Real-Time Wildfire Monitoring Using Scientific Database and Linked Data Technologies

Manolis Koubarakis
National and Kapodistrian
University of Athens, Greece
koubarak@di.uoa.gr

Charalambos Kontoes
National Observatory of
Athens, Greece
kontoes@noa.gr

Stefan Manegold
Centrum Wiskunde &
Informatica, Amsterdam, NLD
Stefan.Manegold@cwi.nl

ABSTRACT

We present a real-time wildfire monitoring service that exploits satellite images and linked geospatial data to detect hotspots and monitor the evolution of fire fronts. The service makes heavy use of scientific database technologies (array databases, SciQL, data vaults) and linked data technologies (ontologies, linked geospatial data, stSPARQL) and is implemented on top of MonetDB and Strabon. The service is now operational at the National Observatory of Athens and has been used during the previous summer by emergency managers monitoring wildfires in Greece.

Categories and Subject Descriptors

H.2.8 [**Database Management**]: Database Applications—
scientific, spatial databases and GIS

General Terms

Management, Languages

Keywords

Earth observation, Scientific databases, Semantic web, Linked data, GIS data

1. INTRODUCTION

Wildfire monitoring and management in Europe, and in the Mediterranean region in particular, is of paramount importance. Almost every summer massive forest wildfires break out in several areas, leaving behind severe destruction in forested and agricultural land, infrastructure and private property, and losses of human lives.

European initiatives in the area of Earth Observation (EO) like the Global Monitoring for Environment and Security initiative (GMES)[1] have therefore undertaken an active role in the area of fire monitoring and management in Europe, and supported the development of relevant European

---

[1] http://gmes.info/

operational infrastructures through projects such as linkER (Supporting the implementation of an operational GMES service in the field of emergency management) and SAFER (Services and Applications For Emergency Response)[2].

In the framework of SAFER, the National Observatory of Athens (NOA) has been developing a real-time fire hotspot detection service for effectively monitoring a fire-front. The service depends on the real-time processing of satellite images of different spectral and spatial resolutions in combination with auxiliary geo-information layers (land use/land cover data, administrative boundaries, and roads and infrastructure networks data). The outputs of the service are validated fire-related products (e.g., hotspot and burnt area maps) for Southern Europe (Spain, France, Italy, Portugal, and Greece).

In this paper we discuss how NOA has redeveloped its entire real-time fire monitoring service using scientific database management and linked data technologies developed in research projects TELEIOS and SWeFS. TELEIOS[3] is a European research project that addresses the need for scalable access to petabytes of Earth Observation data and the effective discovery of knowledge hidden in them. It follows a database approach to the development of Earth Observation applications and pioneers the use of the following state-of-the-art results:

- The query language *SciQL*, an SQL-based query language for scientific applications with arrays as first class citizens [23].

- The *data vault*, a mechanism that provides a true symbiosis between a DBMS and existing (remote) file-based repositories such as the ones used in EO applications [7]. The data vault keeps the data in its original format and place, while at the same time enables transparent data and metadata access and analysis using the SciQL query language. SciQL and the data vault mechanism are implemented in the well-known column store MonetDB[4].

- Publicly available *linked data*[5], especially geospatial ones such as OpenStreetMap[6], Geonames[7], etc.

- The model stRDF, an extension of the W3C standard RDF that allows the representation of geospatial data that changes over time [14]. stRDF is accompanied by stSPARQL, an 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 temporal and geospatial data and are implemented in open source geospatial RDF store Strabon[8].

The contributions of this paper are the following:

- We present a wildfire monitoring service that has been developed entirely in the European project TELEIOS using the scientific database and linked data technologies presented above. The service is currently operational at NOA and has been used during the previous summer by emergency managers monitoring wildfires in Greece. More recently, in the context of project SWeFS (Sensor Web Fire Shield, a recent Greek research project investigating the use of sensor networks in fire monitoring), NOA developed a public Web interface for this service[9].

- 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 and linked data technologies developed in project TELEIOS. We discuss the pre-TELEIOS version of the service, and how scientific database and linked data technologies 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 accuracy of the fire detection processing chain. We also present a performance evaluation of the two main parts of the service demonstrating the technologies discussed in the paper: the fire detection part where SciQL over MonetDB is used and the refinement operations that use linked geospatial data and stSPARQL over Strabon.

The rest of the paper is organized as follows. Section 2 presents the fire monitoring service operational at NOA before TELEIOS, and Section 3 explains in detail how this service was improved using scientific database and linked data technologies. Then, Section 4 evaluates the developed service. Last, Section 5 discusses related work and Section 6 concludes the paper.

2. THE FIRE MONITORING APPLICATION OF NOA

The wildfire monitoring service of NOA is based on the use of satellite images originating from the SEVIRI (Spinning Enhanced Visible and Infrared Imager) sensor on top of the Meteosat Second Generation satellites MSG-1 and MSG-2. Since 2007, NOA operates an MSG/SEVIRI acquisition station, and has been systematically archiving raw satellite
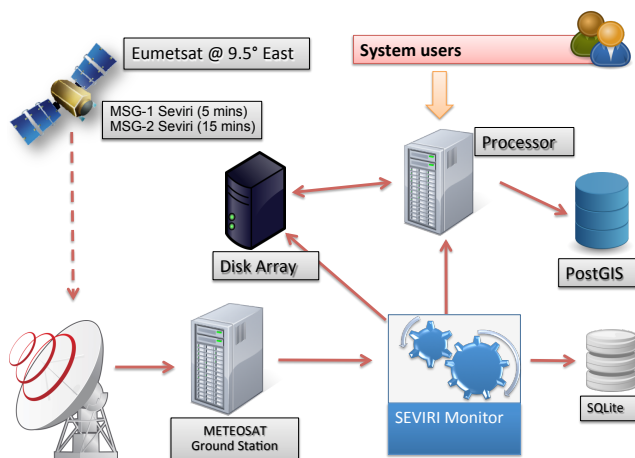
Figure 1: The NOA fire monitoring service

images on a 5 and 15 minutes basis, the respective temporal resolutions of MSG-1 and MSG-2. The archives of raw imagery are now in the order of 2 Terabytes, corresponding to the summer fire periods of the last five years.

The fire monitoring service active in NOA *before TELEIOS* is presented graphically in Figure 1 and can be summarized as follows:

(1) The ground-based receiving antenna collects all spectral bands from MSG-1 and MSG-2 every 5 and 15 minutes respectively.

(2) The raw datasets are decoded and temporarily stored in the METEOSAT Ground Station as wavelet compressed images.

(3) The application SEVIRI Monitor, written in Python, manages the data stream in real-time by offering the following functionality:

1. Extract and store the raw file metadata in an SQLite database. This metadata describes the type of sensor, the acquisition time, the spectral bands captured, and other related parameters. Such a step is required as one image comprises multiple raw files, which might arrive out-of-order.

2. Filter the raw data files, disregarding non-applicable data for the fire monitoring scenario, and dispatch them to a dedicated disk array for permanent storage.

3. Remotely trigger the processing chain by transferring the appropriate spectral bands via FTP to a dedicated machine and initiating the distinct processing steps described in [20]. These steps are: (i) cropping the image to keep only the area of interest, (ii) georeferencing to the geodetic reference system used in Greece (HGRS 87), (iii) classifying the image pixels as "fire" or "non-fire" using the algorithm of [5], and finally (iv) exporting the final product to raster and vector formats (ESRI shapefiles).

4. Dispatch the derived products to the disk array and additionally store them to a PostGIS database system.

The products that are stored in PostGIS cover the geographical area of Greece and are disseminated to the end
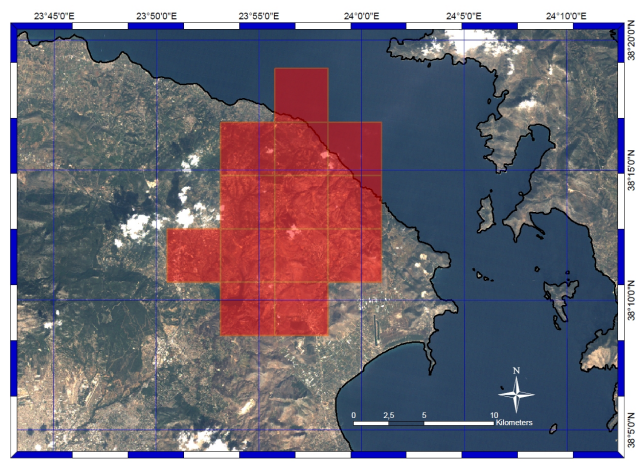
Figure 2: A detailed vector representation of fires at Attica, Greece, in 2010

user community (civil protection agencies, regional authorities, and decision makers) through a web application that uses the interoperable tool GeoServer[10] for sharing geospatial data.

The fire pixels derived by the above processing chain have dimensions equal to the sensor's spatial resolution, in this case nearly 4x4 km. Thus, MSG/SEVIRI is a low resolution observational system, compared to other very high resolution sensors with similar fire detection capabilities (e.g., WorldView-2 at 0.5 m, Quickbird at 2.4 m, IKONOS at 4 m or Formosat-2 at 8 m), high resolution sensors (e.g., Spot-5 at 10 m and Landsat-5 TM at 30 m), or medium resolution sensors (e.g., MODIS Terra and Aqua with 2 bands at 250 m, 5 bands at 500 m and 29 bands at 1 km). However, the unique advantage of MSG/SEVIRI is its geostationary orbit, which allows for a very high observational frequency (5-15 minutes) over the same area of interest. Other satellite platforms with better spatial resolution are forced to undertake orbits that are closer to the earth, which considerably reduces their revisit time. For example, Aqua MODIS, with its near-polar orbit, passes over Greece twice a day (at 00:30 and 11:30) and the same applies for Terra MODIS (at 9:30 and 20:30). Another important advantage of the MSG/SEVIRI sensor is that its sensitivity is not at all affected by its low spatial resolution, i.e., it is not necessary for an entire 4x4 km pixel to be "on fire" to detect the corresponding hotspot. A small portion of a pixel that exhibits increased temperature will suffice to detect a wildfire. In conclusion, the increased five minute temporal resolution of MSG/SEVIRI is an exceptional capability that allows civil protection operators to have an almost real-time overview of the situation in terms of forest wildfires. A typical example that highlights the usefulness of the hotspot products in Greece is shown in Figure 2. Additionally, another comparative advantage of MSG/SEVIRI with respect to higher spatial resolution sensors, is the increased field of view, i.e., its footprint on the Earth. While, for high and very high resolution sensors, this is limited to 10-200 km, MSG/SEVIRI covers with a single image most of Europe and Africa, allowing for applications with a global coverage to be developed.

[10]http://geoserver.org/

One of the goals of TELEIOS is to improve the hotspot detection and fire monitoring service of NOA described above. The main issues that need to be addressed are the following:

- The thematic accuracy of the generated products has to be refined in a clear and systematic way, to ensure the reliability and transferability of the service to other geographic areas. The main problem with the current thematic accuracy is the existence of false alarms and omission errors in the fire detection technique that relate to the following scenarios:

  - Cases of hotspots occurring in the sea or in locations represented by fully inconsistent land use/land cover classes, like urban or permanent agriculture areas. If these hotspots correspond to real fires, these fires occur in the vicinity of coasts or urban areas, but due to the low spatial pixel resolution of the MSG/SEVIRI instrument and errors in image geo-referencing, the hotspots wrongly appear to be over inconsistent underlying land use/land cover classes. This type of error could be easily corrected if derived hotspot products are compared with auxiliary GIS layers by a NOA operator. However, this would certainly require time for manual GIS layer integration and visual interpretation, an operation that is not possible in the available 5 minute time frame.

  - Cases of hotspots located outside forested areas. These can be false fire detections due to known problems with existing hotspot detection algorithms (e.g., inappropriate fire/no-fire thresholds in the algorithm of [5]). They can also be real cases of fires located in big agricultural plains that are started by farmers as part of their agricultural practices. Whichever the case, they are not real forest fires, and they are not emergency situations to be handled. This type of noisy information could be avoided if derived hotspot products are combined together with land use/land cover information, again an operation that cannot be done manually in the 5 minute time frame.

  - Spatial and temporal inconsistencies in the final product. Today hotspot detection at a given time is done by using a single image acquisition corresponding to that time, without taking into consideration hotspots and their locations in previous image acquisitions, e.g., hotspots detected during the last 1 to 2 hours. Given the inaccuracies of existing hotspot detection algorithms [5], 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 attributing a level of confidence to each detected pixel.

- The need to generate added-value thematic maps combining diverse information sources. As a service

provider NOA aims at delivering to the end-user community reliable and comprehensive information for fire related emergency situations. Although vector shape-files are useful for analysis in the aftermath of a crisis, in real-time emergency response scenarios, civil protection agencies and local firefighting teams find it more useful to refer to a map depicting the active fire-front and its evolution in the last hours/days and identify nearby crucial infrastructure (hospitals, schools, industrial sites, fire hydrants, etc.). This is of paramount importance for the effective allocation of resources during the crisis. Therefore, a desired functionality that is currently missing is automatic map generation enriched with easily accessible geo-information layers.

- Dispersion of the various processes of the fire monitoring service in many machines and pieces of software makes it difficult for NOA to keep all functionalities synchronized. There is no consistent management policy, but various independent components (as seen in Figure 1) that are glued together with the Python-based application SEVIRI Monitor. This in not a good solution for effectively managing the raw satellite imagery, the generated products and the static GIS layers. A more robust and user-friendly management system is needed that will allow the integration and customization of the available capacities.

3. IMPROVING THE FIRE MONITORING APPLICATION OF NOA USING TELEIOS TECHNOLOGIES

In this section we describe the implementation of the fire monitoring service of NOA using TELEIOS technologies. Let us describe briefly the improvements that have been done. First, loading can be performed without any pre-processing of raw data because the data vault module has been developed that transforms input data into SciQL arrays. Secondly, the processing chain and other operations such as georeferencing, cropping images and classification of measurements have been implemented using SciQL. This leads to more expressive queries that can easily be changed if needed. Finally, using stSPARQL and combining standard products with auxiliary data enables a user to easily create added-value thematic maps and increase their thematic accuracy.

Figure 3 depicts the new fire monitoring application of NOA developed in TELEIOS. The system consists of the following parts:

- The data vault, which is responsible for the ingestion policy and enables the efficient access to large archives of image data and metadata in a fully transparent way, without worrying about their format, size and location.

- The back-end of the system. The back-end relies on MonetDB for two tasks: (i) the implementation of the hotspot detection processing chain (using the SciQL front-end) and (ii) the evaluation of semantic queries for improving the accuracy of the product shapefiles and generating thematic maps (using an stSPARQL front-end, i.e., Strabon).

- A geospatial ontology, which links the generated hotspot products with stationary GIS data (Corine
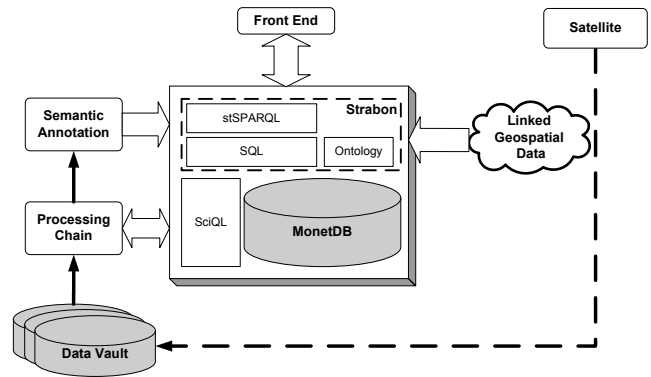

Figure 3: The improved fire monitoring service

Land Cover, Coastline, Greek Administrative Geography), and with linked geospatial data available on the web (LinkedGeoData, GeoNames). This ontology is expressed in OWL. This ontology is described in more detail in Section 3.2.1 below.

- The front-end interface, for controlling the back-end functionality with user-friendly tools, and disseminating the products to the end-user community. A visual query builder is currently being developed as well to allow NOA personnel to express complex stSPARQL queries easily.

Let us now describe in more detail two of the more interesting, from a database perspective, functionalities of the fire monitoring service: the implementation of the hotspot detection processing chain using data vaults and SciQL, and the improvement of the thematic accuracy of the hotspot products and production of thematic maps using stSPARQL.

3.1 The processing chain

The processing chain as described in Section 2 comprises the following submodules: (a) ingestion, (b) cropping, (c) georeference, (d) classification, and (e) output generation. All submodules are implemented inside the MonetDB DBMS using SciQL. In the following we describe each of them in detail.

3.1.1 Loading

One of the major issues that arise when dealing with earth observation data is the abundance of available file formats. In this particular application the input format is High Rate Information Transmission (HRIT) or Low Rate Information Transmission (LRIT). These are the CGMS standards agreed upon by satellite operators for the dissemination of digital data originating from geostationary satellites to users via direct broadcast. Loading such data requires an external program that transforms the original satellite image format into a representation as table or array that the DBMS can handle. The reason therefore is that DBMSs in general do not know anything about any external file formats. Thus, the knowledge of how to convert a given file format into a relational table or an array needs to be available and kept outside the DBMS. This can be a major hurdle, not only in terms of inconvenience for the user, but also in terms of performance. All external files that are to be loaded into the

database first need to be converted entirely to the appropriate format required for query processing at a subsequent stage, even if not all files or not all data of each file are actually required for answering queries. As a first solution, we exploited the extensibility of MonetDB and developed an extension module that can load a satellite image given as HRIT file into an SQL table or SciQL array. The module provides a user-defined SQL/SciQL function "HRIT_load_image()" that returns the table/array. The function expects as parameters URIs indicating the location of the respective image files. A similar function, returning an SQL table, is responsible for reading image metadata such as number of rows, columns and bands.

The Data Vault [7] goes one step further, into a more generic solution that addresses the principal problem of ingestion of data from external file formats into database tables or arrays. The main idea of the Data Vault is to make the DBMS aware of external file formats and keep the knowledge how to convert data from external file formats into database tables or arrays inside the database. With this, inserting external files (of known format) into the database basically consists of copying the files "as-is" into a directory that is under exclusive control of the database. Only after issuing queries that actually access data of a certain file, the DBMS will take care of loading the data from the file into the respective table or array.

3.1.2 Cropping and georeference

The classification algorithm used for the fire monitoring application requires as input IR bands 3.9 and 10.8. Following the data loading step, both bands are stored into a SciQL array. The input of these two bands is subsequently transformed into temperature values. Thus, it is safe to assume that the input looks like the arrays created by the following SciQL statements:

```
CREATE ARRAY hrit_T039_image_array
          (x INTEGER DIMENSION, y INTEGER DIMENSION, v FLOAT);
CREATE ARRAY hrit_T108_image_array
          (x INTEGER DIMENSION, y INTEGER DIMENSION, v FLOAT);
```

NOA is interested only in a specific part of the image that is received from the satellite. Cropping only the relevant parts of the image, which contain the area of interest is performed in a straightforward manner using a range query. Cropping the image early on, significantly reduces the input size of the remaining image processing operations and thus the time required for the execution of the processing chain.

After the cropping operation the algorithm georeferences the image by transforming it to a new image where the location of each pixel is well known. The MSG satellite is geostationary, so in effect remains stationary above a point on the earth. Thus, after the necessary transformation has been calculated by hand, every image can be transformed in exactly the same way. The NOA application resamples the image into a slightly larger size and applies a two degree polynomial in order to map pixels of the old image to the pixels of the new image. The coefficients of the polynomial as well as the target image dimensions are all precalculated. The implementation of these operations is expressed in a very concise way using SciQL. Any occasional drift of the satellite, over a long period of time, is handled by recalculating the transformation coefficients and updating the corresponding SciQL script.

```
CREATE ARRAY hrit_T039_image_array
          (x INTEGER DIMENSION, y INTEGER DIMENSION, v FLOAT);
CREATE ARRAY hrit_T108_image_array
          (x INTEGER DIMENSION, y INTEGER DIMENSION, v FLOAT);

SELECT [x], [y],
 CASE
  WHEN v039 > 310 AND v039 - v108 > 10 AND v039_std_dev > 4    AND
       v108_std_dev < 2
   THEN 2
  WHEN v039 > 310 AND v039 - v108 >  8 AND v039_std_dev > 2.5 AND
       v108_std_dev < 2
   THEN 1
  ELSE  0
 END AS confidence
FROM (
 SELECT [x], [y], v039, v108,
  SQRT( v039_sqr_mean - v039_mean * v039_mean ) AS v039_std_dev,
  SQRT( v108_sqr_mean - v108_mean * v108_mean ) AS v108_std_dev
 FROM (
  SELECT [x], [y], v039, v108,
   AVG( v039 ) AS v039_mean, AVG( v039 * v039 ) AS v039_sqr_mean,
   AVG( v108 ) AS v018_mean, AVG( v108 * v108 ) AS v108_sqr_mean
  FROM (
   SELECT [T039.x], [T039.y], T039.v AS v039, T108.v AS v108
   FROM hrit_T039_image_array AS T039
   JOIN hrit_T108_image_array AS T108
     ON T039.x = T108.x AND T039.y = T108.y
  ) AS image_array
  GROUP BY image_array[x-1:x+2][y-1:y+2]
 ) AS tmp1;
) AS tmp2
```

Figure 4: Hotspot detection algorithm in SciQL

3.1.3 Classification

The fire classification module of the processing chain receives as input the cropped, resampled and georeferenced image with the two pixel temperatures, each derived from one band. The algorithm [5] slides a 3x3 window over every pixel of the image and computes the standard deviation of the temperatures inside the window. Figure 4 shows the classification algorithm in SciQL.

The query first computes for each pixel the standard deviation for each of the two bands. It uses the structural grouping capabilities of the SciQL, in order to gather for each pixel the values of its neighbors inside a 3x3 window. The classification process outputs a per-pixel value of 0, 1, or 2. The value 2 denotes fire, value 1 denotes potential fire while 0 denotes no fire. The decision is based on thresholding. A set of 4 thresholds, one for the temperature of the IR 3.9 band, one for the difference between the temperatures of the IR 3.9 and the IR 10.8 band, and two for the standard deviations of the two temperatures, are used for the classification of the pixel. The actual choice of thresholds used in the figure are those for an image acquired during the day. During the night a different set of thresholds is used. *Day* is defined with a local solar zenith angle lower than $70°$ while *night* with a solar zenith angle of higher than $90°$. For solar zenith angles between $70°$ and $90°$ the thresholds are linearly interpolated. While not shown in the query, the solar zenith angle is computed on a per-pixel basis given the image acquisition timestamp and the exact location of the pixel, which is already known after the georefencing step.

3.1.4 Output generation

The final output is produced by a SciQL query, which selects pixels classified as fire or potential fire and outputs a POLYGON description in Well-known Text (WKT) for-

mat. The location of each pixel is already known after the georeference step and its shape is a 4x4 km square.

3.2 stRDF and stSPARQL in the NOA application

In TELEIOS, standard products produced by processing chains of EO data centers can be combined with auxiliary data to offer to users functionalities that go beyond the ones currently available to them. In this section we give concrete examples of this by showing how to improve the outputs of the hotspot detection processing chain discussed above. We start by presenting an ontology for annotating NOA standard products. Then, we present one by one all the geospatial datasets utilized in the fire monitoring application. Last, we present stSPARQL queries that improve the accuracy of NOA standard products and enable us to produce rich thematic maps.

3.2.1 Ontology for NOA standard products

To annotate semantically standard products produced by the hotspot detection processing chain of NOA, we have developed an ontology (called the NOA ontology from now on). The ontology is encoded in OWL and it is publicly available[11]. The main classes of the current version of the NOA ontology, which is depicted graphically in Figure 5, are `RawData`, `Shapefile`, and `Hotspot`, which represent files with raw data (e.g., sensor measurements), ESRI shapefiles, which are the outputs of the hotspot detection processing chain and hotspots, which are extracted from shapefiles, respectively. For interoperability purposes, these classes have been defined as subclasses of corresponding classes of the SWEET[12] ontology. Each instance of these three classes is annotated 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 annotated with the method (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 additionally annotated with a spatial literal and a numeric (float) literal. The former corresponds to the region (pixel) where the hotspot lies and the latter indicates the confidence that a pixel is a hotspot.

3.2.2 Hotspot data

The result of the processing chain described in Section 3.1 is a collection of shapefiles. 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, and the names of the processing chain and the sensor that was used for the acquisition. To be able to query these shapefiles using stSPARQL and combine them with linked data freely available on the web, the produced shapefiles are first transformed into RDF. Due to the simple form of the shapefiles, each attribute of a shapefile becomes a predicate, each attribute value becomes an object and finally a subject is created as a unique URI identifying the corresponding hotspot. The following triples[13] are an ex-
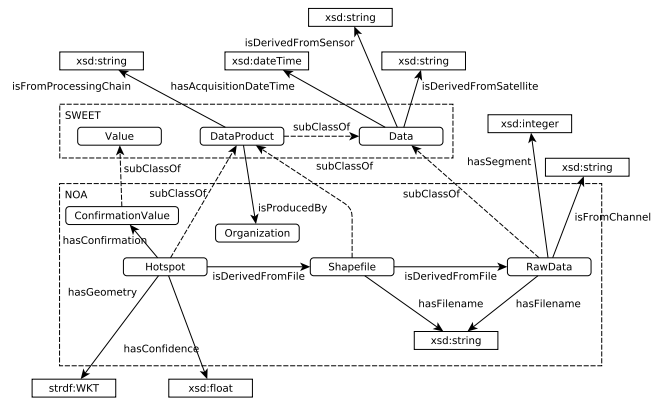
Figure 5: An ontology for NOA products

ample of such information about a hotspot.

```
noa:Hotspot_1 a noa:Hotspot ;
 noa:hasAcquisitionDateTime "2007-08-24T18:15:00"^^xsd:dateTime;
 noa:hasConfidence 1.0 ;
 noa:hasConfirmation noa:confirmed ;
 strdf:hasGeometry "POLYGON ((21.52 37.91,21.57 37.91,21.56
  37.88,21.56 37.88,21.52 37.87,21.52 37.91))"^^strdf:geometry ;
 noa:isDerivedFromSensor "MSG2"^^xsd:string ;
 noa:isProducedBy noa:noa ;
 noa:isFromProcessingChain "cloud-masked"^^xsd:string .
```

3.2.3 Auxiliary data

We now give a short description of the auxiliary datasets utilized in the fire monitoring application.

Corine Land Cover. The Corine Land Cover project[14] is an activity of the European Environment Agency that collects data regarding the land cover of European countries. The project uses a hierarchical scheme with three levels to describe land cover. 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, e.g., forests, while the third level narrows down to a very specific characterization, e.g., coniferous forests. The land cover of Greece is available as an ESRI shapefile that is based on this classification scheme. This shapefile is transformed in RDF as follows. Every land cover type is represented with a class (e.g., `ConiferousForest`), and the hierarchy of land cover types is expressed with the respective class taxonomy. For each specific area in the shapefile, a unique URI is created and it is connected with an instance of the third level. Additionally, the geometry of each area is represented by a spatial literal. Some sample triples representing such an area are shown below.

```
clc:Area_45 a clc:Area ;
    strdf:hasGeometry "POLYGON ((22.07 40.62, ...,
                        22.07 40.62))"^^strdf:geometry ;
    clc:hasLandUse clc:coniferousForest .
```

Coastline of Greece. This is an ESRI shapefile describing the geometry of the coastline of Greece. For each polygon contained in the shapefile, a unique URI is created and a spatial literal is attributed to it. The spatial literal corresponds to the closed polygon, which defines the underlined area. For example:

```
coast:Coastline_1 a coast:Coastline ;
    strdf:hasGeometry "POLYGON ((24.12 34.80, ...,
                                 24.12 34.80))"^^strdf:geometry .
```

Greek Administrative Geography. This is an ontology
that describes the administrative divisions of Greece (pre-
fecture, municipality, district, etc.). The ontology has been
populated with relevant data that are available in Greek
open government data portal[15]. For each administrative
unit in the ontology (e.g., a municipality) various pieces of
information are available (e.g., population and geographical
boundaries). The following is a small example of such kind
of information for the municipality of Athens.

```
gag:munAthens a gag:Municipality ;
    rdfs:label "Athens" ;
    gag:hasPopulation "655780"^^xsd:integer ;
    gag:isPartOf gag:preAttica ;
    strdf:hasGeometry "POLYGON((23.74,38.03, ...,
                               23.74,38.03))"^^strdf:geometry .
```

LinkedGeoData. LinkedGeoData (LGD)[16] is a project fo-
cused on publishing OpenStreetMap (OSM)[17] data as linked
data. OSM maintains a global editable map that depends on
users to provide the information needed for its improvement
and evolution. The respective ontology is derived mainly
from OSM tags, i.e., attribute-value annotations of nodes,
ways, and relations. A sample from the LGD dataset de-
scribing a fire station is shown in the following triples.

```
lgd:node1119854639 a lgdo:Amenity, lgdo:FireStation, lgdo:Node;
    lgdo:directType lgdo:FireStation ;
    rdfs:label "Fire Service of Stagira - Akanthos" ;
    strdf:hasGeometry "POINT(23.8778 40.4003)"^^strdf:geometry .
```

GeoNames. GeoNames[18] is a gazetteer that collects both
spatial and thematic information for various placenames
around the world. GeoNames data is available through vari-
ous Web services but it is also published as linked data. The
features in GeoNames are interlinked with each other defin-
ing regions that are inside the underlined feature (*children*),
neighboring countries (*neighbors*) or features that have cer-
tain distance with the underlined feature (*nearby features*).
A sample from the GeoNames dataset describing the city of
Patras is shown in the following triples.

```
<http://sws.geonames.org/255683/> a gn:Feature ;
    gn:alternateName "Patrae" ;
    gn:alternateName "Patras"@en ;
    gn:name "Patras" ;
    gn:countryCode "GR" ;
    gn:featureClass gn:P ;
    gn:featureCode gnP.PPLA ;
    gn:parentADM1 <http://sws.geonames.org/6697810/> ;
    gn:parentCountry <http://sws.geonames.org/390903/> ;
    strdf:hasGeometry "POINT(21.73 38.24)"^^strdf:geometry .
```

3.2.4 Improving hotspot products using linked data

Let us now see how the datasets presented above can be
combined to improve the thematic accuracy of the gener-
ated hotspot products enabling the automatic generation of
related thematic maps.

---

[15]http://geodata.gov.gr/

[16]http://linkedgeodata.org/

[17]http://www.openstreetmap.org/

[18]http://www.geonames.org/

Improving the thematic accuracy. The thematic accu-
racy of the shapefiles generated by the processing chain is
improved by an additional processing step that refines them
by correlating them with auxiliary geospatial data. This is
done by a series of stSPARQL update statements that up-
date the RDF representation of the generated shapefiles by
taking into account relevant RDF data sets from the ones
presented above. As an example, consider the following up-
date query.

```
DELETE {?h ?property ?object}
WHERE {
    ?h a noa:Hotspot;
        strdf:hasGeometry ?hGeo;
        ?hProperty ?hObject.
    OPTIONAL {
        ?c a coast:Coastline ;
            strdf:hasGeometry ?cGeo .
        FILTER (strdf:anyInteract(?hGeo, ?cGeo))}
    FILTER(!bound(?c))}
```

The condition in the first FILTER pattern of this state-
ment utilizes the function `strdf:anyInteract`, which checks
if two spatial literals intersect with each other, while the con-
dition in the second FILTER pattern ensures that retrieved
hotspots do not intersect with land. Thus, it retrieves and
deletes hotspots lying entirely in the sea. Similarly, the fol-
lowing update statement retrieves hotspots that partially lie
in the sea and deletes the part of their geometry that lies in
the sea.

```
DELETE {?h strdf:hasGeometry ?hGeo}
INSERT {?h strdf:hasGeometry ?dif}
WHERE {
    SELECT DISTINCT ?h ?hGeo
        (strdf:intersection(?hGeo, strdf:union(?cGeo)) AS ?dif)
    WHERE {
        ?h a noa:Hotspot ;
            strdf:hasGeometry ?hGeo .
        ?c a coast:Coastline ;
            strdf:hasGeometry ?cGeo .
        FILTER(strdf:anyInteract(?hGeo, ?cGeo))}
    GROUP BY ?h ?hGeo
    HAVING strdf:overlap(?hGeo, strdf:union(?cGeo))}
```

In the above query, the spatial aggregate function
`strdf:union` of stSPARQL is utilized. For each hotspot
all coastline regions that intersect with it are grouped and
their union is calculated. Afterwards the part that is not
contained in this union is deleted from the geometry of the
hotspot.

Improving automatic map generation. In Section 2 we
explained that the automatic generation of fire maps en-
riched with relevant geo-information is of paramount impor-
tance to NOA since the creation of such maps in the past
has been a manual process. Using an stSPARQL endpoint
where the RDF datasets described above reside, a NOA op-
erator can now simply overlay the retrieved data using some
GIS software (e.g., QGIS or GoogleEarth). For example,
by posing the following queries and overlaying their results,
NOA operators can create a map like the one shown in Fig-
ure 6 that exploits information from the above datasets.

Query 1: "Get all hotspots in southeastern Peloponnese
that were detected from 23rd to 26th of August 2007" (from
Hotspot data).

```
SELECT ?hotspot ?hGeo ?hAcqTime ?hConfidence ?hProvider
        ?hConfirmation ?hSensor
WHERE {
```
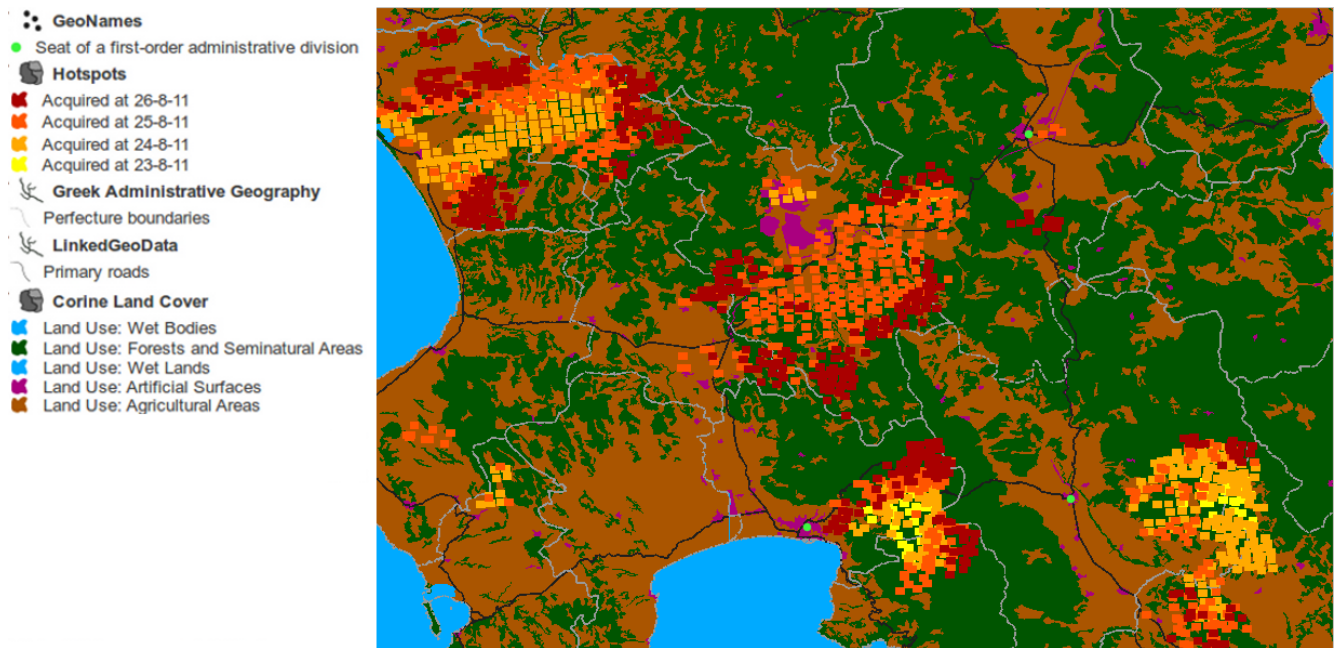
Figure 6: A map that can be created by overlaying data computed by stSPARQL queries

```
?hotspot a noa:Hotspot ;
noa:hasGeometry ?hGeo ;
noa:hasAcquisitionTime ?hAcqTime ;
noa:hasConfidence ?hConfidence ;
noa:isProducedBy ?hProvider ;
noa:hasConfirmation ?hConfirmation ;
noa:isDerivedFromSensor ?hSensor ;
FILTER( "2007-08-23T00:00:00" <= str(?hAcqTime) ) .
FILTER( str(?hAcqTime) <= "2007-08-26T23:59:59" ) .
FILTER( strdf:contains("POLYGON((21.027 38.36,
        23.77 38.36, 23.77 36.05, 21.027 36.05,
        21.027 38.36))"^^strdf:WKT, ?hGeo)).}
```

This query retrieves all hotspots that were detected between 23rd and 26th of August 2007 in southeastern Peloponnese the geometry of which is represented by the WKT literal mentioned in the last filter. Along with the URIs, additional information about the hotspots is retrieved, e.g., time of acquisition ?hAcqTime, provider ?hProvider and geometry ?hGeo.

Query 2: "Get the land cover of areas located in southeastern Peloponnese" (from Corine Land Cover data).

```
SELECT ?aLandUseType
WHERE {
    ?area a clc:Area ;
          clc:hasLandUse ?aLandUse ;
          noa:hasGeometry ?aGeo .
    ?aLandUse a ?aLandUseType .
    FILTER( strdf:contains("POLYGON((21.027 38.36,
        23.77 38.36, 23.77 36.05, 21.027 36.05,
        21.027 38.36))"^^strdf:WKT, ?aGeo) ) . }
```

Query 3: "Get all primary roads in southeastern Peloponnese" (from LinkedGeoData).

```
SELECT ?road ?rGeo
WHERE {
    ?road a lgdo:Primary ;
          noa:hasGeometry ?rGeo .
    FILTER( strdf:contains("POLYGON((21.027 38.36,
        23.77 38.36, 23.77 36.05, 21.027 36.05,
        21.027 38.36))"^^strdf:WKT, ?rGeo) ) .}
```

Query 4: "Get all capitals of prefectures of southeastern Peloponnese" (from GeoNames).

```
SELECT ?n ?nName ?nGeo
WHERE {
    ?n a gn:Feature ;
       strdf:hasGeometry ?nGeo ;
       gn:name ?nName ;
       gn:featureCode gn:P.PPLA .
  FILTER( strdf:contains("POLYGON((21.67 36.87,
        22.74 36.87, 22.74 37.68,
        21.67 37.68, 21.67 36.87))"^^strdf:geometry, ?nGeo))}
```

This query returns all features of the GeoNames dataset that are contained in a specific rectangle covering southeastern Peloponnese and their `gn:featureCode` equals to `gn:P.PPLA`. Such features are first-order administrative divisions (for Greece this corresponds to capitals of prefectures). Apart from thematic information about a feature (variables ?n and ?nName), its geometry (variable ?nGeo) is also returned so that it can be depicted on a map.

Query 5: "Get all municipality boundaries in southeastern Peloponnese" (from Greek Administrative Geography).

```
SELECT ?municipality ?mYpesCode ?mContainer ?mLabel
      ( strdf:boundary(?mGeo) as ?mBoundary )
WHERE {
    ?municipality a gag:Dhmos ;
                  noa:hasYpesCode ?mYpesCode ;
                  gag:isPartOf ?mContainer ;
                  rdfs:label ?mLabel ;
                  strdf:hasGeometry ?mGeo .
    FILTER( strdf:contains("POLYGON((21.027 38.36,
        23.77 38.36, 23.77 36.05, 21.027 36.05,
        21.027 38.36))"^^strdf:WKT, ?gGeo) ) . }
```

This query makes use of information related to the Greek Administrative Geography. Especially, it retrieves all lowest administrative divisions (municipalities) along with some information about them (?gYpesCode, ?gContainer and ?gLabel) along with their boundaries.

4. EVALUATION

In this section we compare the thematic accuracy achieved using hotspots derived by the original processing chain vs. using the refined ones obtained after applying the series of stSPARQL queries discussed in Section 3.2.4. We also present experimental results that enable us to compare the execution processing times of the hotspot detection chain that uses SciQL with the legacy versions of the same chain available at NOA. Finally, we evaluate the performance of Strabon in executing the refinement queries of Section 3.2.4.

4.1 Thematic Accuracy

Evaluating the thematic accuracy of the real-time fire monitoring products is not a straightforward task, as it entails the cross-validation of the hotspot products with ground-truth data. The latter is impossible to acquire for the large scale monitoring scenario applicable in this use case. Therefore, we adopted another approach by estimating the relevant thematic accuracy of the MSG/SEVIRI hotspot products with respect to the similar products generated using the MODIS sensor on top of the Terra and Aqua satellite platforms.

The methodology we followed for the comparison of the two hotspots products included: (i) Selection of an appropriate time span to run the experiments. This was selected to cover three full days in 2007 (24/08, 25/08, 26/08), when Greece was struck by the most severe forest wildfires of the last 20 years. (ii) Collection of the corresponding MODIS data from FIRMS[19] a NASA portal that integrates remote sensing and GIS technologies to deliver global MODIS fire locations and burned area information. (iii) Pre-processing of the MSG/SEVIRI dataset to be evaluated. Since the temporal resolution of the two datasets is different, we merged 30 minutes of MSG acquisitions (maximum three hotspot products) around the corresponding MODIS acquisition times. (iv) Application of some vector manipulation functions (from point to polygon and vice versa) in order to reliably estimate the overlapping regions of the two hotspot products.

A characteristic example of the false alarms and omission errors that the MSG/SEVIRI hotspot detection chain is prone to can be seen in Figure 7. Table 1, summarizes the resulting indexes of the thematic accuracy obtained using hotspots derived via the original processing chain and the refined ones after applying a series of stSPARQL queries. The validation protocol adopted to identify the thematic evaluation indexes is the following: for all MODIS timestamps available in the selected time span, the points corresponding to the total MODIS hotspot detections (second column in Table 1) were overlaid with the polygons of the total MSG hotspots (fifth column in Table 1). The number of MODIS hotspots falling inside the MSG polygons, with 700 m tolerance (accounting for the 1 km pixel size of MODIS), are registered in the third column of Table 1. Similarly, the MSG hotspots falling inside the MODIS polygons are noted in the sixth column of Table 1.

The output of the original, plain, chain produces an omission error of 12.71% and a false alarm rate of 26.20%. The corresponding values after the refinement process are 10.03% and 29.46% respectively. In general these values are about

---

[19]http://earthdata.nasa.gov/data/near-real-time-data/firms
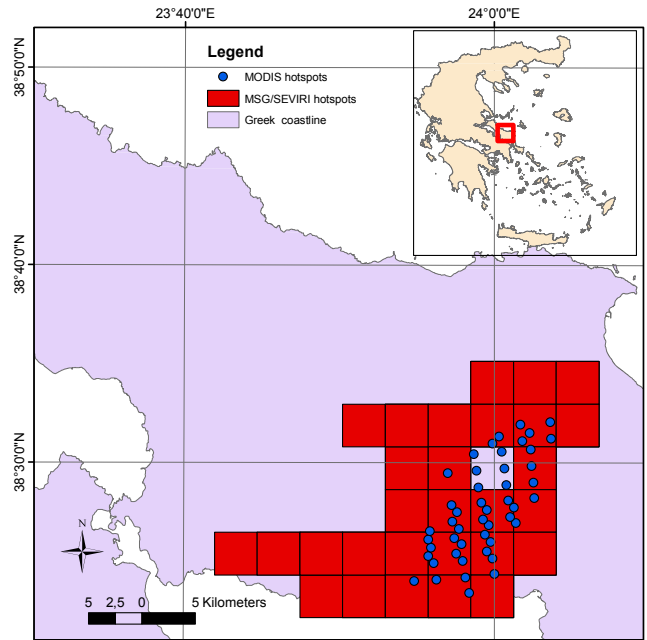
Figure 7: Typical example of false alarms due to smoke and omission error (void rectangle) of the MSG/SEVIRI product

the level expected based on joint data uncertainty. The improvement achieved via the application of semantic refinement queries is identified in a) the quantitative reduction of the omission error, and b) the enhanced qualitative consistency of the false alarms. The slightly elevated false alarm rate can be attributed to the inherent discrepancy in the spatial resolutions of the two sensors and the increased temperature sensitivity of MSG/SEVIRI near intense wildfires. Therefore, the false positives do not occur as isolated hotspots but as pixels classified as fires near neighboring pixels detected as fires by both the MSG/SEVIRI and MODIS sensors. Additionally, other types of false alarms, such as hot smoke fumes from nearby fires over inconsistent areas (sea, urban and agricultural areas) have now been eliminated completely.

4.2 Processing Times

In this section we include a performance evaluation of the two distinct phases of the fire monitoring procedure. Section 4.2.1 evaluates the fire detection algorithms and Section 4.2.2 the refinement steps performed using additional information stored in Strabon.

4.2.1 Hotspot detection algorithms

In this section we compare the original implementation of the NOA processing chain using the C programming language, which we denote as the *legacy* implementation, with the current implementation using SciQL.

Both versions are evaluated using all the image acquisitions of a particular day, namely the 22nd of August 2010. The dataset for this particular day contains image acquisitions for almost every 5 minute period, i.e., it contains 281 images, most of which contain confirmed fires. All experiments where performed on a Intel(R) Core(TM) i7 CPU 920

Table 1: Thematic accuracy for the original chain and after the implementation of the refinement queries

Processing Chain	Total number of MODIS hotspots	MODIS hotspots detected by MSG	**Omission error (%)**	Total number of MSG hotspots	MSG hotspots detected by MODIS	**False alarm rate (%)**
Plain chain	2542	2219	**12.71**	2710	2000	**26.20**
After refinement	2542	2287	**10.03**	3262	2301	**29.46**

Table 2: Processing times per image acquisition

Processing chain	Avg time per image (sec)	Min time per image (sec)	Max time per image (sec)
Legacy C	1.481058	1.215612	1.607081
SciQL	2.067308	1.902349	2.432782

processor running at 2.67 GHz with 3 GB of main memory. The system was running Linux while both MonetDB and the legacy C implementation were compiled with a 32-bit version of GCC 4.4 using the same optimization flags.

The assessment of both processing chains was performed as if the chains are black boxes, measuring the wall time that each chain requires for each image. Additionally we also recorded the total running time in seconds for the execution of the processing chains. The total running time includes a small additional overhead due to bookkeeping, such as moving around the necessary files, etc. Note that the most expensive part of the processing chain is the decompression of the original satellite image, which is performed in both cases by the same library. Table 2 presents the minimum, maximum and average time per image that is required for the execution of each chain.

In all 281 timestamps at this particular day of the year, both processing chains behave similarly. The legacy C chain is slightly faster than the SciQL implementation. This increase in execution time does not pose any significant problems in NOA, as it does not influence in any significant way the amount of time available to the refinement queries. Recall that both the processing chain execution and the refinement queries need to finish in less than 5 minutes. Moreover, NOA expects that the execution time of the SciQL chain will improve during the remainder of TELEIOS as the implementation of SciQL matures inside the MonetDB. On the other hand, the use of a high-level scripting language, such as SciQL, for the implementation of the processing chain significantly reduces the development effort. Common small changes, such as changing threshold values, are as easy as changing a few tuples in the DBMS, avoiding lengthy development sessions.

4.2.2 Refinement steps

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, could be performed.

Our experiments were carried out on an Ubuntu 11.04 installation on two Intel Xeon E5620 with 12 MB L2 cache
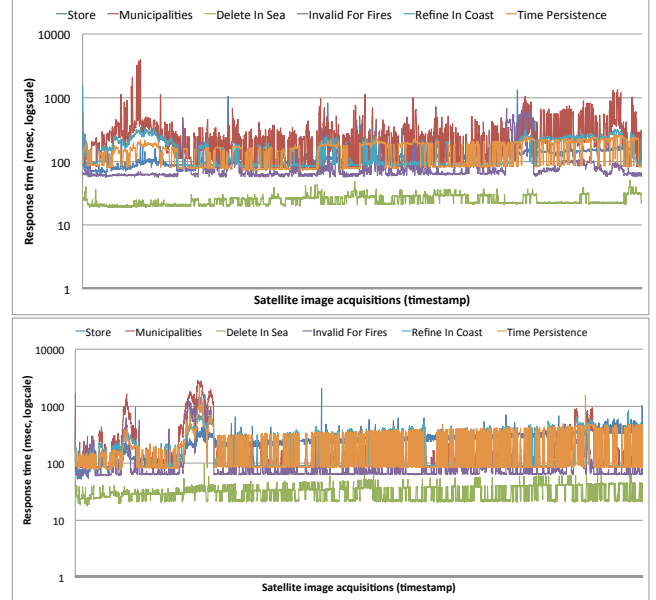
Figure 8: Response times for each (top) MSG1 and (bottom) MSG2 acquisitions

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 query posed for the respective operation by measuring the elapsed time from query submission till a complete iteration over each query's results had been completed.

In Figure 8, we observe that all operations are executed efficiently, mostly in less than a second, except for the operation of associating the detected hotspots with the municipality they belong to. This operation is labeled as "Municipalities" and is shown in red color, and, although for the most cases the query processing time does not exceed two seconds, there are cases where it needs four seconds to be completed. 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. In fact, Strabon is one of the most functional and probably the most scalable geospatial RDF store available today as the detailed analysis of [14] shows.

5. RELATED WORK

We now review some of the most relevant research efforts related to the topics discussed in this paper.

In the area of remote sensing, most of the fire detection al-

gorithms using MSG/SEVIRI data are based on variations of EUMETSAT's classification methodology [5] (EUMETSAT is the European organization managing the Meteosat series of geostationary meteorological satellites). The pre-TELEIOS approach used by NOA for this problem has been discussed in detail in [20, 8].

TELEIOS, the project in which the presented fire monitoring service has been developed, is a multidisciplinary research effort bringing together contributions from database management, semantic web and linked data, remote sensing and knowledge discovery from satellite images. The vision of TELEIOS has been discussed in [10, 12] and a demo of the fire monitoring service has been presented in [11, 13].

The chosen approach of using and extending database technology to satisfy the ever growing data management needs of domain scientists builds on the experiences and lessons learned from previous successful projects such as Paradise [17], Sequoia 2000 [21], Microsoft TerraServer [1], Sloan Digital Sky Survey [6]. TELEIOS extends this work by exploiting modern column-store technology, seamlessly integrating native array storage and processing functionality, as well as including linked data and semantic web technologies.

With respect to database systems offering array query processing capabilities there are only few systems that can handle sizable arrays efficiently. RasDaMan [3] is a domain-independent array DBMS for multidimensional arrays of arbitrary size and structure. RasDaMan provides a SQL-92 based query language, RasQL [2], to manipulate raster images using foreign function implementations and provides raster web services, which are based on OGC standards. Such web services are beyond the scope of TELEIOS. A recent attempt to develop an array database system from scratch is undertaken by the SciDB group [22]. Its mission is the closest to SciQL, but Version 0.5 and the design documents indicate that their language is a mix of SQL syntax and algebraic operator trees, instead of a seamless integration with SQL:2003 syntax and semantics. SciQL takes language design a step further by providing a seamless symbiosis of array-, set-, and sequence- interpretation using a clear separation of the mathematical object from its underlying implementation. A key innovation of SciQL is the extension of value-based grouping in SQL:2003 with structural grouping, which leads to a generalization of window-based query processing with wide applicability in science domains. Recently[20], MonetDB/SciQL, SciDB and RasDaMan have formed a working group to design a common syntax for an array manipulation and query language as well as an algebra for common array operations, acknowledging the experiences and lessons learned from all three projects.

In the context of the Semantic Web, the development of geospatial extensions to SPARQL has received some attention recently, which resulted in the creation of a recent OGC standard for querying geospatial data encoded in RDF, called GeoSPARQL [16]. Strictly speaking, stSPARQL and GeoSPARQL are incomparable in terms of representational power. If we omit aggregate functions and updates from stSPARQL, its features are a subset of the features offered by the GeoSPARQL core, geometry extension and geometry topology extension components. A detailed comparison of stSPARQL and GeoSPARQL is given in [9].

There have been some works in the past where ontologies

have been applied to the modeling of EO data [18, 4] or in a similar virtual observatory context [19, 15]. TELEIOS has benefited from the modeling concepts developed in these efforts and has tried to re-use parts of these public ontologies whenever possible.

6. CONCLUSIONS

In this paper we report on a virtual earth observatory that we are currently building in the context of the European project TELEIOS. Given the rapidly growing Earth Observation data archives, TELEIOS addresses the need for scalable access to petabytes of Earth Observation data and the discovery of knowledge that can be used in applications. To achieve this, TELEIOS aims at leveraging and extending data management technologies. The main focus is on scientific database technologies (array databases, SciQL, data vaults) and on geospatial Semantic Web technologies (stRDF and stSPARQL). Using a forest fire monitoring application as representative example, we discuss in detail how the developed technologies, integrated into MonetDB, a state-of-the-art open-source column-store database system, can be deployed to support and improve processing of large-scale Earth Observation data. While focusing on Earth Observation within the TELEIOS project, we are confident that the developed technologies can also be deployed in other scientific disciplines like astronomy, meteorology, seismology, biology, etc.

7. ACKNOWLEDGMENTS

8. ADDITIONAL AUTHORS

M. Karpathiotakis, K. Kyzirakos, K. Bereta, G. Garbis, C. Nikolaou (UoA); D. Michail (Harokopio University of Athens, Greece); I. Papoutsis, T. Herekakis (NOA); M. Ivanova, Y. Zhang, H. Pirk, M. Kersten (CWI); K. Dogani, S. Giannakopoulou, P. Smeros (UoA).

9. REFERENCES

[1] T. Barclay, J. Gray, S. Ekblad, E. Strand, and J. Richter. Designing and building terraservice. *IEEE Internet Computing*, 10(5):16–25, 2006.

[2] P. Baumann. A database array algebra for spatio-temporal data and beyond. In *Next Generation Information Technologies and Systems*, pages 76–93, 1999.

[3] P. Baumann, A. Dehmel, P. Furtado, R. Ritsch, and N. Widmann. The Multidimensional Database System RasDaMan. In *SIGMOD*, pages 575–577, 1998.

[4] C. Carlino, S. D. Elia, A. D. Vecchia, M. Iacovella, M. Iapaolo, C. M. Scalzo, and F. Verdino. On sharing Earth Observation concepts via ontology. In *ESA-EUSC*, 2008.

[5] EUMETSAT. Active Fire Monitoring with MSG Algorithm - Theoretical Basis Document. Technical report, EUM/MET/REP/07/0170, 2007.

[6] J. Gray and A. S. Szalay. Where the rubber meets the sky: Bridging the gap between databases and science. *IEEE Data Eng. Bull.*, 27(4):3–11, 2004.

---

[20]http://www.xldb.org/arrayql/

[7] M. Ivanova, M. L. Kersten, and S. Manegold. Data Vaults: a Symbiosis between Database Technology and Scientic File Repositories. In *SSDBM*, pages 485–494, 2012.

[8] C. Kontoes, I. Papoutsis, D. Michail, T. Herekakis, M. Koubarakis, K. Kyzirakos, M.Karpathiotakis, C. Nikolaou, M. Sioutis, G. Garbis, S. Vassos, I. Keramitsoglou, M. Kersten, S. Manegold, and H. Pirk. Wildfire monitoring via the integration of remote sensing with innovative information technologies. In G. R. Abstracts, editor, *EGU2012-PREVIEW*, 2012.

[9] M. Koubarakis, M. Karpathiotakis, K. Kyzirakos, C. Nikolaou, and M. Sioutis. Data Models and Query Languages for Linked Geospatial Data. In T. Eiter and T. Krennwallner, editors, *Reasoning Web. Semantic Technologies for Advanced Query Answering*, volume 7487 of *Lecture Notes in Computer Science*, pages 290–328. Springer Berlin / Heidelberg, 2012.

[10] M. Koubarakis, M. Karpathiotakis, K. Kyzirakos, C. Nikolaou, S. Vassos, G. Garbis, M. Sioutis, K. Bereta, S. Manegold, M. L. Kersten, M. Ivanova, H. Pirk, Y. Zhang, C. Kontoes, I. Papoutsis, T. Herekakis, D. Michail, M. Datcu, G. Schwarz, O. Dumitru, D. Espinoza-Molina, K. Molch, U. D. Giammatteo, M. Sagona, S. Perelli, E. Klien, T. Reitz, and R. Gregor. Building virtual earth observatories using ontologies and linked geospatial data. In *RR*, pages 229–233, 2012.

[11] M. Koubarakis, K. Kyzirakos, M. Karpathiotakis, C. Nikolaou, S. Vassos, G. Garbis, M. Sioutis, K. Bereta, D. Michail, C. Kontoes, I. Papoutsis, T. Herekakis, S. Manegold, M. Kersten, M. Ivanova, H. Pirk, Y. Zhang, M. Datcu, G. Schwarz, O. C. Dumitru, D. E. Molina, K. Molch, U. D. Giammatteo, M. Sagona, S. Perelli, T. Reitz, E. Klien, and R. Gregor. TELEIOS: A Database-Powered Virtual Earth Observatory. *Proceedings of the VLDB Endowment (PVLDB)*, 5(12), 2012.

[12] M. Koubarakis, M. Sioutis, K. Kyzirakos, M. Karpathiotakis, C. Nikolaou, S. Vassos, G. Garbis, K. Bereta, O. C. Dumitru, D. E. Molina, K. Molch, G. Schwarz, and M. Datcu. Building Virtual Earth Observatories using Ontologies, Linked Geospatial Data and Knowledge Discovery Algorithms. In *ODBASE'12*, 2012.

[13] K. Kyzirakos, M. Karpathiotakis, G. Garbis, C. Nikolaou, K. Bereta, M. Sioutis, I. Papoutsis, T. Herekakis, D. Michail, M. Koubarakis, and C. Kontoes. Real Time Fire Monitoring Using Semantic Web and Linked Data Technologies. In *ISWC'12*, 2012.

[14] K. Kyzirakos, M. Karpathiotakis, and M. Koubarakis. Strabon: A Semantic Geospatial DBMS. In *ISWC'12*, Boston, USA, 2012.

[15] D. L. McGuinness, P. Fox, L. Cinquini, P. West, J. Garcia, J. L. Benedict, and D. Middleton. The virtual solar-terrestrial observatory: A deployed semantic web application case study for scientific research. In *AAAI*, 2007.

[16] O. G. C. I. OGC. GeoSPARQL - A geographic query language for RDF data, November 2010.

[17] J. M. Patel, J.-B. Yu, N. Kabra, K. Tufte, B. Nag, J. Burger, N. E. Hall, K. Ramasamy, R. Lueder, C. J. Ellmann, J. Kupsch, S. Guo, D. J. DeWitt, and J. F. Naughton. Building a scaleable geo-spatial dbms: Technology, implementation, and evaluation. In *SIGMOD Conference*, pages 336–347, 1997.

[18] M. Podwyszynski. Knowledge-based search for Earth Observation products. Master's thesis, Passau Univ., 2009.

[19] R. Raskin and M. Pan. Knowledge representation in the semantic web for Earth and environmental terminology (SWEET). *Computers & Geosciences*, 2005.

[20] N. Sifakis, C. Iossifidis, C. Kontoes, and I. Keramitsoglou. Wildfire Detection and Tracking over Greece Using MSG-SEVIRI Satellite Data. *Remote Sensing*, 2011.

[21] M. Stonebraker. Sequoia 2000: A reflection of the first three years. In *SSDBM*, pages 108–116, 1994.

[22] M. Stonebraker, J. Becla, D. J. DeWitt, K.-T. Lim, D. Maier, O. Ratzesberger, and S. B. Zdonik. Requirements for Science Data Bases and SciDB. In *CIDR*, 2009.

[23] Y. Zhang, M. L. Kersten, M. Ivanova, and N. Nes. SciQL: Bridging the Gap between Science and Relational DBMS. In *IDEAS*, pages 124–133, 2011.