and M. Koubarakis, *Creating Virtual Semantic Graphs ontop of Big Data from Space*, BiDS, 2017

[3] K. Bereta, H. Caumont, U. Daniels, D. Dirk, M. Koubarakis, D.-A. Pantazi, G. Stamoulis, S. Ubels, V. Venus, F. Wahyudi, *The Copernicus App Lab project: Easy Access to Copernicus Data*, EDBT, 2019

[4] K. Bereta, P. Smeros and M. Koubarakis, *Representation and Querying of Valid Time of Triples in Linked Geospatial Data*, ESWC, 2013

[5] G. Garbis, K. Kyzirakos and M. Koubarakis, *Geographica: A Benchmark for Geospatial RDF Stores*, ISWC, 2013

[6] K. Kyzirakos, M. Karpathiotakis and M. Koubarakis, *Strabon: A Semantic Geospatial DBMS*, ISWC, 2012

[7] K. Kyzirakos, D. Savva, I. Vlachopoulos, A. Vasileiou, N. Karalis, M. Koubarakis, S. Manegold, *GeoTriples: Transforming Geospatial Data into RDF Graphs Using R2RML and RML Mappings*, Journal of Web Semantics, 2018

[8] C. Nikolaou, K. Dogani, K. Bereta, G. Garbis, M. Karpathiotakis, K. Kyzirakos, and M. Koubarakis, S. Manegold, *Sextant: Visualizing time-evolving linked geospatial data*, Journal of Web Semantics, 35(1), 2015

[9] G. Papadakis, K. Bereta, T. Palpanas and M. Koubarakis, *Multi-core Meta-blocking for Big Linked Data*, SEMANTICS, 2017

[10] M. Perry and J. Herring, *GeoSPARQL - A geographic query language for RDF data*, OGC Implementation Standard, 2012

[11] P. Smeros and M. Koubarakis, *Discovering Spatial and Temporal Links among RDF Data*, LDOW, 2016