

extension of the query language SPARQL 1.1 for querying and updating stRDF data. stRDF and stSPARQL use OGC standards (Well-Known Text and Geography Markup Language) for the representation of geospatial data and are implemented in the open source spatiotemporal RDF store Strabon.⁵

The presented wildfire monitoring service improves significantly the wildfire monitoring service used until now by NOA. The service is available on the web⁶ and is operational and used by decision makers and emergency services in Greece since the fire season of 2012. This wildfire monitoring service, as described in [3], has participated in the Open Track of the 10th Semantic Web Challenge⁷ that was co-located with the 11th International Semantic Web Conference and was awarded the third place.

The contributions of this work are the following:

- We present a wildfire monitoring service that has been developed using the semantic web technologies presented above. The service is currently operational at NOA and has been used on a daily basis in the fire season of 2012 by the emergency managers monitoring wildfires in Greece, such as the Greek Civil Protection Agency, the Fire Brigade, and the Army, for strategy planning and assessment.
- In contrast to similar wildfire monitoring services available at various EO agencies, the NOA service is exclusively built using state-of-the-art scientific database, semantic web and linked data technologies. Here, we discuss how semantic web technologies and linked open data have allowed NOA to build easily from scratch a new version of its service that is much easier to modify and reuse in other GMES environmental monitoring applications.
- We present a preliminary evaluation of the implemented service concentrating on the performance of the main component of the service, namely the RDF store Strabon, and demonstrating the technologies discussed in this work: the refinement operations that use linked geospatial data and which are expressed in stSPARQL.

The rest of the paper is organized as follows. Section 2 discusses related work. Section 3 describes in detail the workflow of the wildfire monitoring service and presents the architecture of the system that delivers the wildfire monitoring service describing its main components. Section 4 introduces the data model stRDF and the accompanying query language stSPARQL which play a crucial role in this application. Section 5 presents the datasets and the ontologies that are used in the fire monitoring service and Section 6 describes how these datasets and ontologies are utilized by stSPARQL updates to increase the accuracy of primary fire monitoring results. Finally, Section 7 presents the evaluation of the developed fire monitoring service and Section 8 concludes the paper.

2. Related work

Recently, Linked Open Data Cloud has started being enriched with geospatial data. Two representative examples are [4,5]. In [4] OpenStreetMap data are made available as RDF and queried using the declarative query language SPARQL. Using similar technologies, [5] makes available as linked data various heterogeneous Spanish public datasets. With the recent emphasis on open government data, some of it encoded already in RDF,⁸

portals such as LinkedGeoData⁹ demonstrate that the development of useful Web applications might be just a few SPARQL queries away.

Regarding EO data, EO data centers have only just started to organize archived data so as to make them available to EO scientists for research purposes. Data centers typically offer Web interfaces such as EOWEB¹⁰ for searching, inspection, and ordering of EO products. While EOWEB organizes archived data in a simple, hierarchical way, more sophisticated and expressive approaches are being explored that are based on ontologies [6–9]. In this context, TELEIOS extends previous work aiming at capturing also the semantics of the content of products that will be generated by knowledge discovery algorithms (e.g., fire detection algorithms) applied on EO images.

3. Service description

In this section, we give a detailed description of the real-time wildfire monitoring service and discuss the architectural design of the web application that offers this service.

3.1. The wildfire monitoring service

The wildfire monitoring service relies on an MSG/SEVIRI acquisition station that is operated by NOA. The station receives satellite images every 5–15 min which are stored in the state-of-the-art column-store DBMS MonetDB.¹¹ In MonetDB pre-processing operations (e.g., cropping, georeferencing) and a fire detection processing chain [10] are applied to the images. The processing chain detects areas (pixels of the images) where fire may exist. These areas are called *hotspots* and for each satellite image a geospatial vector file (ESRI shapefile), which describes the detected hotspots, is produced. A hotspot is mainly described by the time at which it was detected, its spatial extent and a confidence value denoting the probability that there is a fire in this area. The hotspots derived by the processing chain have dimensions equal to the resolution of the sensor, in this case 4 × 4 km. This spatial resolution is low, compared to other high resolution sensors. However, MSG/SEVIRI has geostationary orbit, which allows a very high observational frequency (5–15 min) over the same area of interest. Other satellite platforms with better spatial resolution have very low observational frequency (e.g., 12 h), so they are not suitable for a real-time monitoring application.

In order to increase the inherent coarse spatial resolution an important algorithmic advancement of the hotspot service integrates static geo-information and dynamic meteorological weather prediction data, aiming at enhancing the inherently coarse MSG/SEVIRI spatial resolution (3.5 km). Based on the result of the initial fire detection process (hotspot detection), a fire danger estimation model was designed. This model divides the initial SEVIRI pixel into a 7 × 7 grid, resulting into 49 sub-pixels, of 540 m spatial resolution. For each sub-pixel a new confidence value is attributed according to (i) the initial fire confidence level of a SEVIRI pixel, (ii) the type of the available fuel, based on CORINE Land Cover¹² data, (iii) the direction of the wind during the fire event (based on daily weather prediction models by NOA meteorological monitoring network¹³), and (iv) the geomorphology of the area, where the fire event is taking place (altitude and aspect).

⁹ <http://linkedgeo.org/>.

¹⁰ <http://eoweb.dlr.de/>.

¹¹ <http://www.monetdb.org/>.

¹² <http://www.eea.europa.eu/publications/CORO-landcover>.

¹³ <http://www.meteo.gr/>.

⁵ <http://strabon.di.uoa.gr/>.

⁶ http://ocean.space.noa.gr/seviri/fend_new/.

⁷ <http://challenge.semanticweb.org/2012/>.

⁸ <http://data.gov.uk/linked-data/>.

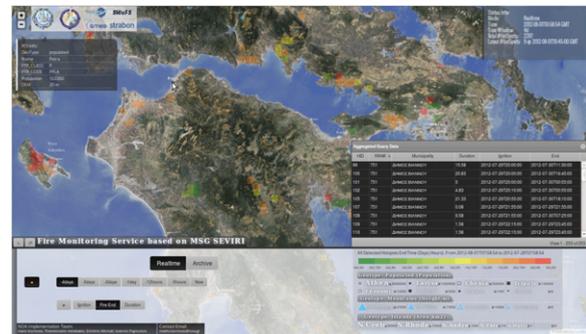
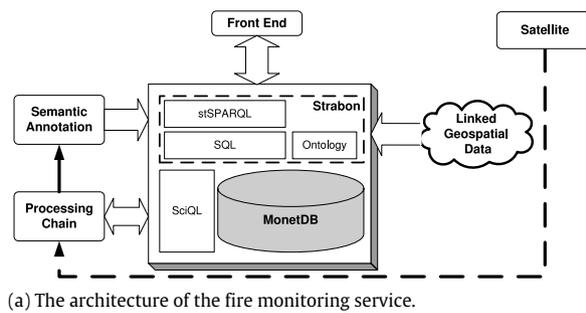


Fig. 1. The architecture and web application of the fire monitoring service.

Because of the low spatial resolution of the SEVIRI instrument, possible errors in image georeferencing, and potential weaknesses of the algorithms in [10] the derived hotspots have limited accuracy for specific scenarios. For example, it is usual to detect hotspots occurring in the sea or in locations represented by fully inconsistent land use/land cover classes (e.g., urban and agricultural areas). These may be false fire detections due to known problems with existing hotspot detection algorithms. Also, they may correspond to real fires in the vicinity of inconsistent areas and appear to be over these areas because of the low spatial resolution of the satellite images and errors introduced by georeferencing. Finally, they may be real fires started by farmers in big agricultural areas as part of their agricultural practices and thus they do not correspond to an emergency situation. These errors could be avoided if derived hotspot products are combined with auxiliary GIS layers about the underlined area. However, this information comes from diverse sources and it cannot be automatically integrated with the derived hotspots. The manual integration would forbid the real-time monitoring of fire-fronts, so we produce the stRDF representation of derived hotspots, which can be easily integrated with linked geospatial data, and remove the noisy information to deliver a final product with high accuracy. This process will be described in more detail in Section 6.

3.2. Architecture

The implementation of the wildfire monitoring service as described above is realized by the architectural diagram depicted in Fig. 1(a). This architecture comprises the following four components:

- (i) a satellite acquisition station that receives every 5 and 15 min satellite images which are used as the input in fire detection algorithms,
- (ii) a column store DBMS, called MonetDB where satellite images are stored and fire detection algorithms are executed,
- (iii) a geospatial RDF store, called Strabon, where the RDF representation of detected fires is stored and refined using SPARQL updates, and finally
- (iv) a web-based user interface is used to display the final product to the end-users.

MonetDB provides the query language SciQL which has been used to implement the fire detection processing chain. SciQL is a new SQL-based query language for scientific applications with arrays as first-class citizens, implemented on top of the state-of-the-art MonetDB database. SciQL provides efficient array manipulation primitives that enables us to perform low level image processing and image content analysis using a high-level declarative query language.

Strabon is a fully implemented semantic geospatial RDF store that can be used to store linked geospatial data expressed in stRDF and query them using stSPARQL. The data model stRDF is an extension of the W3C standard RDF, that allows the representation of geospatial data that changes over time [1,2]. stRDF is accompanied by stSPARQL, an extension of the query language SPARQL 1.1 for querying stRDF data. stSPARQL bears many similarities with the OGC standard GeoSPARQL [11] for querying geospatial data expressed in RDF. stSPARQL offers the same functionalities with the geometry extension and the geometry topology extension of GeoSPARQL in addition to geospatial aggregate functions. A detailed comparison between stSPARQL and GeoSPARQL is given in [12]. More information about stRDF and stSPARQL is given in Section 4 below.

The data model stRDF and the query language stSPARQL are used in the real-time wildfire monitoring service to seamlessly integrate derived hotspots with auxiliary geospatial information. In particular, we employ stRDF to represent not only knowledge extracted from satellite images (e.g., detected hotspots), but also to represent satellite image metadata (e.g., time of acquisition, geographical coverage), and auxiliary geospatial datasets encoded as linked data. Since hotspot products are encoded in stRDF, they can easily be combined with auxiliary linked geospatial data using stSPARQL queries and updates. In short, we use stSPARQL updates to enhance the information captured by the hotspots (e.g., add a specific municipality that may be at risk because of the fire) and we detect and discard false positives (e.g., hotspots in sea). We also detect and restore false negatives, for example, hotspots that occasionally turn off and on again are updated to be always on fire.

The final products are visualized on a map through a web based front-end that is depicted in Fig. 1(b). The service enables real-time wildfire monitoring while new satellite acquisitions arrive and are instantly processed. Additionally, a user can search the archive for a wildfire detected in the past and get a snapshot of it or watch (via simulation) how it proceeded. Information about hotspots (e.g., fire ignition time, nearby municipality) that are important for decision makers and crisis managers are also presented to the user.

Additionally, we can combine diverse information sources and generate added-value thematic maps which are very useful to civil protection agencies and firefighting teams during emergency situations. For example, one can then use stSPARQL to express in a single query an information request such as the following: “Find an image taken by a Meteosat second generation satellite on August 25, 2007 which covers the area of Peloponnese and contains hotspots corresponding to forest fires located within 2 km from a major archeological site”. Encoding this information request today in a typical interface to an EO data archive is impossible, because domain-specific concepts such as “forest fires” are not included in the archive metadata, thus they cannot be used as search criteria.

A more detailed description of the redevelopment of the entire NOA fire monitoring service may be found in [13]. [13] discusses also the utilization of scientific database management and array database technologies as the means to enabling transparent data and metadata access and analysis of EO data, the reimplementation of the fire detection algorithms in MonetDB, and a preliminary evaluation of the efficiency and accuracy that the new fire monitoring service succeeds against its precursor.

4. The data model stRDF and the query language stSPARQL

Let us now discuss the data model stRDF that is used to represent the datasets utilized by the fire monitoring service, and the query language stSPARQL that is used to manipulate such data and increase the quality of the original hotspot data.

stRDF is an extension of RDF for representing geospatial data that changes over time in RDF. For representing geospatial data stRDF relies on OGC standards such as Well-Known Text (WKT) and Geography Markup Language (GML). So, the data types `strdf:WKT` and `strdf:GML` are introduced. The union of these data types is defined by the data type `strdf:geometry`.

stRDF impose minimal new requirements to Semantic Web developers that want to represent spatial objects with stRDF; all they have to do is utilize a new literal data type. For example, a hotspot with a confidence value 0.5 and a specific spatial extent is represented in stRDF by the following triples:

```
noa:h1 rdf:type noa:Hotspot;
noa:hasConfidence "0.5"^^xsd:float;
noa:hasGeometry "POLYGON((22 38, ...))"^^strdf:WKT.
```

stSPARQL extends SPARQL 1.1 with functions from the "OpenGIS Simple Feature Access - Part 2: SQL Option" standard (OGC-SFA)¹⁴ for querying and updating stRDF data. For example, the function

```
xsd:boolean strdf:intersects
(strdf:geometry a, strdf:geometry b)
```

introduced in stSPARQL corresponds to the function `ST_Intersects` defined in the OGC-SFA standard. A user can ask for hotspots in a specific area by posing a query like the following:

```
SELECT * WHERE {
?h rdf:type noa:Hotspot;
noa:hasConfidence ?c;
noa:hasGeometry ?g.
FILTER(strdf:intersects(?g,
"POLYGON((20.95 38.15, ...))"^^strdf:WKT))}
```

In stSPARQL we have also introduced the following three *aggregate functions* that deal with geospatial data:

- `strdf:geometry strdf:union(set of strdf:geometry a)`
This aggregate function returns a geometry that is the union of the set of input geometries.
- `strdf:geometry strdf:intersection(set of strdf:geometry a)`
This aggregate function returns a geometry that is the intersection of the set of input geometries.
- `strdf:geometry strdf:extent(set of strdf:geometry a)`
This aggregate function returns a geometry that is the minimum bounding box of the set of input geometries.

These functions exist in all geospatial relational DBMS while the only geospatial RDF store that offers them is Strabon. We decided to include such kind of functions in stSPARQL and implement them

in Strabon in order to be able to serve real application scenarios with complex requirements in handling geospatial vector data.

5. Ontologies and linked geospatial data

In this section, we describe the geospatial data which are used in this application. We start by describing how hotspot data, derived by the processing chain described in Section 3, are transformed into RDF. Then, we describe some datasets that we transformed into RDF in order to combine them with hotspots. Finally, we describe two well known datasets of the Linked Open Data Cloud (LODC) which we used to further enrich our application.

5.1. The NOA ontology

To be able to transform the detected hotspots into RDF we have developed an ontology that models raw data and products which are being consumed or produced by NOA. This ontology which is called the NOA ontology is publicly available¹⁵ and is depicted in Fig. 2. The NOA ontology mainly consists of the classes `RawData`, `Shapefile`, and `Hotspot`. The instances of `RawData` represent files with raw data (e.g., satellite images) which are the input of the fire detecting processing chain. The output of this processing chain, which are some shapefiles, are represented by `Shapefile` instances. These files hold information about the coordinates of detected fire locations, the date and time of image acquisition, the level of reliability in the observations, the names of the processing chain and the sensor that was used for the acquisition. Every hotspot described by a shapefile is extracted and an instance of `Hotspot` is created to represent it in RDF.

To achieve interoperability the main classes of the NOA ontology have been defined as subclasses of corresponding classes of the SWEET ontology. Each instance of these three classes is connected with the satellite and the sensor from which it is derived, as well as with the date and time at which it was detected. Products (instances of `Hotspot` and `Shapefile`) are also associated with the method (type of processing chain) which was used for their production and with the organization which is responsible for the production (e.g., NOA). For each file (instances of `Shapefile` and `RawData`) its filename is stored. Finally, hotspots are connected with a representation of their geographical location, using a spatial literal as described in Section 4, and a confidence value representing the possibility that a pixel is a hotspot.

5.2. Compiled datasets

To be able to integrate hotspots with auxiliary geospatial information we transformed some datasets into RDF. We also published them as linked open data¹⁶ so that they can be reused for other relevant application. These datasets are about the landscape and coastline of Greece and the Greece administrative geography.

5.2.1. Greek landscape

The CORINE Land Cover (CLC) project is an activity of the European Environment Agency (EEA) that collects data regarding the land cover of European countries and publish them as shapefiles. The project uses a hierarchical scheme with three levels to describe land cover types and has published the landscape of European countries in shapefiles. The first level indicates the major categories of land cover on the planet (e.g., forests and semi-natural areas). The second level identifies more specific types of land cover

¹⁴ http://portal.opengeospatial.org/files/?artifact_id=25354.

¹⁵ <http://www.earthobservatory.eu/ontologies/noaOntology.owl>.

¹⁶ <http://linkedopendata.gr/>.

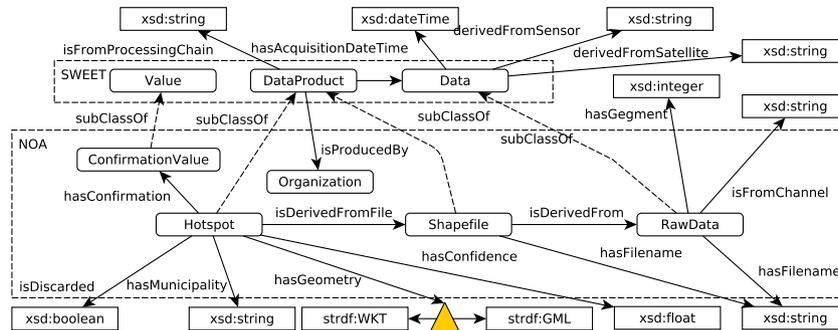


Fig. 2. The NOA ontology.

(e.g., forests) while the third level narrows down to a very specific characterization (e.g., coniferous forests). The landscape of Greece is available as a shapefile that is based on this classification scheme. This shapefile is transformed into RDF as follows. We have developed an ontology that captures this scheme in a class hierarchy with three levels. Every land cover type is represented with a class (e.g., *ConiferousForest*) and each URI representing a specific area in the shapefile is connected with an instance of these classes to denote the land use of this area. Additionally a data property of each area indicates its geospatial extent.

5.2.2. Greek coastline

This dataset originates from a shapefile, published on the Greek open government data portal,¹⁷ which describes the geometry of the coastline of Greece. For each area contained in this shapefile a unique URI is created and a spatial literal (which defines the geometry of the underlined area) is attributed to it.

5.2.3. Greek administrative geography

We developed an ontology that describes the administrative divisions of Greece (prefecture, municipality, district etc.). The ontology captures the hierarchy of administrative divisions (e.g., a *municipality* belongs to a *regional unit*) and has been populated with relevant data (formatted as CSV and SHP files) that are also available on the Greek open government data portal. Additionally, for each administrative unit in the ontology (e.g. a municipality) various pieces of information are available (e.g., population).

5.3. Datasets from Linked Open Data Cloud

To further enrich our dataset we used two well known datasets from Linked Open Data Cloud (LODC), namely OpenStreetMap¹⁸ (OSM) and GeoNames.¹⁹

OSM maintains a global editable map based on information provided by users. The respective ontology is derived mainly from OSM tags, i.e., attribute-value annotations of nodes, ways, and relations. OSM data is published as linked open data by the LinkedGeoData²⁰ (LGD) project.

GeoNames is a gazetteer that collects both spatial and thematic information for various place names around the world. GeoNames data is published through various Web services and as linked data as well. Also, specific properties are used to express inside, nearby, and part of relations.

These datasets can be correlated with hotspot data to derive more sophisticated information (e.g., big roads that may stop a fire). In addition, these datasets cover a large variety of geospatial entities, ranging from fine-grained geometric objects (e.g., fire stations, hospitals) to coarser ones like countries. So one can exploit the ability of Strabon to expose data in KML or GeoJSON and create a map by posing queries on these datasets and overlaying the results.

6. Hotspots semantic refinement

In Section 3 we described some drawbacks of the main fire detection process which can be solved by correlating information about hotspots with auxiliary information about underlined area. In this section, we describe a series of refinement steps, using stSPARQL updates, that enrich hotspot data with information about nearby municipalities and increase its accuracy by detecting and correcting false positives or omission errors.

Notably, the queries described below are sophisticated update statements that exploit extensively the expressivity of SPARQL 1.1 and stSPARQL (e.g., GROUP BY, HAVING, aggregations, OPTIONAL) to cover the needs of the real-time wildfire monitoring application of NOA.

6.1. Attribute enrichment

First, every hotspot is correlated with the Greek administrative geography dataset and is connected with a municipality where it is located. This is crucial information offered to decision makers and crisis managers for the optimal allocation of their firefighting resources. This refinement is performed by the following statement that inserts as a hotspot property the name of a municipality that spatially intersects with it.

```
INSERT {?h gag:hasMunicipality ?municipal}
WHERE { SELECT ?h (SAMPLE(?mLabel) AS ?municipal)
WHERE { ?h rdf:type noa:Hotspot;
noa:hasGeometry ?hGeo;
?m rdf:type gag:Municipality;
rdfs:label ?mLabel;
gag:hasGeometry ?mGeo.
FILTER(strdf:intersects(?hGeo, ?mGeo))}
GROUP BY ?h}
```

6.2. Consistency refinement

The thematic consistency of the hotspots generated by the processing chain is achieved by the refinement step that correlates them with auxiliary geospatial data. This is done by a series of stSPARQL updates on the hotspots by taking into account relevant datasets from the ones presented in Section 5. The first step is

¹⁷ <http://geodata.gov.gr/geodata/>.

¹⁸ <http://www.openstreetmap.org/>.

¹⁹ <http://www.geonames.org/>.

²⁰ <http://linkedgeodata.org/>.

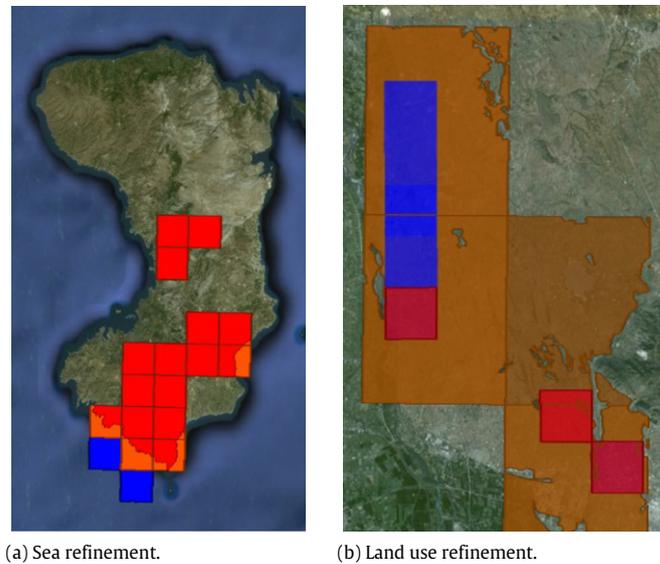


Fig. 3. Refinements results.

to discard all hotspots that lie in the sea, using the following stSPARQL.

```
INSERT {?h noa:isDiscarded "1"^^xsd:int}
WHERE { ?h rdf:type noa:Hotspot.
  OPTIONAL {?h gag:hasMunicipality ?municipality}
  FILTER(!bound(?municipality))}
```

The above statement inserts a new property for each retrieved hotspot, depicting that it is discarded. We are using the municipality update which is applied first. If a hotspot is not connected to a municipality (`!bound(?municipality)`), then we consider it as lying in the sea and correspondingly it is discarded. In a similar way, hotspots that are in the mainland, but lie in non-consistent land cover areas are also discarded. Using the Greek landscape dataset, the non-consistent classes are defined as (i) artificial surfaces, (ii) agricultural areas (arable land & permanent crops), and (iii) wetlands and water bodies. This operation is performed by the following stSPARQL update.

```
INSERT {?h noa:isDiscarded "1"^^xsd:int}
WHERE { SELECT ?h
  WHERE { ?h rdf:type noa:Hotspot;
    noa:hasGeometry ?hGeo.
    ?a rdf:type clc:ExcludeArea;
    clc:hasGeometry ?aGeo.
    FILTER(strdf:mbbIntersects(?hGeo, ?aGeo))}
  GROUP BY ?h ?hGeo
  HAVING strdf:contains(strdf:union(?aGeo), ?hGeo)}
```

This update selects all areas of the Greek landscape dataset which are defined as non-consistent. Then for each hotspot it groups the areas that may intersect with the hotspot and computes the union of these areas. Finally if the hotspot is fully contained in the union of non-consistent areas then it is considered false fire detection and is discarded. The update takes advantage of the function `strdf:mbbIntersects` of stSPARQL. This function returns true if the minimum bounding boxes of two geometries intersect with each other. This is an easy computation that facilitates quickly filtering only areas of the Greek landscape dataset that may intersect with a hotspot. Also, the union of these areas is computed by the spatial aggregate function `strdf:union`. Instead of removing invalid hotspots we preferred to keep them as discarded, and also keep some provenance information. So, an EO scientist can review

the original results of different fire detecting algorithms and perform analyses on both historical and newly acquired data.

Finally, to ensure the product visualization consistency, we also utilize the Greek Coastline dataset and keep only the part of a hotspot polygon that lies in land, and eliminate the part that lies in the sea. The result of these refinements is depicted in Fig. 3. In Fig. 3(a) hotspots lying in the sea are depicted blue, these hotspots are discarded. The part of hotspots overlapping the sea is depicted orange, these hotspots are updated so that the orange part of their geometry is deleted. The final result of both operations is depicted red. In Fig. 3(b) areas classified as non-consistent are depicted brown and hotspots that are discarded are depicted blue. The red hotspots are not discarded because they are not fully contained in non-consistent areas.

6.3. Temporal persistence

Apart from refining hotspots according to their location by comparing them with auxiliary geospatial information we also resolve temporal inconsistencies in the final product. The usual approach of hotspot detection at a given time is using a single image acquisition corresponding to that time, without taking into consideration hotspots and their locations in previous image acquisitions. Given the inaccuracies of existing hotspot detection algorithms, this single-scene processing approach results in some spatial and temporal inconsistencies between the different observations. A simple heuristic, which would result in significant noise removal, is to check the number of times a specific fire was detected over the same or near the same geographic location during the last hour(s), considering the observation's temporal and spatial persistence, and hence insert virtual hotspots with lower level of confidence. In our application we apply an stSPARQL update which finds all hotspots with the same geospatial extent in a time span of 30 min, examines the history of fire detections of each pixel, and based on a threshold of the average confidence value of real detections it inserts a virtual hotspot with confidence value lower than the real detections.

7. Evaluation of the service

In this section, we present an evaluation of the fire monitoring service which has been reimplemented using linked geospatial

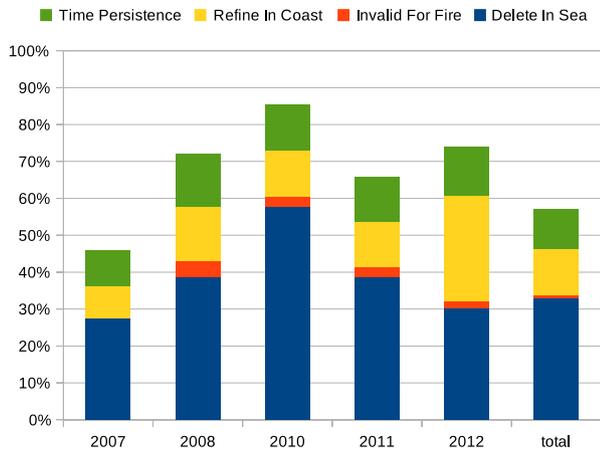


Fig. 4. Percentages of refined hotspots per year.

data and ontologies. First we present the ratio of original hotspots which were refined by the operations described in Section 6 in order to have an impression about how much the original dataset of hotspots can be improved by being combined with Linked Open Data. Additionally, we evaluate the performance of Strabon in executing the refinement updates of Section 6 for a real-time fire monitoring application. Since the application is considered to operate in real-time the refinements should not take more time than the time span between two sequential image acquisitions. In our case this time span is five minutes. Finally, we discuss the feedback we got from real users of the service.

7.1. Increasing accuracy

In order to estimate the utility of the hotspot refinements that we implemented, we applied the whole chain of our refinements to datasets containing hotspots derived from sensors MSG1 and MSG2 during the fire seasons of years 2007, 2008, 2010, 2011, and 2012 and we measured for each year the ratio of the original dataset that was refined by our stSPARQL updates.

Let us start with the detection of false alarms. We perform two operations that discard false alarms. The first discards hotspots that lie in the sea, the second discards hotspots that lie in non-consistent areas. All datasets contain about 60 000 original hotspots. The first operation found and discarded 20 000 hotspots. These are so many discarded hotspots because this operation does not discard only false fire detections that lie on the sea but also real fire detections that lie out of the Greek borders. The used satellite images cover a big area out of Greece where fires also occur and are detected. These emergencies are not handled by the Greek brigade or civil protection agency so they are also discarded to keep only hotspots of interest. The second operation found and discarded 600 hotspots. Second we refine the geometry of hotspots that overlap with the sea. This operation updated 7400 hotspots. Finally, we are trying to detect omission errors and insert virtual hotspots according to the time persistence of original hotspots. This operation inserted 6500 virtual hotspots. The results of this experiment are summarized in Fig. 4.

The implemented updates refine a big part (roughly 70%) of the original produced hotspots. Some of the updates (e.g., deleted hotspots or part of hotspots in the sea) correct detections which are for sure false positives. Other updates (e.g., inserting virtual hotspots according to the time persistence of hotspots) can be calibrated to be more or less aggressive and a lot of work should be done regarding the correct calibration of such updates or the definition of new ones. This is a work of domain experts who study how EO products can be combined with auxiliary geospatial

information and the added value which come about. However, we observe that using Semantic Web and combining original EO products with Linked Open Data can substantially improve the final results of an EO application. In our case this improvement is very important because it permits NOA to offer a significantly more precise map with areas at risk to decision makers and crisis managers. This improvement is also done automatically and enables the real-time monitoring of fire-fronts.

7.2. Refinement steps performance

In order to be able to offer an application for real-time monitoring, it is very important that the refinements that we apply can be performed quickly. The satellites that NOA uses to receive raw data send one acquisition every five or fifteen minutes. The refinements should be completed by this time period in order to be able to depict this live feed on a map at real time.

We have carried out several experiments in different machines for each version of the refinement queries. The datasets we used contained hotspots derived from sensors MSG1 and MSG2 during the fire seasons of the years 2007, 2008, 2010, 2011, and 2012 (up to 19/07/2012), combined with the Greek Administrative Geography dataset and the CLC dataset. The size of the dataset related to hotspot information is around 542,000 triples. The geometry of the Greek coastline was also included so that the respective spatial joins, as we described in the previous sections, could be performed. In this section we present two of our most recent experiments which were executed in one of our machines. The experimental environment and the results are described in the sections that follow.

7.2.1. Experimental setup

Our experiments were carried out on an Ubuntu 11.04 installation on two Intel Xeon E5620 with 12MB L3 cache running at 2.4 GHz. The system has 48 GB of RAM and 4 disks using RAID configuration as two mirrored sets in a striped set (RAID level 1+0). The metric we used to measure performance is the response time for each update posed for the respective operation by measuring the elapsed time from query submission until a complete iteration over each query's results had been completed.

7.2.2. Experimental results

For the experiments we present herein, the following datasets were used:

- Hotspots derived from the sensor MSG1 during the fire period of the year 2012
- Greek administrative geography
- The coastline of Greece
- The landscape of Greece

Sensor MSG1 detects hotspots every five minutes. The refinement operations described in Section 6 were applied to the products of each acquisition and the response time for each operation was measured. The results of this experiment are shown in Fig. 5(a).

In Fig. 5(a), we observe that all operations are executed efficiently, mostly in less than a second, except for the operation of associating each detected hotspots with the municipality they belong to. This operation is labeled as "Municipalities" in the figure and although for the most cases the query processing time does not exceed the two seconds, there are cases where it needs four seconds to be completed. Even in these cases, the performance of the system is satisfactory.

The second experiment we present was executed in the same experimental environment. The datasets used are the following:

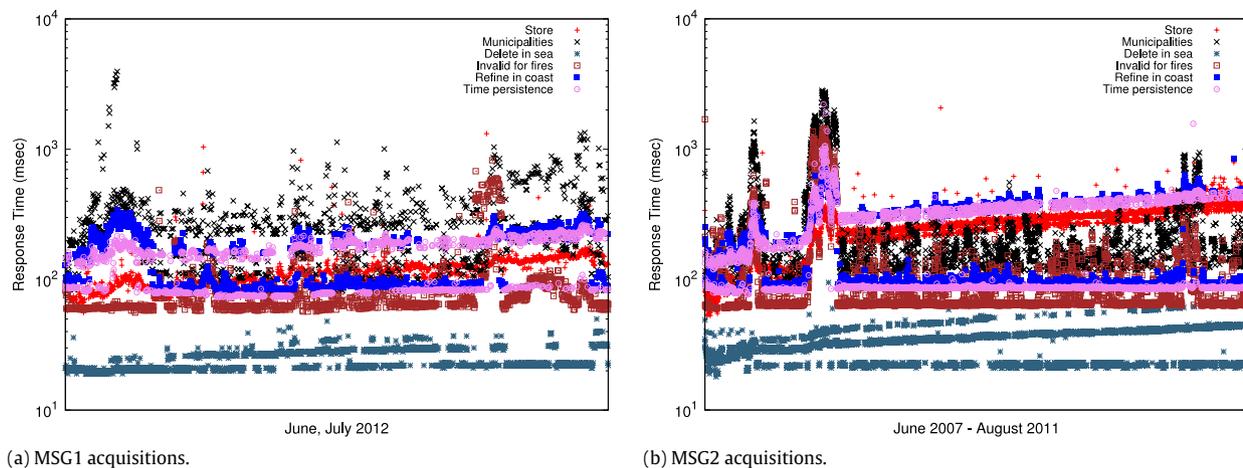


Fig. 5. Response times for each MSG1 and MSG2 acquisitions.

1. Hotspots derived from the sensor MSG2 during the fire periods of the years 2007, 2008, 2010, and 2011
2. Greek administrative geography
3. The coastline of Greece
4. The landscape of Greece

Sensor MSG2 detects hotspots every fifteen minutes. Similarly, refinement operations are performed to the MSG2 acquisitions and the response time of each respective query is measured, presented in Fig. 5(b).

In Fig. 5(b), similarly to Fig. 5(a), we observe some spikes in the response time of Strabon while inserting the Greek Administrative Geography information. An additional observation is that query processing time grows in acquisitions with a larger number of hotspots. After this preliminary evaluation, we observe that the performance of Strabon is satisfactory, given that the sensors MSG1 and MSG2 provide an acquisition every five and fifteen minutes respectively.

7.2.3. User feedback

The service we developed has a web based interface tailored for decision makers and crisis managers interested in real time wildfire monitoring. While new hotspots are detected the application animates the evolution of a fire-front along with useful auxiliary information (e.g., affected municipalities). Refinement operations are hidden from the end-user for simplicity. Also, a user-friendly option is offered to retrieve and watch past wildfires. The application was used during the fire seasons of 2012 and 2013 by the Greek Civil Protection Agency, the Fire Brigade and the Army both during the fire events for strategy planning and after them to assess the strategies that were followed. The service has also been thoroughly tested during the 3rd user workshop of the project TELEIOS. The spectrum of users which participated this test includes both end users which use fire monitoring products on an operational basis (e.g., civil protection agencies) and stakeholders from the EO and IT communities. In general the collected feedback was very encouraging: most users found the applications very useful, specifically when it concerned stakeholders that need fire monitoring products as part of their daily work practice (e.g. Greek Ministry of Environment, Energy and Climate Change, Italian civil defense agency, or foresters in local administrative units). The value of applying semantic queries for the thematic refinement of the hotspot products has been appreciated by EO community. It was also shown, and it is envisaged, that rapid mapping applications can be easily deployed using semantic technologies with distributed data. EO service providers are enabled to use

Linked Data together with fire products (e.g. hotspots or burnt areas) to further enhance their products' value and create thematic maps that can be used by stakeholders. Greek government data (geodata.gov.gr), Administrative Geography of Greece, Open Street Map, Wikipedia, and Gazetteers (e.g. Geonames) can be incorporated in a couple of simple steps.

8. Conclusion

In this paper we presented the wildfire monitoring service that we have developed in the European project TELEIOS and the Greek project SWeFS highlighting the contributions of ontologies and linked geospatial data.

Today EO practices rely on processing EO data as it is. By using Semantic Web technologies we achieve to automatically compare EO data with auxiliary data and derive more complex kinds of information (e.g., fire in sea) that basic EO practices (e.g., hotspots detection) cannot capture. The service uses a variety of information sources under different ownership and control. The core of our dataset contains fire related products extracted from satellite images owned by NOA enriched with data published by the European Environment Agency and by the Greek government and with data published in LODC. Representing this data using RDF and using ontologies to conceptualize the different structure of each dataset enabled us to seamlessly homogenize highly heterogeneous data. New ontologies and datasets encoded in RDF have been created for this purpose and have been combined with existing open datasets. Thus, complex refinement operations can be easily expressed in SPARQL and integrated in the workflow of NOA to produce validated results in real-time. The processing chain of the service is executed efficiently on top of Strabon, and the user is able to access real-time and archived hotspot data via a user-friendly interface. The same principles can be adopted in other aspects of wildfire management or other applications. For example, in the context of SWeFS we are currently extending the service with fire prediction models. This requires creating and integrating more linked geospatial data about weather, human activities and flammability of specific regions.

References

- [1] K. Kyzirakos, M. Karpathiotakis, M. Koubarakis, Strabon: a semantic geospatial DBMS, in: Proceedings of the 11th International Semantic Web Conference, 2012.
- [2] K. Bereta, P. Smeros, M. Koubarakis, Representation and querying of valid time of triples in linked geospatial data, in: ESWC, 2013.

- [3] K. Kyzirakos, M. Karpathiotakis, G. Garbis, C. Nikolaou, K. Bereta, I. Papoutsis, T. Herekakis, D. Mihail, M. Koubarakis, C. Kontoes, Wildfire monitoring using satellite images, ontologies and linked geospatial data, in: 10th Semantic Web Challenge, 2012.
- [4] S. Auer, J. Lehmann, S. Hellmann, LinkedGeoData: adding a spatial dimension to the web of data, in: ISWC'09, 2009, pp. 731–746.
- [5] A. de León, V. Saquicela, L.M. Vilches, B. Villazón-Terrazas, F. Priyatna, O. Corcho, Geographical linked data: a Spanish use case, in: I-SEMANTICS, 2010.
- [6] M. Podwyszynski, Knowledge-based search for Earth Observation products, Ph.D. Thesis, University of Passau, 2009.
- [7] R. Raskin, M. Pan, Knowledge representation in the semantic web for earth and environmental terminology (sweet), *Computers & Geosciences* 31 (9) (2005) 1119–1125.
- [8] D.L. McGuinness, P. Fox, L. Cinquini, P. West, J. Garcia, J.L. Benedict, D. Middleton, The virtual solar-terrestrial observatory: a deployed semantic web application case study for scientific research, in: AAAI, 2007, pp. 1730–1737.
- [9] C. Carlino, S.D. Elia, A.D. Vecchia, M. Iacovella, M. Iapaolo, C.M. Scalzo, F. Verdino, On sharing Earth Observation concepts via ontology, in: ESA-EUSC, 2008.
- [10] N.I. Sifakis, C. Iossifidis, C. Kontoes, I. Keramitsoglou, Wildfire detection and tracking over Greece using MSG-SEVIRI satellite data, *Remote Sensing* 3 (3) (2011) 524–538.
- [11] Open Geospatial Consortium, OGC GeoSPARQL—a geographic query language for RDF data, OGC Implementation Standard, 2012.
- [12] M. Koubarakis, M. Karpathiotakis, K. Kyzirakos, C. Nikolaou, M. Sioutis, Data models and query languages for linked geospatial data, in: Reasoning Web. Semantic Technologies for Advanced Query Answering, 2012.
- [13] M. Koubarakis, C. Kontoes, S. Manegold, M. Karpathiotakis, K. Kyzirakos, K. Bereta, G. Garbis, C. Nikolaou, D. Michail, I. Papoutsis, T. Herekakis, M. Ivanova, Y. Zhang, H. Pirk, M. Kersten, K. Dogani, S. Giannakopoulou, P. Smeros, Real-time wildfire monitoring using scientific database and linked data technologies, in: Proceedings of the 16th International Conference on Extending Database Technology, 2013, pp. 33–38.